

CONFERENCES IN RESEARCH AND PRACTICE IN
INFORMATION TECHNOLOGY

VOLUME 87

DATA MINING AND ANALYTICS 2008



AUSTRALIAN
COMPUTER
SOCIETY



DATA MINING AND ANALYTICS 2008

Proceedings of the
Seventh Australasian Data Mining Conference (AusDM'08),
Glenelg, South Australia, 27-28 November, 2008

John F. Roddick, Jiuyong Li, Peter Christen and
Paul Kennedy, Eds.

Volume 87 in the Conferences in Research and Practice in Information Technology Series.
Published by the Australian Computer Society Inc.



Published in association with the ACM Digital Library.

Data Mining and Analytics 2008. Proceedings of the Seventh Australasian Data Mining Conference (AusDM'08), Glenelg, South Australia, 27-28 November, 2008

Conferences in Research and Practice in Information Technology, Volume 87.

Copyright ©2008, Australian Computer Society. Reproduction for academic, not-for-profit purposes permitted provided the copyright text at the foot of the first page of each paper is included.

Editors: **John F. Roddick**
School of Computer Science, Engineering and Mathematics
Flinders University
GPO Box 2100, Adelaide, SA, 5001, Australia
Email: john.roddick@flinders.edu.au

Jiuyong Li
School of Computer and Information Science
University of South Australia, Mawson Lakes
GPO Box 2471, Adelaide, SA, 5001, Australia
Email: jiuyong.li@unisa.edu.au

Peter Christen
Department of Computer Science
Faculty of Engineering and Information Technology
The Australian National University
Canberra ACT 0200 Australia
Email: peter.christen@anu.edu.au

Paul J. Kennedy
Faculty of Engineering and Information Technology
University of Technology, Sydney
Broadway, NSW, 2007, Australia
Email: paulk@it.uts.edu.au

Series Editors:
Vladimir Estivill-Castro, Griffith University, Queensland
John F. Roddick, Flinders University, South Australia
Simeon Simoff, University of Western Sydney, NSW
crpit@csem.flinders.edu.au

Publisher: Australian Computer Society Inc.
PO Box Q534, QVB Post Office
Sydney 1230
New South Wales
Australia.

Conferences in Research and Practice in Information Technology, Volume 87
ISSN 1445-1336
ISBN 978-1-920682-68-2

Printed November 2008 by Flinders Press, PO Box 2100, Bedford Park, SA 5042, South Australia.
Cover Design by Modern Planet Design, (08) 8340 1361.

The *Conferences in Research and Practice in Information Technology* series aims to disseminate the results of peer-reviewed research in all areas of Information Technology. Further details can be found at <http://crpit.com/>.

Table of Contents

Proceedings of the Seventh Australasian Data Mining Conference (AusDM'08), Glenelg, South Australia, 27-28 November, 2008

Preface	vii
Organising Committee	viii
AusDM Sponsors	ix
Conference Programme	x

Keynote Papers

Minors as Miners - Modelling and Evaluating Ontological and Linguistic Learning.....	3
<i>David M. W. Powers</i>	
Multi-Strategy Ensemble Learning, Ensembles of Bayesian Classifiers, and the Problem of False Discoveries	15
<i>Geoff Webb</i>	
Volume, Velocity and Variety - Key Challenges for Mining Large Volumes of Multimedia Information	17
<i>Richard Price</i>	

Contributed Papers

Algorithms

On Inconsistencies in Quantifying Strength of Community Structures	21
<i>Wen Haw Chong</i>	
A New Evaluation Measure for Imbalanced Datasets.....	27
<i>Cheng G. Weng and Josiah Poon</i>	
LBR-Meta: An Efficient Algorithm for Lazy Bayesian Rules	33
<i>Zhipeng Xie</i>	

Approaches for Business and Organisations

Exploratory Mining over Organisational Communications Data	41
<i>Alan Allwright and John F. Roddick</i>	
Towards Scalable Real-Time Entity Resolution using a Similarity-Aware Inverted Index Approach...	51
<i>Peter Christen and Ross Gayler</i>	
Customer Event Rate Estimation Using Particle Filters	61
<i>Harsha Honnappa</i>	
Priority Driven K -Anonymisation for Privacy Protection.....	73
<i>Xiaoxun Sun, Hua Wang and Jiuyong Li</i>	

Association Rules / Frequent Patterns

ShrFP-Tree: An Efficient Tree Structure for Mining Share-Frequent Patterns	79
<i>Chowdhury Farhan Ahmed, Syed Khairuzzaman Tanbeer, Byeong-Soo Jeong and Young-Koo Lee</i>	
Rare Association Rule Mining via Transaction Clustering	87
<i>Yun Sing Koh and Russel Pears</i>	
S ² MP: Similarity Measure for Sequential Patterns	95
<i>Hassan Saneifar, Sandra Bringay, Anne Laurent and Maguelonne Teisseire</i>	
Mining Medical Specialist Billing Patterns for Health Service Management	105
<i>Yin Shan, David Jeacocke, D. Wayne Murray and Alison Sutinen</i>	

Biomedical Data Mining

Comparison of Visualization Methods of Genome-wide SNP Profiles in Childhood Acute Lymphoblastic Leukaemia	111
<i>Ahmad Al-Oqaily, Paul J. Kennedy, Daniel Catchpoole and Simeon Simoff</i>	
Classification of Brain-Computer Interface Data	123
<i>Omar AlZoubi, Irena Koprinska and Rafael A. Calvo</i>	
Kernel-based Visualisation of Genes with the Gene Ontology	133
<i>Hamid Ghous, Paul J. Kennedy, Daniel R. Catchpoole and Simeon J. Simoff</i>	
wFDT - Weighted Fuzzy Decision Trees for Prognosis of Breast Cancer Survivability	141
<i>Umer Khan, Hyunjung Shin, Jong Pill Choi and Minkoo Kim</i>	

Engineering Applications

Identifying Stock Similarity Based on Multi-event Episodes	153
<i>Abhi Dattasharma, Praveen Kumar Tripathi and Sridhar G</i>	
Evaluation of Malware clustering based on its dynamic behaviour	163
<i>Ibai Gurrutxaga, Olatz Arbelaiz, Jesús M^aPérez, Javier Muguerza, José I. Martín and Iñigo Perona</i>	
Service-independent payload analysis to improve intrusion detection in network traffic	171
<i>Iñigo Perona, Ibai Gurrutxaga, Olatz Arbelaiz, José I. Martín, Javier Muguerza and Jesús M^aPérez</i>	
Graphics Hardware based Efficient and Scalable Fuzzy C-Means Clustering	179
<i>S.A. Arul Shalom, Manoranjan Dash and Minh Tue</i>	
Indoor Location Prediction Using Multiple Wireless Received Signal Strengths	187
<i>Kha Tran, Dinh Phung, Brett Adams and Svetha Venkatesh</i>	

Text Mining

Clustering and Classification of Maintenance Logs using Text Data Mining	193
<i>Brett Edwards, Michael Zatorsky and Richi Nayak</i>	
Categorical Proportional Difference: A Feature Selection Method for Text Categorization	201
<i>Mondelle Simeon and Robert Hilderman</i>	
Structure-Based Document Model with Discrete Wavelet Transforms and Its Application to Document Classification	209
<i>Supphachai Thaicharoen, Tom Altman and Krzysztof J. Cios</i>	
Combining Structure and Content Similarities for XML Document Clustering	219
<i>Tien Tran, Richi Nayak and Peter Bruza</i>	

Author Index	227
--------------------	-----

Preface

It is our pleasure to welcome you to the Seventh Australasian Data Mining Conference (AusDM'08) being held this year at Glenelg in South Australia. AusDM started in 2002 and is now the annual flagship meeting for data mining and analytics professionals in Australia. Both scholars and practitioners present the state-of-the-art in the field. Endorsed by the peak professional body, the Institute of Analytics Professionals of Australia, AusDM has developed a unique profile in nurturing this joint community. The conference series has grown in size each year from early workshops held in Canberra (2002, 2003), Cairns (2004), Sydney (2005, 2006) and the Gold Coast (2007). This year's event has been supported by

- Togaware, again hosting the website and the conference management system, coordinating the review process and other essential expertise;
- Flinders University and the University of South Australia for providing the venue, registration facilities and various other support;
- the Institute of Analytic Professionals of Australia (IAPA) for facilitating the contacts with the industry;
- the ARC Research Network on Data Mining and Knowledge Discovery, for providing financial support;
- the Australian Computer Society, for publishing the conference proceedings;
- data mining postgraduate students from Flinders University and the University of South Australia for their local support.

The conference program committee reviewed 55 submissions, out of which 25 submissions were selected for publication and presentation. 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. We would like to thank all those who submitted their work to the conference. We will continue to extend the conference format to be able to accommodate more presentations.

In addition, three keynote speakers were invited. Professor Geoff Webb from Monash University talked on *Multi-Strategy Ensemble Learning, Ensembles of Bayesian Classifiers, and the Problem of False Discoveries*, Professor David Powers from Flinders University gave a talk on *Minors as Miners: Modelling and Evaluating Ontological and Linguistic Learning* and Dr Richard Price from the Defence Science and Technology Organisation presented a talk on *Volume, Velocity and Variety - Key Challenges for Mining Large Volumes of Multimedia Information*.

The conference also included the first AusDM Doctoral Consortium and a tutorial on *Privacy preserving data sharing and mining* from Jiuyong Li, Peter Christen, Vladimir Estivill-Castro and Artak Amirbekyan.

We would like to extend our special thanks to the program committee members. The final quality of selected papers depends on their efforts. The review cycle runs on a very tight schedule and we would like to thank all reviewers for their commitment and professionalism.

**John F. Roddick,
Jiuyong Li,
Peter Christen and
Paul Kennedy**

AusDM08 Organising Chairs
November 2008

Organising Committee

Program Chair	John Roddick, Flinders University, Adelaide
Organising Chair	Jiuyong Li, University of South Australia
General Chairs	Peter Christen, Australian National University Paul Kennedy, University of Technology, Sydney
Steering Committee Chairs	Simeon Simoff, University of Western Sydney Graham Williams, Australian Taxation Office
Other Steering Committee Members	Vladimir Estivill-Castro, Griffith University Warwick Graco, Australian Taxation Office Inna Kolyshkina, Westpac Banking Corp. Richi Nayak, Queensland University of Technology, Brisbane
Programme Committee Members	Hussein Abbass, ADFA, Canberra, Australia Rohan Baxter, Australian Taxation Office, Canberra, Australia Vladimir Estivill-Castro, Griffith University, Queensland, Australia Ross Gayler, Veda Advantage, Melbourne, Australia Raj Gopalan, Curtin University of Technology, Perth, Australia Lifang Gu, Australian Taxation Office, Canberra, Australia Robert Hilderman, University of Regina, Canada Joshua Huang, University of Hong Kong, Hong Kong Warren Jin, NICTA, Canberra, Australia Yun Sing Koh, Auckland University of Technology, New Zealand Gang Li, Deakin University, Victoria, Australia Xuemin Lin, University of New South Wales, Sydney, Australia John Maindonald, Australian National University, Canberra, Australia Bradley Malin, Vanderbilt University, Nashville, USA Arturas Mazeika, Free University of Bozen, Italy Yvonne Morrow, Centrelink, Canberra, Australia Richi Nayak, Queensland University of Technology, Brisbane, Australia Christine O'Keefe, CSIRO, Canberra, Australia Kok-Leong Ong, Deakin University, Victoria, Australia Tom Osborn, The Leading Edge, Sydney, Australia Robert Pearson, Canberra, Australia Francois Poulet, IRISA - Texmex, Rennes, France Ben Raymond, Australian Antarctic Division, Hobart, Australia Claude Sammut, University of New South Wales, Sydney, Australia Kate Smith-Miles, Deakin University, Victoria, Australia David Taniar, Monash University, Melbourne, Australia Jim Warren, University of Auckland, New Zealand John Yearwood, University of Ballarat, Victoria, Australia Ting Yu, University of Sydney, Australia Huaifeng Zhang, Centrelink, Canberra, Australia Yanchang Zhao, University of Technology, Sydney, Australia

AusDM Sponsors

We wish to thank the following sponsors for their contribution towards this conference.



<http://www.togaware.com>



FLINDERS UNIVERSITY
ADELAIDE • AUSTRALIA

School of Computer Science, Engineering and Mathematics
<http://csem.flinders.edu.au>



**University of
South Australia**

School of Computer and Information Science
<http://www.cis.unisa.edu.au>



<http://www.iapa.org.au>



ARC Research Network on Data Mining and Knowledge Discovery
<http://www.dmkd.flinders.edu.au>

Conference Programme

Thursday, 27 November, 2008

08:00 - 09:00 **Arrival Coffee**

09:00 - 09:05 **Opening and Welcome**

09:15 - 10:15 **KEYNOTE - Ballroom 1**

Minors as Miners: Modelling and Evaluating Ontological and Linguistic Learning
David M.W. Powers, *Flinders University*

10:15 - 10:40 **Coffee break**

10:40 - 12:40 **Session 1: Text Mining - Ballroom 1**

10:40 - 11:10 *Clustering and Classification of Maintenance Logs using Text Data Mining*
Brett Edwards, Michael Zatorsky, Richi Nayak

11:10 - 11:40 *Combining Structure and Content Similarities for XML Document Clustering*
Tien Tran, Richi Nayak, Peter Bruza

11:40 - 12:10 *Structure-Based Document Model with Discrete Wavelet Transforms
and Its Application to Document Classification*
Supphachai Thaicharoen, Tom Altman, Krzysztof Cios

12:10 - 12:40 *Categorical Proportional Difference: A Feature Selection Method for
Text Categorization*
Mondelle Simeon, Robert Hilderman

10:40 - 12:40 **Session 2: Engineering Applications - Colley 1**

10:40 - 11:10 *Indoor Location Prediction Using Multiple Wireless Received Signal Strengths*
Kha Tran, Dinh Phung, Brett Adams, Svetha Venkatesh

11:10 - 11:40 *Evaluation of Malware clustering based on its dynamic behaviour*
Ibai Gurrutxaga, Olatz Arbelaitz, Jesús M^aPérez, Javier Muguerza,
José I. Martín and Iñigo Perona

11:40 - 12:10 *Service-independent payload analysis to improve intrusion detection
in network traffic*
Iñigo Perona, Ibai Gurrutxaga, Olatz Arbelaitz, José I. Martín, Javier Muguerza and
Jesús M^aPérez

12:10 - 12:40 *Graphics Hardware based Efficient and Scalable Fuzzy C-Means Clustering*
S.A. Arul Shalom, Manoranjan Dash, Minh Tue

12:40 - 13:30 **Lunch**

13:30 - 14:30 **Tutorial - Ballroom 1**

Privacy preserving data sharing and mining
Jiuyong Li, Peter Christen, Vladimir Estivill-Castro and Artak Amirbekyan

13:30 - 14:30 **PhD Consortium - Colley 1**

14:30 - 15:00 **Coffee**

13:30 - 14:30 **Tutorial continued - Ballroom 1**

Privacy preserving data sharing and mining
Jiuyong Li, Peter Christen, Vladimir Estivill-Castro and Artak Amirbekyan

13:30 - 14:30 **PhD Consortium continued - Colley 1**

19:00 **Conference Dinner**

Friday, 28 November, 2008

08:00 - 09:00 **Arrival Coffee**

09:00 - 10:00 **KEYNOTE - Ballroom 1**

*Multi-Strategy Ensemble Learning, Ensembles of Bayesian Classifiers,
and the Problem of False Discoveries*

Geoff Webb, Monash University

10:00 - 10:30 Coffee break

10:30 - 12:30 **Session 3: Biomedical Data Mining - Ballroom 1**

10:30 - 11:00 *Classification of Brain-Computer Interface Data*

Omar AlZoubi, Irena Koprinka, Rafael Calvo

11:00 - 11:30 *Kernel-based visualisation of genes with the Gene Ontology*

Hamid Ghous, Paul Kennedy, Daniel Catchpoole, Simeon Simoff

11:30 - 12:00 *Comparison of visualization methods of genome-wide SNP profiles in
childhood acute lymphoblastic leukaemia*

Ahmad Al-Oqaily, Paul Kennedy, Daniel Catchpoole, Simeon Simoff

12:00 - 12:30 *wFDT - Weighted Fuzzy Decision Trees for Prognosis of
Breast Cancer Survivability*

Umer Khan, Hyunjung Shin, Minkoo Kim, Jong Pill Choi

10:30 - 12:00 **Session 4: Association Rules / Frequent Patterns - Colley 1**

10:30 - 11:00 *Rare Association Rule Mining via Transaction Clustering*

Yun Sing Koh, Russel Pears

11:00 - 11:30 *S2MP: Similarity Measure for Sequential Patterns*

Hassan Saneifar, Sandra Bringay, Anne Laurent, Maguelonne Teisseire

11:30 - 12:00 *Mining medical specialist billing patterns for health service management*

Yin Shan, David Jeacocke, D. Wayne Murray, Alison Sutinen

12:30 - 13:30 Lunch

13:30 - 14:30 **KEYNOTE - Ballroom 1**

*Volume, Velocity and Variety - Key Challenges for Mining Large
Volumes of Multimedia Information*

Richard Price, Defence Science and Technology Organisation

14:30 - 15:00 Coffee break

15:00 - 16:30 **Session 5: Approaches for Business and Organisations - Ballroom 1**

15:00 - 15:30 *Priority Driven K-Anonymisation for Privacy Protection*

Xiaoxun Sun, Hua Wang, Jiuyong Li

15:30 - 16:00 *Exploratory Mining over Organisational Communications Data*

Alan Allwright, John Roddick

16:00 - 16:30 *Towards Scalable Real-Time Entity Resolution using a
Similarity-Aware Inverted Index Approach*

Peter Christen, Ross Gayler

15:00 - 16:30 **Session 6: Algorithms - Colley 1**

15:00 - 15:30 *On Inconsistencies in Quantifying Strength of Community Structures*

Wen Haw Chong

15:30 - 16:00 *A New Technique for Evaluating Imbalanced Datasets*

Cheng Weng, Josiah Poon

16:00 - 16:30 *LBR-Meta: An Efficient Algorithm for Lazy Bayesian Rules*

Zhipeng Xie

KEYNOTE PAPERS

Minors as Miners

Modelling and Evaluating Ontological and Linguistic Learning

David M W Powers

AILab, School of Computer Science, Engineering and Mathematics
Flinders University of South Australia
PO Box 2100, Adelaide 5001, South Australia

David.Powers@flinders.edu.au

Abstract

Growing up is in large measure learning about the world and our social and linguistic environment. We might call this data mining, although it is far more multimodal and immersive than most applications. This paper describes computational research into how children learn, with a particular focus on evaluation in both supervised and unsupervised paradigms.

Conversely, we gain additional insight into association mining by considering psycholinguistic experiments that quantify the way human association by both adults and children relate to a variety of association measures. Learning and evaluation are not dealt with in isolation, but a program of formal and application-based evaluation is expounded and exemplified to show how to evaluate discovered patterns with and without a gold standard.

In this context, some serious issues with current evaluation techniques and accuracy measures are identified and the unbiased techniques identified.

Keywords: Natural Language Learning, Data Mining, Text Mining, Signal Processing, Speech Processing, AudioVisual Speech Recognition, Cognitive Linguistics, Computational Psycholinguistics, DeltaP, Receiver Operating Characteristics, Bookmaker Informedness and Markedness, Brain Computer Interface.

1 Introduction

Over the last 60 years, human-like performance by computers in tasks requiring broad cognitive and linguistic competence has remained elusive. In many specific areas, solid algorithms and useful methodologies have been developed and been hived off from Artificial Intelligence as fields in their own right, or have emerged independently from the seeds of AI. In the 70s and 80s Cognitive Science emerged as an interdisciplinary nexus that took over the traditional psychological modelling of AI in the 50s and 60s, leaving AI to become increasingly algorithm oriented and focussed on engineering goals. On the other hand, Computational Intelligence emerged to espouse the softer fuzzier aspects that the AI community seemed to be resistant to (leaving behind GOFAL, Good Old-Fashioned AI). These fuzzier aspects included Fuzzy Logic, Neural Networks, Ant Colony Optimization,

Genetic Programming and a variety of other approaches based on biological or physical metaphors, whilst Cognitive Science explored sometimes similar models based on stronger ideas of biological plausibility.

This paper is written in the context of a program of research undertaken by the author since the mid-70s, focussed on the idea of getting computers to learn to understand the world, and language, the way a baby does. However, this paper will not attempt a logical or chronological development of Computational Psycholinguistics, but will focus specifically on aspects of relevance to Data Mining, including in particular *evaluation*.

2 Evaluation

How do you evaluate patterns? Where we are doing *unsupervised learning* with no teacher to guide us with appropriate examples and no marker to grade our efforts, how do we know how useful our patterns or rules are? Where we are doing *supervised learning* and are aiming to achieve a specific objective, how do we rate our system and which of the many competing measures do we use? And how do children rate the patterns and rules they learn?

2.1 Evaluation in Applications

One answer to the problem of evaluation in unsupervised learning is to find and use an appropriate gold standard – which begs the questions of where this comes from, how reliable it is, and if it is reliable why we are bothering with trying to learn it. If the answer is that it is one person's theory, then it is inherently subjective and begs the circular question as to how that theory was evaluated.

Another answer is to turn it into a supervised problem with measurable outcomes. Thus phonologies, grammars, ontologies, etc. may be evaluated by embedding them in an application where there is inherent and objective performance evaluation – for example in web search, machine translation, speech recognition, lip reading, electroencephalographic computer interface, etc.

If the question is which paper is more relevant, or which gloss (translation) of a word is more appropriate, there is usually little doubt unless they are roughly equally good, and human raters are well qualified to make these judgements. On the other hand, if it comes to deciding between two competing grammars, it seems that there are more grammars than linguists, and none of them are likely to have much relationship with what goes on in our heads.

A third approach is to appeal to some concept of parsimony. The child's problem of learning about the

world is very similar to the scientist's problem of learning about the world, and good scientific method has specific biases to theories that are simple and testable. However parsimony and testability relate to theories that are already shown to be equally good on some objective evaluation.

Evaluation measures based on parsimony tend to have an information theoretic basis, using overall evaluation paradigms like Minimum Message Length, or local measures employing conditional entropy or mutual information. In many cases, such as log-likelihood models, this use is blind to the effectiveness of the outcome and more about significance. In other cases, such as in building decision trees, the usage is more like a heuristic and aimed at building a smaller model rather than a more correct model. In both cases, it recognizes that improved performance from overtuning is misleading.

2.2 Supervised Evaluation & Gold Standards

Although the focus in this paper is unsupervised learning and data mining, we commence by examining evaluation in the context of supervised learning, as well as association learning as investigated in children. We will however in the process relate this to unsupervised learning, clustering and association rule mining before considering these paradigms closely in the next section.

We will consider the value of rules that predict a Result **R** based on a Precondition **P**, where we assume **P** and **R** take the same labels representing the predicted class and the real class. In the binary or dichotomous case we have in evaluating a single rule **P**→**R**, **P** or **R** may take only the two values true (+) or false (-). Table 1 shows two standard notations for labeling the contingency table showing the 4 combinations possible. Both of these are used in upper case variants summing to N and lower case variants normalized to probabilities that sum to 1, with the first version being mnemonic (e.g.true or false positive, predicted or real negative).

Precision, known as Confidence in data mining, is a form of accuracy based on the proportion of positive predictions that have correct outcomes (true positive accuracy, $tpa = tp/pp = TP/PP$). Recall measures the rate of finding positives and is the proportion of positive outcomes that have correct predictions (true positive rate, $tpr = tp/rp = TP/RP$). Support measures $tp = TP/N$, which is proportional to Recall as RP and RN and N are assumed to be constants related by fixed Prevalence $rp = RP/N$ and Inverse Prevalence, $rn = RN/N$, being the set of real marginal statistics. One of the sources of problems in evaluation is that the prediction marginal statistics are not in general fixed and act as biases, Bias $pp = PP/N$ and Inverse Bias $pn = PN/N$.

Severe bias problems with Recall and Precision have been demonstrated by Powers (1997 with Entwisle, 2003, 2007 and 2008) from a theoretical and empirical statistical perspective (proposing Informedness and Markedness), Perruchet and Peereman (2004) from a theoretical and empirical psychological perspective (proposing DeltaP and DeltaP'), and Flach (2003 and 2005 with Fürnkranz) from a theoretical machine learning perspective (proposing the concept of skew and WRAcc). Similar issues with Confidence and Support in

association mining date back equally far with for example Brin et. al (1997) proposing Conviction and Interest (aka Lift). Note that Lift ($tp/[pp \cdot rp]$) is a ratio of actual frequency to expected frequency (joint probability to product of Prevalence and Bias) and Leverage is the difference between actual and expected frequency, being proposed by Piatetsky-Shapiro (1991) even before the classic advocacy of support in Apriori (Agrawal et al., 1993). Note further that pointwise Mutual Information uses $\log(Lift)$ to assess individual rules. Conviction ($[pp \cdot rn]/fp$) is the *reciprocal* of Lift applied to the cell/rule **+P**→**-R**, reflecting a desire not only to see that tp is *high* relative to chance, but that fp is *low* relative to chance. However, recall that rp and rn are constants of the dataset that are assumed to apply to any random sample (from the dataset or collected in the future). This means that Lift is equivalent to Confidence or Precision apart from a linear scale factor (which may be useful if thresholds are employed). Similarly, Lift is equivalent to $1/[1-Confidence]$ and thus is equivalent to Confidence or Precision or Overgeneralization (see below) except for the non-linear scaling (which may be useful if thresholds are employed).

Other measures that normalize fp are Fallout (false positive rate, $fpr = fp/rn = FP/RN$, the proportion of negative outcomes that incorrectly have positive predictions) and Imprecision = $1 - Precision = 1 \div Overgeneralization$ (false positive accuracy, $fpa = fp/pp = FP/PP$, the proportion of positive predictions that incorrectly have negative outcomes). It is also possible to normalize fn as Missrate or Inverse Fallout (false negative rate, $fnr = fn/rp = FN/RP$) and as Inverse Imprecision (false negative accuracy, $fna = fn/pn = FN/PN$, the proportion of negative predictions that incorrectly have positive outcomes). Similarly tn has normalizations corresponding to Inverse Precision and Inverse Recall reflecting the result of application of Precision and Recall to the inverse rule **-P**→**-R**, and all Inverse measures are interpretable this way and are also complements of other named measures.

This Inverse problem is technically a different (dual) problem as it uses the rule in the opposite way to its logical intent (it is abductive and equivalent to **P**←**R**). However under conditions of forced single choice it is effectively used this way by virtue of the closed world assumption (if we don't say it's positive it is negative and vice-versa, which is typical of a neural net or decision tree, but not of association rules, as discussed below).

	+R	-R		
+P	tp	fp	pp	
-P	fn	tn	pn	
	rp	rn	1	

	+R	-R		
+P	A	B	A+B	
-P	C	D	C+D	
	A+C	B+D	N	

Table 1. Systematic and traditional notations in a binary contingency table. Colour coding indicates correct (green) and incorrect (pink) rates or counts in the contingency table. Left table is systematic terminology based on true/false/real/predicted positives and negatives and in lower case represents probabilities and in UPPER case counts. The right table is an common alternative notation.³³

2.2.1 Informedness, Markedness & Correlation

We now introduce Informedness (DeltaP' or skew-insensitive WRAcc) and Markedness (DeltaP). Informedness has been advocated by several authors under its various names as discussed previously, and shown to be unbiased, corresponding to the probability of making an informed decision versus a chance decision (Powers, 2003). Shanks (1995) calls DeltaP "the normative measure of contingency" in that it explains human association data so much better than direct unnormalized measures such as Precision or Confidence.

In the dichotomous case we have been discussing,

$$\text{Informedness} = \text{Recall} - \text{Fallout} = \text{Recall} + \text{InvRecall} - 1 \\ = [\text{Recall} - \text{Bias}] / \text{Inverse Prevalence}.$$

We can thus see that it takes into account Fallout (f_p a constant scaled normalization of f_p) as well as Recall (t_{pr} a constant scaled normalization of t_p), that it reflects equally both the forward and inverse problems, that it is effectively a (constant scaled) renormalization after subtracting the Bias. Although directly based on Recall-like measures, the difference of t_{pr} and f_{pr} , Informedness is also qualitatively similar to Precision as the ratio of t_p and f_p . When Precision is 1, $t_p = p_p$ and $f_p = 0$, so that $f_{pr} = 0$, and Informedness = Recall. When Recall=1, Informedness = InverseBias/InversePrevalence and is thus only maximized when Bias=Prevalence. This matching of Bias to Prevalence is a common heuristic.

The dual of Informedness is Markedness or DeltaP:

$$\text{Markedness} = \text{Precision} + \text{Inverse Precision} - 1 \\ = [\text{Precision} - \text{Prevalence}] / \text{Inverse Bias}.$$

This is thus based on the Precision-like measures but has some similarity to Recall.

Both Informedness and Markedness are unbiased unlike other common averages of Precision and Recall, the F-Factor and Rand Accuracy. Flach's skew insensitive version of F-Factor and Precision remain similar, whilst the skew insensitive (skins) version of Accuracy and Weighted Relative Accuracy (WRAcc) are equivalent to Informedness (BMI) and the ROC area under the curve (AUC) with the relationship:

$$\text{skinsAcc} = \text{AUC} = [\text{BMI} + 1] / 2 = [\text{skinsWRAcc} + 1] / 2.$$

Informedness and Markedness are not in general independent as they are regression coefficients for dual problems, and thus by definition their geometric mean is Correlation (Perruchet & Peereman, 2004; Powers, 2007 & 2008). The correlation of Informedness and Markedness themselves tends to be about 0.5 in a study by Perruchet & Peereman (2004) that investigated the learning of phonological associations in children and adults and found that Frequency, Markedness, Informedness and Pearson/Matthews Correlation (their geometric mean) correlated significantly more strongly with children and adult performance than Precision and Recall, in the indicated order of increasing correlation ($p < 0.005$ in all cases for adults, and $p < 0.05$ in all cases for children, where the difference was less marked). This indicates that we learn associations based on both forward and backward predictability (corresponding to both classical and operant conditioning) but give slightly more weight to the forward direction (using well marked predictors to successfully predict well marked outcomes).

2.2.2 An Example of Need for Informedness

Precise formulae and vague statements about bias are neither of them enough to give a good feel for how serious the problem is with Precision and Recall, or Confidence and Support, or F-Factor and Accuracy. Thus it is appropriate to give some examples. In the discussion of the various experiments below we will see examples where Accuracy increases and Informedness decreases – and this has been mind-blowing for the students working on the project who were sceptical about technicalities.

However, here I will go into one example, which was one of those that originally inspired the development of Informedness and is discussed in Entwisle and Powers (1997): the problem is *when water is a noun or a verb*. The real problem is that several real-life parsers and taggers made the deliberate decision that their systems were so bad at deciding part of speech that they could increase their F-scores and/or Accuracies by saying water was always a noun, which it is 90% of the time. It is instructive to do the maths and see that chance level for guessing with Bias matching Prevalence gives Recall = Precision = Bias = Prevalence = 90%, Inverse Precision = Inverse Recall = Inverse Bias = Inverse Prevalence = 10% and F-Factor is thus 30% and Rand Accuracy 82%. However, by saying it is always a noun we set Recall = 100%, Precision = 90%, F-Factor = 95% and Rand Accuracy = 90%.

2.2.3 A Feel for Informedness & Markedness

For Informedness and Markedness any form of guessing, whether following Prevalence or some other random Bias, whether always setting positive or always setting negative, always gives the same long term result: an expected value of 0. For a perfect performance with no errors, Informedness and Markedness will both be 1. Informedness tells you the proportion of the time your predictor made an informed (correct) decision versus guessed (and averaged the expected value over time). Markedness tells you the proportion of the time the outcome actually marked the predictor correctly (gave rise to the symptom or indicator), as opposed to it taking a random value (viz. expected value over time).

2.2.4 The General Case ($K > 2$)

In the above we considered a single rule $P \rightarrow R$ where each of the variables was restricted to take a Boolean or dichotomous value. In general, there may be more than one choice or more than one rule. To the extent that the rules are independent, this latter point need not concern us as we can calculate Informedness and Markedness separately for each rule. To the extent that we can build additional evidence for a particular decision we are moving beyond the paradigm of the contingency table. In fact, we can combine weightings for different rules in any way we like, but this introduces the concept of cost, where as the Bookmaker and ROC principles behind Informedness and Markedness are based on an unbiased model with skew or costs determined by relative prevalence – the more likely a horse is to win, the lower the odds the Bookmaker will give you. Adding different costs or penalties to specific outcomes, changes the biases that are appropriate to achieve an optimum payoff. So we will ignore this for the time being, and revisit the question

of multiple overlapping rules and their effect on cost/skew.

Here we will deal with only the fact that contingency tables may be any size, and for $K \times K$ tables with $K > 2$ the above formulae don't work. In fact the modification is fairly simple – we perform a weighted average of the Informedness or Markedness determined for a single label. For each label we effectively have a binary contingency table regarding whether that label was the prediction and whether it was the outcome. For Informedness we weight by Prevalence, and this is why Informedness tends to be of most practical value, and can be empirically significantly more predictive of human performance (Perruchet & Peereman, 2004). For Markedness we weight by Bias.

The original Bookmaker Informedness derivation (Powers, 2003) was expressed in probabilistic notation in a form that weighted over a table of costs associated with the respective cells of the contingency table, with the costs being determined by Bookmaker bets and payoffs (for fair odds). Mutual Information is similarly a weighted average based on an information theoretic value equivalent to $\log(\text{Lift})$ as discussed above (Powers, 2003). Multiplying this Mutual Information by N in the general case (Powers, 2008), gives the χ^2 significance (log-likelihood or G^2), whereas multiplying Correlation by N in the binary case (Perruchet & Peereman, 2004) fits the χ^2 distribution. For the general case it is necessary to multiply the Correlation by $(K-1)N$ to approximate χ^2 and Powers (2003) derives corrections that allow greater accuracy. In addition χ^2 significance can be calculated separately in a similar way directly from Informedness and Markedness, and confidence intervals can also be estimated directly (Powers, 2007 and 2008).

The general Informedness and Markedness formulae can be elegantly expressed in ways which clearly show the simplified dichotomous form (the original formulation did not reveal the simple connections with Recall, Bias and Prevalence, or with WRAcc and DeltaP'). We define Bookmaker Informedness (BI) and Bookmaker Markedness (BM) as follows, and we refer to the secondary sums we average over as the dichotomous Informedness $B(l)$ and Markedness $M(c)$, and the weighted terms as $BI(l)$ and $M(c)$:

$$BI = \sum_{l \in P} \text{Bias}(l) \sum_{c \in R} \text{Recall}_l(c) \frac{\text{Prev}_c(c)}{\pm \text{Prev}_l(c)} \quad (1)$$

$$BM = \sum_{c \in R} \text{Prev}(c) \sum_{l \in P} \text{Precision}_c(l) \frac{\text{Bias}_l(l)}{\pm \text{Bias}_c(l)} \quad (2)$$

$$\pm \text{Prev}_l(c) = \text{Prevalence}(l) - (c \neq l) \quad (3)$$

$$\pm \text{Bias}_c(l) = \text{Bias}(c) - (l \neq c) \quad (4)$$

The cost factors, the reciprocal probabilities represented by the \pm terms in the denominators, will have a sign depending on whether the prediction was accurate (rewarded) or inaccurate (penalized). In the dichotomous case it will cancel with its numerator as the stake you lose is what the bookmaker wins and amount you win is what the bookmaker loses – notice that this cancelation says the expected win is +1 and the expected loss is -1 (sum of prevalence times a payoff is the weighted average).

However, in the multi-horse case, for Bookmaker Informedness for the classic “edge”, your loss if you lose is dependent only on the odds for the horse you bet on (l), not the horse that actually won (c). Your win (expected +1) comes at the expense of many losers, and your loss (expected to be better than -1) is not the whole of the winner's gain. For Informedness your risk is determined by your prediction (which horse you bet on), not your outcome (which horse won). Markedness reverses this as if the outcome was the bet and the prediction determined the payoff (in practice odds *are* set by bias in the bets).

Note that for the dichotomous case you have only one degree of freedom and are making only one binary decision, so $BI = B(l)$ and $BM = M(c)$ for both positive and negative cases.

2.3 Unsupervised Evaluation by Gold Standard

The previous examples assumed a Gold Standard, viz. that we knew what the correct answers were, implying a supervised training and test set. But in fact we can do unsupervised training and still test with a Gold Standard if one exists – often this will be a small hand tagged corpus, perhaps tagged by multiple annotators to allow testing for subjective interannotator differences. In such a case the Informedness and Markedness calculations can proceed as above, effectively discounting for the chance baseline. We can also calculate Informedness and Markedness between two annotators and choose the one with greater Informedness as our Gold Standard (this will be the Markedness for the other annotator). This provides a human baseline which is not discounted automatically, and in fact it is often treated incorrectly as an upper bound – some of the experiments reported here exceed human performance.

2.3.1 Hard Clustering

One particular form of unsupervised learning is clustering, and there are a wide range of techniques that can be used to compare clusterings, or clusterings with a Gold Standard (Pfitzner, Leibbrandt & Powers, 2008). Some of these are based on pair counting (how many of the possible pairs occur in the same cluster for both clusterings), and the pair counting results themselves can be put into a contingency table allowing use of all of the measures we have been discussing.

However, a small set of clear cases can be used to match up unsupervised clusters and Gold Standard classes, for purposes of evaluation (generally, the classes or clusters are wanted for some purpose and can be used directly without use of seeding by a Gold Standard, but sometimes specific classes are required and the best possible matching is required for our application – a good evaluative application will not have this semisupervised requirement).

The question is how to match these up. The original Bookmaker paper (Powers, 2003) dealt with this form of unsupervised learning, assuming a hard clustering (every case is a member of exactly one class with no fuzzy membership function) and determining that for each cluster it is allocated to the class which it had the highest probability of labelling correctly – that is the highest number of instances, Precision or Confidence in the row. For a number of classes $C > K$, we would expect to

sometimes get more than one cluster contributing to a class, and even with $C=K$ this will be the case if one class doesn't get assigned a cluster.

However, it is appropriate, in supervised or semisupervised approaches, to use Informedness directly as the measure to optimize, rather than some arbitrary heuristic. Equation 1 can be applied pointwise to each unsupervised rule or cluster u we are considering adding to the support for a particular class label l . At this point we are assuming hard clustering and hard classification – a given data point or situation is in exactly one cluster and exactly one gold class, and we will assign it exactly one label. Omitting a class effectively includes it with a chance level informedness of 0, so we multiply $B(l)$ terms, the dependent internal sum of (1), by the true bias, $\text{Bias}(l) = p(l)$, according to (1), where all probabilities are calculated relative to the total number of items, N .

The internal sum involves weighted recall, where $\text{Recall}(c) = p(l|c)$, and will need to be accumulated across all clusters assigned label l . This allocation has usually been done by some heuristic that may introduce a bias, e.g. Powers (2003) used the most popular label in each cluster (or weighted them if equal), which corresponds to maximizing precision for each cluster. Equation (1) seems to suggest we should maximize Recall, but it is not so simple because of the weighting by the Bias, Prevalence and Cost terms. Moreover, even maximizing the pointwise Bookmaker $B(l)$ terms is not sufficient due to the $\text{Bias}(l)$ weighting. Empirically, maximizing Precision works better than Recall. However to maximize unbiased cost benefit of the predictions it is recommended to optimize for BI. It is also convenient that the cost factor is independent of the labelling of u for BI (1), although the biases do depend on u and l for BM (2).

If we seek to optimize the cluster labelling iteratively or recursively, by exposing the probabilities underlying $\text{Bias}(l)$ and $B(l)$ it is clear that both factors, $p(l)$ and $p(l|c)$, will be incremented, by respectively $p(u)$ and $p(u|c)$, so we must not maximize $p(u) \cdot p(u|c)$, but rather the increase in $p(l) \cdot p(l|c)$ that would be achieved by making the $u=l$ assignment. Viz. Maximize $\Delta \text{BI}(l) = \text{BI}(l) - \text{BI}(l+u)$.

This is reminiscent of Ward's method, which empirically usually gives the best discrimination in clustering, where we effectively consider the effect of merging two clusters (in this case l and u) rather than using direct distance measures (Powers, 1997a).

2.3.2 Soft Associations

Soft clustering allows items to be in more than one class, and often associates a weight. Data-oriented methods can associate items according to relative distance from cluster centroids. Fuzzy classes or sets can have weights or membership functions that express degree of membership in a class or applicability of a label. Bags allow multiple instances of an item in the same class, which can also therefore be represented as a set with associated counts, also interpretable as a form of weighting. Association mining will in general allow items or itemsets to predict multiple distinct items, and conversely some items will never be predicted. Strengths associated with rules are also a form of weighting.

All of these paradigms take us away from the contingency table with its assumptions of mutual exclusion between categories, its binary yes/no nature, and its effective closed world assumption – anything not stated to be true is false and vice-versa. In referring to a conjunction of items rather than items, we move to a contingency table on the powerset of the items in which many items do not occur.

Under the constraint that there is a weight of 1 associated with each label and class, a contingency table can be produced by accumulating weighted membership information (e.g. predicted Coke or Pepsi \rightarrow Coke 0.6, Pepsi 0.4). Also for any labelling, such a normalization constraint can be achieved for a set of latent classes by finding the eigenvectors using singular valued decompositions or similar algorithms (Powers, 1997a).

The problem of unselected or underselected items, labels whose weights don't sum to one, can be solved by simply including an additional 'no-prediction' group used as a class for items that don't predict anything particular (predicts just about everything at near chance levels and hence not significantly), and a label on dummy rules for items that are never predicted (predicted at close to chance level and hence not significantly). These dummy classes will reduce Recall and Precision, Informedness and Markedness, to correct levels – without them they would be overinflated.

There are also some issues that can most easily be clarified using the notation of clausal logic – reversing the traditional form of an association rule to the traditional form of clausal logic:

Coke \leftarrow Frozen Fish \wedge Frozen Chips

Pepsi \leftarrow Frozen Fish \wedge Frozen Chips

do not mean quite the same thing as

Coke \vee Pepsi \leftarrow Frozen Fish \wedge Frozen Chips

which implies Coke and Pepsi are alternatives rather than being independent. This is the difference between non-Horn and Horn notations, with the Horn limitation to exactly one prediction per item requiring explicit statement of exclusions (e.g. \neg Pepsi). The assignment of weights can thus add even more precision, but has essentially the character of alternation when they are conditional probabilities that sum to 1:

Coke(0.6) \vee Pepsi(0.4) \leftarrow Frozen Fish \wedge Frozen Chips

In this case we seem to have confidence- or precision-like weights, indicating what proportion of the time we predict Coke or Pepsi, once this rule fires, but the true precision for Coke and Pepsi based on this rule involves multiplying by the precision of the rule, the probability that the rule is correct, irrespective of weights. If the weights sum to more than one, it indicates that some households buy both Coke and Pepsi and they are not totally mutually exclusive.

However, as we have been discussing, Precision is misleading because it reflects overall Prevalence – the fact that 80% of people buy Coke and only 20% by Pepsi would seem to mean that people who buy Fish and Chips are *less* likely to buy Coke! Nonetheless these are appropriate as the respective weights for the original pair of rules (and thus precision and prevalence) in reporting

results in a contingency table, and we can then calculate total or pointwise Informedness in the standard way.

Informedness tells you what proportion of sales you have predicted rather than guessed, while Markedness tells you what proportion of bought products are markers of other needs rather than chance associations.

3 Language Technology Applications

We now return to our primary focus on text mining and unsupervised learning of linguistic and ontological rules and categories. We will review the work in this area bottom up, starting from raw audio, video or character data, starting with textual input.

3.1 Structural Learning

Notice the emphasis on character data – that is the form in which text comes, and conventions about words and spaces are not universal and not reliable, so even for English there is some effort required to establish what the words are. This is the word segmentation problem and relates to other specialized problems such as named entity recognition (International Business Machines), other similar noun collocations that aren't entities as such (Object-Oriented Programming), separable and inseparable verbs involving particles (put up X, put X up, put up with), and composite content and function words (object-oriented, 'in your face', to day, vs to-day vs today, into vs out of). Note the convention of either hyphenating or quoting when a phrase is pressed into service as an adjective. Note that spaces and quotes tend to moderate to hyphenation and eventually disappear as a phrase becomes accepted as a word.

A similar problem occurs in Chinese where the characters are like English morphemes or syllables, and content words normally consist of multiple characters. Spoken English doesn't come nicely packaged into words either, and we are increasingly wanting to work with spoken language. As we aggregate units into bigger units, segmentation becomes the basis for a kind of structural learning that encompasses the phonological, morphological, grammatical and prosodic aspects of language – all without any semantic information.

Techniques based on conditional entropy (confidence- or precision-like measures) are fairly good for assessing how likely the next character or syllable is to be part of the word, and techniques based on mutual information (leverage-like) are good for determining boundaries of words or other higher level units (Magerman, 1991). The combination of the two techniques can be even more powerful (Huang & Powers, 2004). These techniques can also be used to detect affixes and clitics – functional words, prefixes and suffixes, which are important foundations for full syntactic analysis. With just this information, entire sentences can be parsed (Entwisle, 1997) without knowing the actual content words: cf. the slithy toves did gyre and gimble in the wabes (Lewis Carroll, *Alice in Wonderland*).

These functional elements (morphs, which include also the null morph or null inflection \emptyset) have also been used to achieve effective spelling correction (Powers, 1997b; Huang and Powers, 2001) and have similar applications in disambiguation of confusable words for

speech recognition, optical character recognition, sign language recognition and machine translation.

The approach taken here is pure text mining and reflects several of the issues discussed in section two. Each functional context (e.g. for the previous sentence from Alice: the $-y$ $-s$ did $-\emptyset$ and $-\emptyset$ in the $-s$) provides a solid grammatical basis for distinguishing confusable words or multiple meanings that can be disambiguated on the basis of part of speech – it deliberately avoids semantic information.

This reverse approach of clustering based on contextual information is also powerful and by allowing pairs or triples, implicit segmentation can be performed while categorizing characters or words into classes (Powers, 1983, 1991). By replacing segments by non-terminal symbols, a finite or context-free grammar can be induced, including left-, right- or centre-recursive rules by allowing the proposed non-terminal to be included in its own contexts (Powers, 1992), although generally the non-recursive grammars were found to be more stable.

Given the assumption of word segmentation, functional words can be reasonably well distinguished by frequency alone – the 150 most frequent words of English constitute about half of any text, and are mainly functional words with a primarily grammatical function, or placeholder words (like thing, person or place) that have a similar function. Again we can generalize and collect sequences of words (out of) that are very frequent, and define these as templates that connect to an adjacent word ('the X' or 'out of the X' are both templates that characterize nouns, although the second has stronger implications that X refers to some place). It is also possible to have contexts with two separated open class and/or two separated closed class words ('the X of Y').

In child speech and child-directed speech, many sentences have this character, whereas in adult speech sentences will tend to combine several phrases and/or clauses each of which have this character. The child already recognizes key aspects of their native language by the time they are born, and at a very early age English-exposed babies can be shown to be sensitive to these "closed class" functional words that characterize English. There are also intonational and voicing features that characterize these closed class words as functional in nature – they tend to be less stressed, and for English words that start with the voiced /dh/ sound of 'the' are *all* functional words and constitute around 20% of a typical text.

By using a corpus of conversations with children (CHILDES) it is possible to pick up frequent templates that are entire utterances and have forms like those illustrated above. It is hypothesized that children listen only to those template-bound portions of any sentences that are currently too complex for them. But as templates become units in their own right, more complex templates involving them can be learned, including recursive usages. Leibbrandt (2008) has successfully modelled this process.

3.2 Semantics and Ontology

Powers (1983, 1989 with Turk, 1992) argued that structural learning can go only so far in learning language, and that it is necessary to learn about the world,

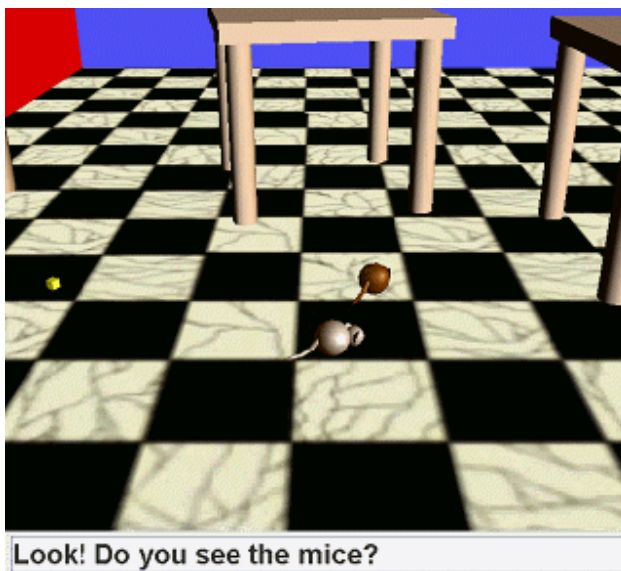


Figure 1. Example teaching scenario in MicroJaea robot world developed for teaching computer syntax and semantics, and also used by the Teaching Head.

Reproduced by permission of DMW Powers and R Leibbrandt

introducing the word ‘Ontology’ into Artificial Intelligence and Natural Language Processing from Philosophy where it denotes the study of what is and the development of models or theories of the world. Children are like scientists, finding patterns and testing theories, and this applies both to the structural aspects of language and to the way they structure and make sense of their world. The thesis that it is not possible to learn language fully without this kind of knowledge of the world later came to be called symbol grounding – as symbols are meaningless until grounded in reality (Harnad, 1990).

Powers and Turk (1989) also claim that this grounding contributes further to overcoming the so-called Poverty of the Stimulus problem touted in the 1980s by Chomskian linguists, but that there is no theoretical necessity to require such grounding to learn syntax, exposing a paradigm they called anticipated correction to explain how children could recognize that erroneous utterances didn’t sound right. Syntax by its nature is just rules that are slavishly obeyed by speakers and hearers in producing and interpreting language – if the meaning of the words is known in context, and hence the grammatical role of the word is clear, then syntax determines the grammatical rules of ordering the words as well as the cohesive connections between words (such as agreement and anaphora). Ambiguity in context is rare, and is usually corrected or repaired before or shortly after completion of the utterance, or observed with a wry “pun not intended”, or is a deliberate pun or a related form of humour.

Another important aspect that relates to semantics and ontology is metaphor. This is not just a term for tired phrases your English teacher explained to you, but is at the heart of how both language and learning work. Nothing is ever exactly the same, as time marches on, so does age, decay, dust, etc. It’s called entropy! So we are always classifying things as similar rather than dissimilar, and the classes we come up with have to do with the prevalence of the different exemplars and the need for particular features to contrast functionally different things

– two different fruit or two different people. If all apples or all oranges, or all Asians or all Caucasians, look the same, that’s because of lack of experience of naming them apart for some purpose.

Clustering is about grouping things together that are similar, and the density of clusters tends to relate to the density of items to be clustered – the more items in a region of attribute space, the more clusters as the smaller the thresholds on distance between member and non-member. This is why absolute thresholds are inappropriate, and it also brings into question nearest neighbour type algorithms. However, Powers (1991, 1992) is the only algorithm I know of based on cardinality rather than distance. On the other hand, self-organizing maps (e.g. Kohonen maps) do self-organize with a cluster area that is a direct function of density.

Powers (1983) introduced the idea that the same grammars and learning algorithms we use to learn language, we can use to learn about the world, specifying the development of a robot world simulation (Hume, 1984) using a grammar like notation that allowed learning meanings of nouns and verbs (Powers and Turk, 1989) as well as prepositions (Homes, 1994). The current version (Leibbrandt, 2008) is illustrated in Fig. 1. In this view, we have mechanisms that are designed to learn about the world and when we use them to interpret utterances we will do best with ones that exhibit the same biases we have in the world, the part-whole kind of structure, objects holding together rather than persisting in parts that move around independently, the various conservation laws. Thus we would expect grammar to derive from ontological learning rather than independently.

The field of Cognitive Linguistics is based on the centrality of the idea of metaphor, and distances itself from the Chomskian claim that language is a separate modality, claiming that it is integral with and inseparable from the rest of our cognitive processing. In particular, Deane (1992) extensively developed the idea of the part-whole nature of grammar deriving from the part-whole nature of the world.

But how we learn about the world and language should also be useful for teaching about the world (inc. maths, science, numeracy) and language (inc. literacy).

Recently however, the term Ontology is being used to describe taxonomies, thesauri, semantic networks, and text mark-up based on these. These are not truly grounded and don’t correspond to a true semantics, but to a pseudosemantics or logical semantics. Nonetheless they

Noun Similarity	r	r ²
Resnik*	0.791	0.626
Jiang & Conrath	0.828	0.686
Lin*	0.834	0.696
Average Human	0.902	0.814
Yang & Powers*	0.921	0.848
Verb Similarity	r	r ²
Yang & Powers	0.833	0.694
Average Human	0.866	0.751

Table 2 Comparison of results of published noun and verb similarity algorithms using Wordnet or Roget.

*Difference versus *human baseline* significant to $p < .05$.

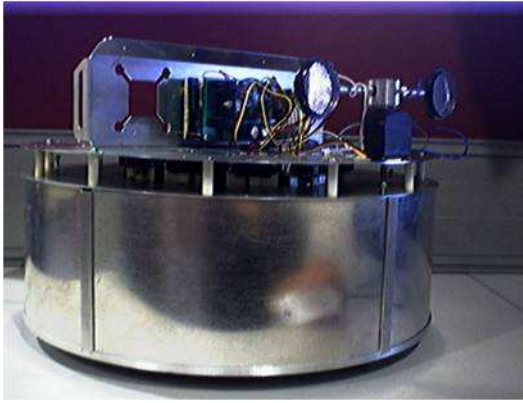


Figure 2. Can Robot with Sonar, Infrared and Webcam tracking options, plus can mount additional webcams and additional stages surmounted by a laptop – this has been used for Wizard of Oz building guide research.

can be useful, and we have explored algorithmic use of such WordNet to determine word similarity, as well as unsupervised self-organization of semantic networks and thesauri, achieving results that are comparable with average human subjects versus averages across subjects and published gold standards in fact significantly achieving significantly better than human performance for nouns (Yang and Powers, 2005) and breaking new ground for verbs (Yang and Powers, 2006b) – see Table 2. This has translated to extremely high accuracy in selecting the correct gloss for French to English Machine Translation, and is being explored as an automatic mechanism for disambiguation for the speech recognition, emotion recognition and topic selection components of the Thinking Head.

Unsupervised semantic network and automatic thesaurus construction (ATC) is difficult to evaluate, but for comparable size similarity classes relative overlap between each pair of Roget, WordNet and ATC are not significantly different. The ATC was developed using simple text mining techniques as described above, based on up to three templates for each part of speech (Yang and Powers, 2006a, 2008).



Figure 3. Mark 1 Robot Baby, 8 mikes and 8 touch sensors + 5 motors – crawl or look towards touch or sound.



Figure 4. Mark 2 Robot Baby, has 2 USB AV webcams mounted with additional eye convergence motor.

4 Heads Up!

4.1 The Talking Head

As well as working with simulated robots, it is worth trying to learning language and ontology in the real world, with real robots, sensors and actuators.

We have worked both with baby-like robots (Fig. 3-4, Powers, 2001) and with garbage-can-like robots (Fig. 2) – not to mention micromice, lego robots and a variety of other physical critters of the mechatronic persuasion. But whilst this has produced nice demonstrations, most researchers revert sooner or later to simulation to tune their systems and develop their learning algorithms, and we are no exception. Our robot baby could crawl, turn its head to the sound of a voice or a touch, and that was about it. The micromice can zoom around a maze, and the can can guide a visitor round the building. But it is too much work and too much maintenance and much too irrelevant for everyday language learning research.

4.1.1 Sensors, Signals and Fusion

On the other hand the sensors can be deployed separately – the sensors for detecting faces, eyes and lips, and then gaze-tracking and lip-reading (Lewis and Powers, 2002), the sensors for detecting objects, detecting and calibrating motion (Matsumoto, Powers and Asgari, 2008), the sensors for detecting people coming and going, their identity and their expressions and emotions (Luerssen, Lewis, Leibbrandt and Powers, 2008) – these sensors are just simple cameras and microphones, mostly cheap webcams. The Informedness measures proved particularly important in the fusion of different sources of information with different numbers of classes and different biases and prevalences, enabling a clear unbiased evaluation.

Fusion of information from multiple sources becomes a major goal when we add multiple sensors – simply throwing everything in together (early fusion) tends to produce catastrophic results (sometimes worse than either source alone). Similarly analysing each separately and then combining can lead to catastrophic fusion with a result significantly worse than the best signal alone. We have therefore worked on developing techniques that guarantee that the fusion will not be significantly worse than the best signal, and will usually be better (Lewis and Powers, 2005). This is achieved by identifying orthogonal features and training them separately, and we also have investigated techniques to automatically assess error.

Researchers have tended to become too specialized – one only works on gaze-tracking, or dialogue, text mining, or grammar induction, or speech recognition, or speaker, or expression/emotion recognition. But common techniques are used in many of these areas, and what is noise for one researcher is the goal for another. We have also used many of the techniques we developed for language and learning in biomedical image processing and brain computer interface research using electroencephalography (EEG), including unsupervised techniques for signal separation, supervised techniques for optimal fusion, and it was in this context that we first got clear examples where conventional accuracy measures were increasing but the true utility, as measured by Bookmaker Informedness, was decreasing (Fitzgibbon,

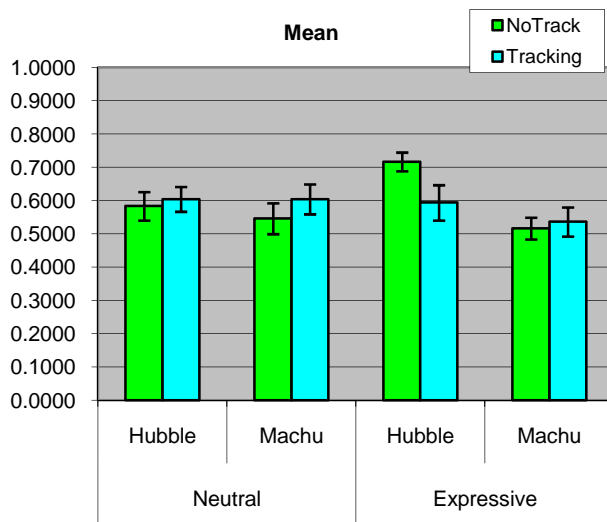


Figure 5. Up to 20% absolute gain in comprehension achieved by the Teaching Head's students comparing the most appropriate and least appropriate gaze tracking and expression mark up. What is appropriate still needs further formal evaluation.

Powers, Pope and Clark, 2007), and much of the methodology developed in this context has been reapplied back in the speech and language area.

The original motivation for our work with EEG was to study the predictions about closed and open class words that emerged from our unsupervised learning research, as well as to understand some of the conscious and unconscious processing of speech – indeed we demonstrated a clear affect from inaudible subliminal audio (Powers, Dixon, Clark and Weber, 1996). We have also used EEG to determine where a subject is in their learning curve, investigating also how this relates to their attention and awareness available for other purposes – this is a theme that recurs in our human factors approach to user interface design. Currently we are looking at how to fuse biological signals and audio-visual signals for improved learning.

The Talking Head as a surrogate for a robot can be run on any old computer or laptop, can make use of any old webcam or even the built-in laptop camera and mike, and provide many of the benefits of a robot – being with its cameras and microphones embedded in the world, embodied notwithstanding its lack of a body, grounded although not crawling around on the ground.

4.2 The Thinking Head

Connected to a simple dialogue engine or “bot”, a Talking Head becomes a Thinking Head that can engage in conversations or act as a kiosk in a museum, describing and answering questions about its accompanying exhibit. With cameras and microphones, it immediately becomes more human-like if it can do speech recognition (in the noisy environment – using lip reading) and face recognition (recognizing people returning, or even just change of speaker and colour of shirt). By adding emotional expression to face or voice, and including appropriate eye tracking, it can become not only more human-like but achieve higher performance in getting information across (Powers, Leibbrandt, Pfitzner,

Luerksen, Lewis, Abrahamyan and Stevens, 2008; see Figure 5Figure 6).

But instead of operating in, understanding and being grounded in a real environment, like a museum, we can provide simulated grounding in a simulated environment doing simulated museum tours, or anything else we like.

4.3 The Teaching Head

What is becoming our major application for the Thinking Head at Flinders, is the Teaching Head. Our research has been focussed on teaching computers language – speech, syntax, semantics, ontology, etc. Our robot worlds and virtual environments were developed for this purpose and set up as teaching scenarios. The obvious application is to turn the scenarios around and make the computer, the Teaching Head, the teacher rather than the learner.

Our initial target for this is teaching English and German as a second or foreign language, with a particular focus on the German noun phrases, and the associated prepositional/declensional system (Leibbrandt, Luerksen, Matsumoto, Treharne, Lewis, Li Santi and Powers, 2008).

A key aspect of the Teaching Head is its hybrid environment (Figure 6) – user and environment are monitored by three cameras and the same props/toys are simulated in the virtual world so both teacher and student can illustrate sentences or obey commands. We have a 3D touch screen and are also exploring camera-based tracking, so there is no need for the user to use a keyboard and mouse – it thus doesn't feel like you are using a computer at all!

A variety of additional teaching opportunities have emerged for the Teaching Head, including several related to health, and several related to specialist education...

4.3.1 VALIANT - Virtual Agent for Literacy and Numeracy Tutoring

The Thinking and Teaching Head have been displayed at many exhibitions and art-galleries and open days. The original head captivates with conversations about just about anything, and particularly attracts the attention of

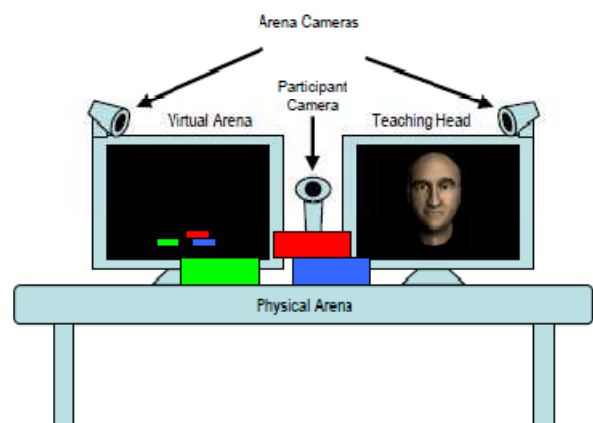


Figure 6. The Teaching Head set up classically has two screens angled at around 120° and three orthogonal webcams. The enclosed physical arena is reproduced in the virtual arena in a monitor or window – we can use a 3D scanner to scan in virtual objects or our 3D printer to print out virtual objects.

Reproduced by permission of DMW Powers and T Lewis

primary school kids, who queue up and chat for hours.

By including some very simple teaching scenarios – where the computer knows the correct answer and what to look for or listen for, it is very easy to adapt this to teaching literacy (helping with correct reading, pronunciation and spelling) and the Head gives us an advantage over other educational software, in providing a teacher-like or peer-like focus that children like to interact with, in being able to monitor what the child is doing without relying on keyboard skills, in being able to sensitively use expression and emphasis to point out gently what is wrong and how to fix it. Numeracy is even easier! A number of demonstration videos are available from the authors for both literacy and numeracy applications.

At the moment, the main interest is in relation to helping indigenous children and remote communities, including extending the system to training re basic health and hygiene issues, as well as training health workers.

4.3.2 MANA – Memory, Appointment and Navigation Assistant

We now move to the other end of the age scale, as people retire and want to retain their ability to live independently. The Thinking Head is being developed as a memory aid and calendar service, given an ability to help in emergencies, and a mobile phone version will help people find their way around town on public transport. Some of these goals are reflected in the Memories for Life (M4L) Grand Challenge of the British Computer Society (<http://www.memoriesforlife.org>), but we also have goals relating to teaching them to retain their memory skills, who their grandchildren are, enhancing their cognitive skills and maintaining their interest in sports and current affairs.

To what extent it will become the “companion” the news reports picked up on remains to be seen!

4.4 The Social Head

Many of the companion and teaching functions have particular applicability to those with disabilities, many of which have an impact on people’s ability to function socially, to learn effectively in standard classes, and to feel a useful part of society. Analogous applications are being defined here – mainly encouraging conversation and good social practice.

But it should also be noted that what is best in this respect may differ from child to child, and these differences may in themselves have diagnostic value. For example, deaf children need to focus on your face more than the conventional norms allow, but ADHD children should not be allowed to be distracted by your lip movements as this has a negative effect on their learning.

4.5 The Instructive Head

The Teaching Head is quite unique in that the linguistic and conversational aspects appeal to students interested in language rich subjects and the social and people aspects of life. The robot world simulation provides opportunity for those interested in mathematics and physics, or computer games, or multimedia and creative arts, to explore their interests by designing worlds that reflect what they want to learn or do. Then there’s the

engineering and biological sides of audio and video, speech and vision processing – there’s something for everyone...

We are now running regular workshops for schools (typically one or two a week, with versions from year 5 to year 11) focussing on one or more aspects of the project – and indeed have developed a full 10 week course for year 10 students based around these topics, allowing students the flexibility to spend more time on the aspects that interest them. Children are finding their conception of science and the idea of interdisciplinary collaboration extremely broadening.

Our focus in this aspect of the project is to address the decline in numbers of students taking mathematics and science subjects, or seeing them as relevant to their futures. We believe we are getting the point across!

5 Human Factors and Fling Search

We have already discussed experiments and results from experiments where we have compared human performance with computer performance (Table 2) or compared human performance with specific variations in experimental conditions relating to a specific aspect of a user interface (Figure 5).

One of the main applications that has emerged as a key one both in terms of commercial interest and evaluation of language technology, as well as for human factors research, is search. We have touched on other applications including speech recognition and synthesis, emotion recognition and expression evaluation, spelling correction and machine translation. We have touched on biometric approaches to study aspects of language and learning. But so far we have not discussed search and it seems appropriate as a case study illustrating further the multimodal aspects of our research.

Our approach to Human Factors is the same as our approach to Language Learning – we don’t want to rely on introspection or theories, and we specifically decry computing researchers and software manufacturers that inflict their own ideas on ideas on others whilst ignoring decades, sometimes centuries, of psychological research. We don’t want to assume we know the answer in discovering grammar rules or syntactic categories, or the best measures or attributes to use in an algorithm or interface, so we can’t use this kind of information for training or evaluation, although we inform ourselves of relevant work from both computer and cognitive science.

In relation to search our human factors/user interface research focuses in two areas, the visual interface and the text interface.

5.1 The Textual Search Interface

Here we are seeking to answer questions about the way people use particular words, how many words they use, how these words relate to the statistics of the documents they see as relevant, and to the commonly used ranking methods, and then of course, whether these characteristics differ for search versus description, or based on experience. The short answer to these last two questions is yes, but to the others the answer is that there is not a good match between human choices and rankings, and those provided by standard algorithms (Pfitzner, 2008).

This research, along with the previously discussed Teaching Head evaluation and the visually oriented work that follows, is being undertaken in computer-delivered experiments that present search, comprehension, tracking or other tasks, and automatically collate the results. Our lean simple system is available in a heavily trafficked lab, and many experiments are also able to be delivered over the web and thus receive additional exposure (Treharne, Pfitzner, Leibbrandt and Powers, 2008).

5.2 The Visual Search Interface

In the visual interface we are seeking to improve websearch by allowing navigation of the web in a very physical way – navigating hyperspace like the Enterprise! Each word or phrase or topic in a document or corpus is a potential dimension. Generally we can reduce the dimensionality by using semantic information such as WordNet or a thesaurus, or by using dimension reduction techniques like Singular Valued Decomposition (or Latent Semantic Indexing as it is known in this context). The choice of such reduction techniques belongs in the text part of the project. But how we display these thousands of dimensions on a 2D or 3D computer screen is another question.

Yes 3D – our human factors system is deployed on a Philips WOW! lenticular screen that shows 9 different views of each pixel, giving a 3D effect. Most of the current attempts at search visualization spend a lot of effort using just the right shades and shadows and reflections and perspective to give a great 3D effect that is totally useless as an interface, whilst many interfaces also waste a lot of screen real estate (Pfitzner, Hobs and Powers 2002).

Our project is controlling, and experimenting on dimension by dimension, each attribute of the domain (words, phrases, sizes, clusters, metadata) versus each attribute of the screen, physical dimension including real stereoscopic depth, versus shading, colour and animation effects, as well as leaving size available as size. This illustrates another point – we want the best mapping possible between screen attributes and domain attributes, as well as the best choices in each case.

The results are not surprising – but they are damning of most current interfaces (Treharne et al., 2008).

6 Acknowledgements

The Thinking Head project is funded under ARC SRI TS0669874, in the context of the Australian Research Council and National Health and Medical Research Council joint Special Research Initiative in Thinking Systems. Australian Partners include the University of Western Sydney, Flinders University, Macquarie University, University of Canberra, Industry Partners include Seeing Machines Ltd, International Partners include Carnegie Mellon University, Berlin University of Technology and Technical University of Denmark.

Development of the German-language Head is supported by the Deutsches Forschungs Gemeinschaft (DFG). Some of our search research has been commercialized by YourAmigo Pty Ltd or undertaken under contract with them, and some is also being pursued under contracts with EducationAU Pty Ltd and the Australian Defence

Science and Research Organization (DSTO). Some of our EEG Learning research was undertaken under contract with DSTO, whilst other BCI/EEG research is being commercialized by Biox Pty Ltd. Our speech control technology was developed in association with I2Net Pty Ltd and the Clipsal Homespeak version of this system is being distributed internationally by Clipsal.

We also acknowledge and appreciate the assistance of the Goethe Institute and numerous schools, teachers and students.

7 References

- Agrawal, R., Imielinski, T. and Swami, A. (1993): Mining association rules between sets of items in large databases. *Proc. ACM SIGMOD International Conference on Mgt of Data*, 22:207-216, ACM Press.
- Brin, Sergey, Motwani, Rajeev, Ullman, Jeffrey D. and Tsur Shalom (1997). Dynamic itemset counting and implication rules for market basket data, *Proc. ACM SIGMOD Int'l Conf. on Mgt of Data*, pp 265-276.
- Deane, Paul D. (1992). *Grammar in Mind and Brain. Explorations in Cognitive Syntax*. Walter de Gruyter.
- Entwisle, Jim (1997) *A constraint parser*, Dept of Computer Science, Flinders University, Adelaide
- Entwisle, Jim and Powers, David M. W. (1997) The Present Use of Statistics in the Evaluation of NLP Parsers, pp215-224, *NeMLaP3/CoNLL98 Joint Conf. on New Methods in Language Processing and Conference on Natural Language Learning*.
- Fayyad, U., Piatetsky-Shapiro, G. and Smyth, P. (1996): From data mining to knowledge discovery: an overview. In *Advances in Knowledge Discovery and Data Mining*. 1-34. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P. And Uthurusamy, R. (eds). AAAI.
- Flach, PA. (2003). The Geometry of ROC Space: Understanding Machine Learning Metrics through ROC Isometrics, *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, Washington DC, 2003, pp. 226-233.
- Fitzgibbon, Sean, Powers, David M W, Pope, Kenneth and Clark, C. Richard (2007). *Removal of EEG noise and artefact using blind source separation*. *Journal of Clinical Neurophysiology* 24(3):232-243
- Fürnkranz Johannes & Peter A. Flach (2005). ROC 'n' Rule Learning – Towards a Better Understanding of Covering Algorithms, *Machine Learning* 58(1):39-77.
- Harnad, S. (1990) *The Symbol Grounding Problem*. *Physica D* 42:335-346
- Homes, David (1997) *Perceptually Grounded Language Learning*, B.Sc. Honours Thesis, Dept of Computer Science, Flinders University, Adelaide.
- Huang, JinHu and Powers, David M W (2004), Adaptive Compression-based Approach for Chinese Pinyin Input, *ACL SIGHAN Workshop*, pp. 24-27
- Huang, Jin Hu and David M W Powers (2001), Large scale experiments on correction of confused words, *Proc. Australian Computer Science Conference (ACSC01)*, pp77-82

- Hume, David (1984) *Creating Interactive Worlds with Multiple Actors*, Computer Science Honours Thesis, EECS, Uni. of NSW, Sydney, AUSTRALIA
- Leibbrandt, Richard, Luerksen, Martin, Matsumoto, Takeshi, Treharne, Kenneth, Lewis, Trent, Li Santi, Martin and Powers, David M W (2008), An Immersive Game-Like Teaching Environment with Simulated Teacher and Hybrid World, *Computer Games and Allied Technology*, pp217-225.
- Leibbrandt, Richard & Powers, David M. W. (2008), Grammatical category induction using lexically-based templates. *Supplement, Boston Univ. Conference on Language Development* **32**, Nov 2-4, 2007, (8pp).
- Leibbrandt, Richard (2008), *Part-of-speech bootstrapping using lexically-specific frames*, PhD thesis, Computer Science, Engineering & Mathematics, Flinders Univ.
- Lewis, T. W. and D. M. W. Powers (2002). Audio-Visual Speech Recognition using Red Exclusion and Neural Networks. *Australian Computer Science Conference*
- Martin H. Luerksen, Lewis, T.W., Leibbrandt, R. and Powers, David M.W. (2008), Adaptive Multimodal Perception for a Virtual Museum Guide. *3rd Wkshp on Artificial Intelligence Techniques for Ambient Intelligence*
- Magerman, David (1991) Distituent Parsing and Grammar Induction, Powers, David M. W. and Larry Reeker, eds. (1991), *Proceedings of the AAAI Spring Symposium on Machine Learning of Natural Language and Ontology*, pp122-125.
- Takeshi Matsumoto, David Powers and Nasser Asgari (2008), Webcam Configurations for Ground Texture Visual Servo, *IEEE International Conference on Cybernetics & Intelligent Systems, Robotics, Automation and Mechatronics (CIS-RAM 2008)*
- Perruchet, P. and Peereman, R. (2004). The exploitation of distributional information in syllable processing, *J. Neurolinguistics* **17**:97–119.
- Piatetsky-Shapiro, G. (1991). Discovery, analysis, and presentation of strong rules. *Knowledge Discovery in Databases*, pp. 229-248.
- Pfitzner, Darius M (2008), *An Investigation into User Text Query and Text Descriptor Construction*, PhD thesis, CSEM, Flinders University.
- Pfitzner, Darius M, Leibbrandt, Richard E, and Powers, David MW (2008) Characterization and Evaluation of Similarity Measures for Pairs of Clusterings, *Knowledge and Information Systems: An Int'l Journal*.
- Darius Pfitzner, Vaughan Hobbs and David Powers (2002), A unified taxonomic framework for information visualization. *Proc. Australian Symposium on Information Visualization*, Adelaide, pp.57-66,.
- Powers, David. M. W., Richard Leibbrandt, Darius Pfitzner, Martin Luerksen, Trent Lewis, Arman Abrahamyan and Kate Stevens (2008), Language Teaching in a Mixed Reality Games Environment, *Proc. 1st International Conference on Pervasive Technologies Related to Assistive Environments (PETRA) Wkshp "Gaming Design and Experience: Design for Engaging Experience and Social Interaction"*
- Powers, David M. W. (2008), Evaluation Evaluation, *Proc. 18th European Conference on Artificial Intelligence (ECAI'08)*, July 21-25, 2008, Patras (2pp).
- Powers, David M. W. (2007), *Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation*, School of Informatics and Engineering, Flinders University • Adelaide • Australia, Tech. Report SIE-07-001, December 2007.
- Powers, David M. W. (2003). Recall and Precision versus the Bookmaker. *International Conference on Cognitive Science*, University of New South Wales, pp.529-534. See <http://david.wardpowers.info/BM/>
- Powers, David M. W. (2001), The Robot Baby meets the Intelligent Room, *AAAI Spring Symposium on Learning Grounded Representations*, pp59-62.
- Powers, David M W (1997a) Learning and Application of Differential Grammars, *CoNLL97: ACL Workshop on Computational Natural Language Learning*, pp88-96.
- Powers, David M. W. (1997b), Unsupervised learning of linguistic structure: an empirical evaluation, *Int'l Journal of Corpus Linguistics* **2**#1:91-131
- Powers, David M W, Clark, C R, Dixon, S E and Weber, D L, Cocktails and Brainwaves: Experiments with Complex and Subliminal Auditory Stimuli, pp68-71, *Proc. of the Australian and New Zealand Conference on Intelligent Information Systems*, IEEE 96TH8234
- Powers, David M W (1991), How far can self-organization go? Results in unsupervised language learning. *Proc. AAAI Spring Symposium on Machine Learning of Natural Language and Ontology*, pp131-136.
- Powers, David M. W. and Larry Reeker, eds. (1991), *Proceedings of the AAAI Spring Symposium on Machine Learning of Natural Language and Ontology*, Document D-91-09 (205pp), DFKI, Univ. Kaiserslautern FRG.
- Powers, David M. W. (1983), Neurolinguistics and Psycholinguistics as a Basis for Computer Acquisition of Natural Language" *SIGART* **84**, pp. 29-34.
- Shanks, D. R. (1995). Is human learning rational? *Quarterly Journal of Experimental Psychology*, **48A**, pp257-279.
- Treharne, Kenneth, Pfitzner, Darius, Leibbrandt, Richard & David M. W. Powers (2008), A Lean Online Approach to Human Factors Research, *Proc. 1st International Conference on Pervasive Technologies Related to Assistive Environments (PETRA) workshop on "Pervasive Technologies in e/m-Learning and Internet based Experiments"* (PTLIE)
- Yang, Dongqiang and Powers (2008), Automatic Thesaurus Construction, *Australasian Computer Science Conference (ACSC2008)*, pp147-156.
- Yang, Dongqiang and Powers, David M W (2006a), Word sense disambiguation using lexical cohesion in the context. *Proc. Joint Conf. of the Inter'al Committee on Comp. Ling. and the Assn for Comp. Ling. (COLING-ACL 2006)*, Sydney Aust, pp929-936.
- Yang, Dongqiang and Powers, David M W (2006b), Verb similarity on the taxonomy of WordNet, *Proc. Third International WordNet Conference (GWC-06)*, Jeju Island, Korea. pp121-128
- Yang, Dongqiang and Powers, David M W (2005), Measuring Semantic Similarity in the Taxonomy of WordNet, *ACSC'05 Australasian Computer Science Conference*. pp315-322

Multi-Strategy Ensemble Learning, Ensembles of Bayesian Classifiers, and the Problem of False Discoveries

Geoff Webb

Clayton School of Information Technology
P.O. Box 63
Monash University,
Victoria 3800, Australia

This talk covers an ensemble of my research contributions that I believe are likely to resonate with a current audience.

Ensemble Learning combines the predictions of multiple classifiers to enhance accuracy relative to any individual classifier. I will show that combining established ensemble learning techniques further enhances accuracy without computational overhead.

Naive Bayes is a popular approach to classification learning due to its computational efficiency, strong theoretical foundation and its capacity to predict probabilities rather than just the most probable outcome. I will present a simple extension that creates an ensemble of naive-Bayes like classifiers, improving naive Bayes' accuracy without undue computational burden.

Finally, I will discuss false discoveries, a problem that plagues many modern pattern discovery systems. Quite simply, many state-of-the-art approaches to pattern discovery are prone to 'discover' patterns that do not exist. I will explain why this is so and discuss approaches to overcome the problem.

Volume, Velocity and Variety - Key Challenges for Mining Large Volumes of Multimedia Information

Richard Price

Defence Science and Technology Organisation
PO Box 1500
Edinburgh, South Australia 5111

New challenges are emerging, as both government and commercial organisations attempt to exploit the potentially important information in their ever increasing volumes of collected data.

This presentation will focus on some of the major challenges involved in the processing and analysis of large multimedia databases. The presentation will present and discuss a range of data mining and visual analytic tools and techniques that DSTO have either developed or acquired to assist organisations uncover potentially interesting patterns of behaviours, trends, links and associations that exist in their data.

CONTRIBUTED PAPERS

On Inconsistencies in Quantifying Strength of Community Structures

Wen Haw Chong

DSO National Laboratories
20 Science Park Drive Singapore 118230

cwenhaw@dso.org.sg

Abstract

Complex network analysis involves the study of the properties of various real world networks. In this broad field, research on community structures forms an important sub area. The strength of community structure is typically quantified by the modularity measure. The measure is based on summing the differences in actual and expected fraction of edges per community (across all communities in the network), whereby the latter is computed based on randomizing the edges subjected to certain constraints. In this paper, we investigate the differences between two commonly used definitions of modularity and highlight one of them as inadequate for quantifying the strength of community structures. We first show this by mathematical proving. We then investigate the empirical differences by developing and testing two variants of a community detection algorithm whereby the variants differ based on their modularity definitions. We observe varying differences in detection accuracy when applying the variants on artificially generated networks. For networks with strong community structures, we show that sensible results are still obtainable with the inadequate measure, which explains why this issue did not come to light previously.

Keywords: community structure, networks, modularity.

1 Introduction

In the real world, a variety of networks exist and their statistical, mechanical and temporal properties are the subject of much research, known broadly as complex network analysis. An important sub-area in the study of complex networks is that of community detection. In various real world networks, communities are of practical interest. For example, communities may be indicative of social groups or cliques in a network of relationships. They may also be indicative of topics of common interest as in the case of the world-wide web (Kleinberg, and Lawrence 2001). In the case of biochemical networks, they may correspond to functional units (Ravasz, et al. 2002). Other examples of community structures can be found in collaboration networks, computer networks and food webs, etc.

Formally, a community is defined as a group which has more than the expected number of links between its members, whereby 'expected' means by random chance. Hence connections within communities are expected to be much denser than between the communities. To quantify the strength of community structures, a measure known as modularity was first proposed by Newman and Girvan (2004). This remains a successful and widely used measure. The modularity measure quantifies the idea that community structures should result in fewer than expected number of inter-community edges and conversely, higher than expected number of intra-community edges in the network. Many recently developed community detection algorithms thus focus on deriving groupings that maximizes modularity. The general consensus was that the best algorithm returns the maximum modularity on real world networks. The problem itself is an NP hard problem with ongoing active research in more efficient and accurate algorithms.

In the next section, we shall describe the commonly used definitions of modularity. Section 3 analyse one of the definitions mathematically. We derive the algorithm optimizing each definition in section 4. Section 5 presents the experimental measure used while section 6 presents the experiments. Finally we conclude in section 7.

2 Modularity Definitions

Given a network with n nodes and m edges (or links), let \mathbf{A} be the corresponding adjacency matrix, whose element A_{ij} is the edge weight between nodes i and j . This is either 1 or 0 for unweighted networks and may be of any other values for weighted networks. Let k_i be the degree of node i . We further denote the community containing node i as its parent community c_i .

We bring the reader's attention to two definitions of modularity commonly occurring in the literature, which we denote as $Q1$ and $Q2$. To avoid any ambiguity, we also list all related and alternative forms that are equivalent for each definition. The first form (Clauset, et al. 2004; Newman, 2006a, 2006b; Fortunato, and Barthélemy 2007) assumes that the expected number of edges between nodes i and j if edges are placed at random is $k_i k_j / 2m$. Modularity is then quantified by the difference between the actual and expected number of edges:

$$Q1 = \frac{1}{2m} \sum_{ij} [A_{ij} - k_i k_j / 2m] \delta(c_i, c_j) \quad (1)$$

where c_i and c_j are the parent communities of nodes i and j respectively and $\delta(c_i, c_j)$ is 1 if c_i equals c_j and is 0 otherwise. If the network has been divided into p communities, then a symmetric $p \times p$ matrix \mathbf{e} can be

defined whose element e_{vw} is the fraction of all edges that link nodes in community v to nodes in community w . The diagonal elements e_{vv} is then simply the fraction of edges falling within each community. $Q1$ has been shown (Clauset, et al. 2004) to be equivalent to

$$\sum_v (e_{vv} - (d_v / 2m)^2) \quad (2)$$

where d_v is the sum of degrees of nodes in community v . Hence $Q1$ can be alternatively computed based on summation of terms over communities, instead of considering node pairs. This is very similar in form to the other modularity definition to be introduced.

The alternative and also commonly used definition for modularity (Newman, and Girvan 2004; Newman 2004; Duch, and Arenas 2005; Ruan, and Zhang 2006) is:

$$Q2 = \sum_v (e_{vv} - a_v^2) \quad (3)$$

where $a_v = \sum_w e_{vw}$ is the fraction of edges which has at least one end in community v . In matrix notation, $Q2$ can be written as $Tr(e) - \|e^2\|$ where $Tr(\cdot)$ is the trace of the matrix and $\|x\|$ indicates the sum of the elements of matrix x . It has generally been assumed that this definition is roughly equivalent to the first, with the same null model, i.e. if there is no community structure, the number of expected links between two nodes are proportional to the product of their degrees. However on closer examination, it is found that the formula does not follow this assumption.

3 Analysis

In this section, we analyse the measure $Q2$ mathematically for an unweighted network. For node i in community c_i , we can divide its edges into those that link to other nodes in the same community, i.e. the inner degrees $k_{i,in}$ and those that link to nodes outside the community, i.e. the outer degrees $k_{i,out}$. It is also obvious that $k_i = k_{i,in} + k_{i,out}$. From inspection, given any community (indexed by v), we can write the number of edges with 1 or both ends in it as

$$\sum_i (k_i - 0.5k_{i,in}) \delta(c_i, v). \quad (4)$$

This is equivalent to summing the degrees of all nodes in the community with the inner degrees weighted by half. Correspondingly,

$$a_v = \frac{1}{m} \sum_i (k_i - 0.5k_{i,in}) \delta(c_i, v) \quad (5)$$

Substituting equation (5) into the definition for $Q2$, we have

$$\begin{aligned} \sum_v (e_{vv} - a_v^2) &= \sum_v \left[\frac{1}{2m} \sum_{ij} A_{ij} \delta(c_i, v) \delta(c_j, v) \right. \\ &\quad \left. - \frac{1}{m} \sum_i (k_i - 0.5k_{i,in}) \delta(c_i, v) \frac{1}{m} \sum_j (k_j - 0.5k_{j,in}) \delta(c_j, v) \right] \\ &= \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{2}{m} (k_i - 0.5k_{i,in})(k_j - 0.5k_{j,in}) \right] \delta(c_i, c_j) \end{aligned} \quad (6)$$

where we have made use of the fact that $\sum_v \delta(c_i, v) \delta(c_j, v) = \delta(c_i, c_j)$.

Hence the second term that is being subtracted and which supposedly corresponds to the null model has undesired terms dependent on the community structures, specifically the inner degrees of each community. This departs from the null model. Nonetheless, if the network of application has strong community structures, we shall see that the communities may still be detected satisfactorily by maximizing $Q2$. This is probably why the inadequacies of $Q2$ were not detected and highlighted earlier in the literature. Another reason is that community detection algorithms are typically applied on real world networks and compared on the basis of their modularity values according to one definition. For the issue to manifest itself, this will require comparing two versions of the same algorithm, one optimizing each modularity definition. In our experiments, we adopt this approach.

4 Algorithm

We select a current community detection algorithm that also proves easy to customize for each modularity definition. Blondel et al. (2008) has proposed an algorithm that they have shown to be extremely fast and feasible for application on networks of up to a billion nodes, constrained only by the amount of memory available for computation. The algorithm is readily applicable on both weighted and unweighted networks. We derive two variants of the algorithm corresponding to $Q1$ and $Q2$.

The algorithm starts with each node defined as a community. The steps are as follows:

1. For each node i , move it out of its own community to its neighbor's community where modularity gain is positive and maximum. If no positive gain is possible, node i remains in its community.
2. Repeat step 1 over all nodes multiple times until modularity has converged to its maximum value.
3. Construct a new network whose nodes now represent the communities. The links between the new nodes are calculated as the sum of the link weights between nodes in the corresponding communities.

Repeat the whole process until there are no possible node movements (in step 1) which will increase modularity. At this stage, the algorithm terminates.

Note that in this algorithm, there is no merging of communities, only inter-community movements of nodes. Communities which end up with no nodes are then dropped from subsequent processing. This differs from earlier agglomerative merging approaches (Clauset, Newman and Moore 2004). The two variants described in this paper differ from each other only in their modularity gain formula in step 1. Both our variants also differ slightly from the original algorithm¹ described by Blondel

¹ It was not clear which modularity definition was optimized. The authors have indicated clarification of their formula in subsequent versions of the paper.

et al. (2008) in the sense that the modularity gain formula in step 1 is different.

4.1 Variant 1

Variant 1 optimizes $Q1$. We show the derivations for unweighted networks. It can easily be generalized to weighted networks. In calculating the modularity gain formula, we need to consider modularity changes both when a node leaves its community and when it joins another community. Let Q_{leave} be the modularity change for a node leaving its community and Q_{join} be the modularity change when the same node joins another community. Then for a single node movement, $\Delta Q1 = Q_{join} - Q_{leave}$.

Consider the case when a node i leaves its current community w to join another community v . In the joining process, new summation terms arise in the equation for $Q1$ to constitute the change in modularity. This can be calculated as

$$\begin{aligned} Q_{join} &= 2 \times \frac{1}{2m} \sum_j (A_{ij} - k_i k_j / 2m) \delta(c_j, v) \\ &= \frac{1}{m} \sum_j A_{ij} \delta(c_j, v) - \frac{2}{(2m)^2} \sum_j k_i k_j \delta(c_j, v) \\ &= \frac{k_{i,in}}{m} - \frac{2k_i}{(2m)^2} \left(\sum_j k_{j,in} \delta(c_j, v) + \sum_j k_{j,out} \delta(c_j, v) \right) \\ &= \frac{k_{i,in}}{m} - \frac{k_i \sum_{in}}{m^2} - \frac{2k_i \sum_{out}}{(2m)^2} \end{aligned} \quad (7)$$

where \sum_{in} is the sum of weights of links inside v and \sum_{out} is the sum of weights of links incident to nodes in v . Similarly, during the leaving process, certain terms drop off from the summation for $Q1$ as follows.

$$Q_{leave} = \frac{k_{i,in'}}{m} - \frac{2k_i}{(2m)^2} \sum_{j \neq i} k_j \delta(c_j, w) \quad (8)$$

where $k_{i,in'}$ is the inner degree of node i in community w .

4.2 Variant 2

For variant 2, $Q2$ is maximized. For the same leaving and joining scenario, we simply need to update quantities corresponding to the two affected communities, i.e. e_{vv} , a_v , e_{ww} and a_w . Let the updated quantities after the node movement be tagged with a subscript $*$. Then the modularity gain can be easily calculated as

$$\begin{aligned} \Delta Q2 &= e_{vv*} - a_{v*} + e_{ww*} - a_{w*} \\ &\quad - (e_{vv} - a_v + e_{ww} - a_w) \end{aligned} \quad (9)$$

5 Distance Measure

We use an entropy based measure, the Variation of Information (VI) to quantify the distances between two sets of partitions, which in this case are the sets of actual and detected communities. This measure is a true metric on the space of community assignments with all the properties of a proper distance measure. The variation of information has also been advocated by Karrer et al. (2008) for measuring the distance between two community sets.

Formally, given two different partitioning of the network, C with p communities and C' with p'

communities, let there be n_v nodes in the v th community of partition C and n'_w nodes in the w th community of C' . Denote the size of the intersection, i.e. the number of nodes that are common to both communities as n_{vw} . The VI distance is then defined as

$$d_{VI}(C, C') = H(C) + H(C') - 2I(C, C') \quad (10)$$

where the entropy of community set C is:

$$H(C) = - \sum_{v=1}^p \frac{n_v}{n} \log \frac{n_v}{n} \quad (11)$$

and the mutual information between the two sets of communities is:

$$I(C, C') = \sum_{v=1}^p \sum_{w'=1}^{p'} \frac{n_{vw'}}{n} \log \frac{n_{vw'}}{n} \frac{n_v}{n_v} \frac{n_{w'}}{n_{w'}} \quad (12)$$

The maximum VI distance achievable between two community sets is $\log n$ which happens when one assignment places all nodes in one community and the other places each node individually in a community of its own. Accordingly the VI distance can be normalized to between 0 and 1 via the following:

$$d_{VI}(C, C') = d'_{VI}(C, C') / \log n \quad (13)$$

In all our experiments, we have tabulated the $d_{VI}(C, C')$ values. The smaller $d_{VI}(C, C')$ is, the closer the match between the detected and actual communities, and the more accurate the community detection algorithm is. For identical match, $d_{VI}(C, C') = 0$.

6 Experiments

It is not straight forward to compare the variants on real world networks. Due to the absence of a ground truth partition for comparing the derived communities against, it is difficult to gauge the quality of the partition found. An easier approach is to apply the variants on artificial networks where we know the actual partition.

We apply both algorithm variants on artificial networks, such that we can systematically vary the strength of the community structure. We generate unweighted networks of 200 nodes with 4 communities of 50 nodes each. Edges are placed probabilistically between pairs of nodes by considering whether they are in the same or different communities.

Let the probability of connection between two nodes in the same community be pin . For nodes in different communities, let the connection probability be $pout$. For community structures to be evident, pin should be higher than $pout$ to obtain more intra-community edges than inter-community edges. If the difference is large, then there is a strong community structure. Also pin can be varied to obtain weakly or strongly connected communities.

We have tested pin values of 0.2 to 0.6 in steps of 0.1. For each pin value, we let $pout = factor \times pin$ and vary $factor$ from 0.1 to 0.5 in steps of 0.1. This provides a systematic way to vary the strength of community structures from strong to weak.

For each combination of pin and $pout$ values, we generate 5 networks for trials. We have observed that for the range of pin values tested, detection accuracy behaves in the same manner and generally decreases as $pout$ is

increased. Hence it suffices to examine the boundary cases corresponding to the strongest and weakest connected communities. In table 1, we display the mean and standard deviations of the normalized VI distances between the actual and derived communities (over 5 trials for each combination) for the cases of $pin = 0.2$ and $pin = 0.6$. The means are plotted in figure 1 for easy comparison. For comparison purpose, we have also included the results from the leading eigenvector (EV) method (with no fine tuning) for community detection. This algorithm was proposed by Newman (2006b) and maximizes $Q1$.

$pin = 0.2$			
$pout$	Variant 1	Variant 2	EV method
0.02	0.048 (0.028)	0.047 (0.029)	0.145 (0.090)
0.04	0.070 (0.047)	0.288 (0.035)	0.155 (0.061)
0.06	0.191 (0.047)	0.567 (0.041)	0.271 (0.014)
0.08	0.262 (0)	0.643 (0.030)	0.298 (0.021)
0.1	0.262 (0)	0.704 (0.031)	0.336 (0.018)
$pin = 0.6$			
$pout$	Variant 1	Variant 2	EV method
0.06	0 (0)	0 (0)	0.010 (0.015)
0.12	0 (0)	0.14 (0.099)	0.055 (0.042)
0.18	0 (0)	0.63 (0.016)	0.093 (0.070)
0.24	0 (0)	0.709 (0.016)	0.207 (0.047)
0.3	0.026 (0.023)	0.738 (0.001)	0.188 (0.029)

Table 1: Average VI distances (normalized) between actual and derived communities. Each value is the average over 5 trials. Standard deviations are bracketed.

As expected, for all algorithms, detection accuracy declines as the strength of community structure decreases. Variant 2 proves to be the least robust of all and the VI distances between its detected communities and the actual communities increase rapidly with $pout$. This simply means that detection accuracy is not robust. However when there are strong community structures, e.g. when $pout = 0.1 \times pin$, corresponding to the left most points on the graphs, it performs identically to variant 1 in detecting the exact communities. Variant 1 is the most accurate in returning the actual communities and slightly outperforms the leading eigenvector method. Results from the latter can be improved with fine tuning node memberships of the individual communities although this will be extremely expensive for large networks. For $pin = 0.6$ where communities are strongly connected, community detection is perfect in most cases for variant 1, as reflected by the zero VI distances.

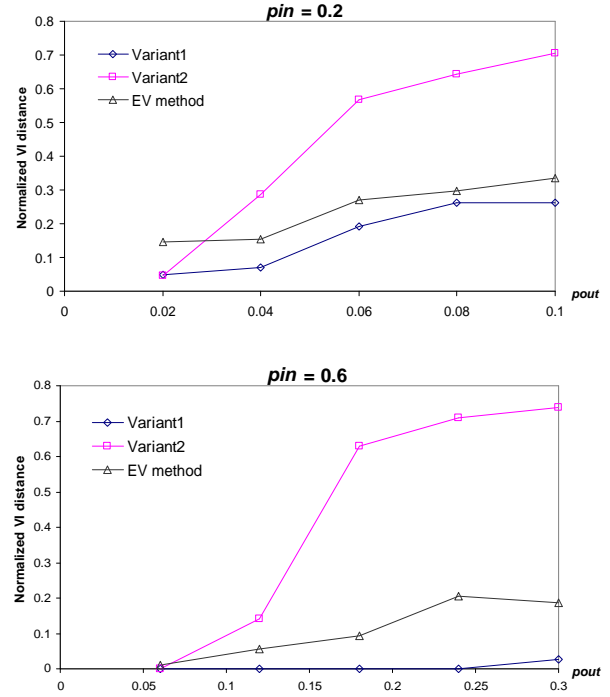


Figure 1: Average VI distances (normalized) from table 1.

This simple experiment is sufficient to demonstrate that $Q2$ is inadequate as a maximization criterion, although it may perform satisfactorily in some cases. This is also evident when we apply both variants on real world networks. We have tested them on several real world networks, namely the “karate club” network from Zachary (1977), the dolphin network from Lusseau (2003) and a network of books about politics (Krebs). It was observed that both variants can result in very similar detected sets of communities. The differences between them can once again be quantified with VI distance.

Networks	Hierarchy levels		
	1	2	3
Karate	0.095	0.070	0
Dolphin	0.105	0.058	0
Book	0.211	0.020	0

Table 2: VI distances (normalized) between community sets of both variants for each hierarchy level and for each network.

The VI distances between the detected communities from both variants are illustrated in table 2 for the mentioned real world networks. For each network tested here, both variants return hierarchies with three levels. We denote the lowest hierarchy level as level 1 and the highest level as level 3. The latter corresponds to the case where all nodes are considered as one community. Hence the hierarchies have a one to one correspondence in levels and can be compared in a straightforward manner. At level 1, detected communities appear significantly different for Krebs’ book network, although at level 2, both variants again return very similar results. For the

other networks at various levels, detected communities differ only slightly. This simply means that although $Q2$ is not an appropriate measure, it may not be very obvious empirically.

For a rough idea of what the figures entail, figure 2 illustrates the karate network and resulting partitions from both variants. It can be seen that partitions of both variants differ only slightly at each hierarchy level. At level 1, although variant 1 indicates seven communities and variant 2 indicates six, the general community structures are quite similar. Two of the communities from variant 1 are essentially “merged” as one in variant 2. At level 2, both variants return three communities with the only difference arising in the membership of one node (numbered 10).

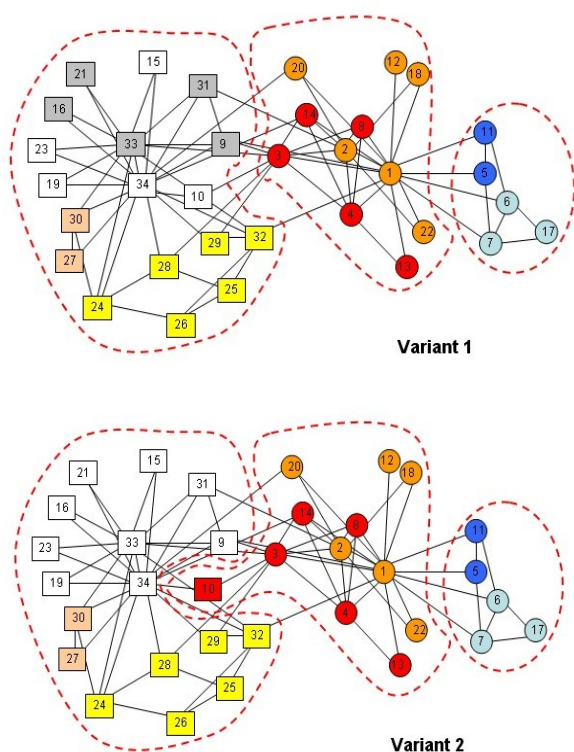


Figure 2: The karate network and partitions returned by the 2 variants. Numbered vertices represent club members while edges represent friendships. The squares and circles represent the actual two factions of the club. Level 1 partitions are colour coded while level 2 partitions are circled.

7 Conclusion

In this paper, we have identified the inconsistencies of a modularity definition that was used frequently to characterize community structures. Although it is fairly straightforward to prove this mathematically, the issue surprisingly has not been highlighted. We also demonstrate this empirically by testing variants of the same algorithm specially customized for each modularity definition. The take away message is that an imperfect measure may still give rise to sensible results on certain

data sets and caution is necessary in the process of defining a measure.

Incidentally, the variant 1 algorithm has performed very accurately in the experiment here and it may be worthwhile to investigate it in experiments with more extensive scenarios, such as using artificial communities of different sizes or incorporating hierarchies. More community detection algorithms in addition to the leading eigenvector algorithm can be compared with it.

While we have shown $Q1$ to be the better modularity definition, it has its limitation in the form of the recently discovered “resolution limit” issue, which was highlighted by Fortunato and Barthélemy (2007). Essentially, algorithms which maximise $Q1$ may have less power to detect smaller communities in large networks. Under certain conditions, it can be shown that groupings which aggregate the smaller communities into larger one(s) can result in larger modularity values than if they were detected individually. It remains for a measure to be discovered that can satisfactorily quantify the strength of community structures without the resolution limit effects. However the resolution limit issue can be easily circumvented by clever algorithm design. For example, the algorithm utilized here provides the user with a hierarchy of community structures at different resolution levels, as was also pointed out by Blondel et al. (2008). This is perhaps a more preferable approach given that many real world networks contain hierarchical community structures.

8 References

- Blondel, V., Guillaume, J-L., Lambiotte, R. and Lefebvre, E. (2008): Fast unfolding of community hierarchies in large networks. arXiv:0803.0476.
- Clauset, A., Newman, M. and Moore, C. (2004): Finding community structure in very large networks. *Phys. Review E* **70**:066111.
- Duch, J. and Arenas, A. (2005): Community detection in complex networks using Extremal Optimization. *Phys. Review E* **72**:027104.
- Fortunato, S. and Barthélemy, M. (2007): Resolution limit in community detection, *Proc. Natl. Acad. Sci. USA* **104**(1):36-41.
- Karrer, B., Levina, E. and Newman, M. (2008): Robustness of community structure in networks. *Phys. Review E* **77**:046119.
- Kleinberg, J. and Lawrence S. (2001): The structure of the web. *Science* **294**(5548):1849-1850.
- Krebs, V.: <http://www.orgnet.com>.
- Lusseau, D. (2003): The emergent properties of a dolphin social network. *Proc. R. Soc. London B (suppl.)* **270**, S186-S188.
- Meila, M. (2007): Comparing clusterings – an information based distance. *Journal of Multivariate Analysis* **98**, 873-895.
- Newman, M. (2004): Fast algorithm for detecting community structure in networks. *Phys. Review E* **69**:066133.

- Newman, M. (2006a): Modularity and community structure in networks. *Proc. Natl. Acad. Sci.*, vol. **4**:8577-8582.
- Newman, M. (2006b): Finding community structure in networks using the eigenvectors of matrices. *Phys. Review E* **74**:036104.
- Newman, M. and Girvan, M. (2004): Finding and evaluating community structure in networks. *Phys. Review E* **69**:026113.
- Ravasz, E., Somera A., Mongru, D. and Barabási A.-L. (2002): Hierarchical Organization of Modularity in Metabolic Networks. *Science* **297**(5586):1551-1555.
- Ruan, J. and Zhang, W. (2006): Identification and evaluation of weak community structures in networks. *Proc. of National Conference on Artificial Intelligence*, Boston, MA, July 2006, AAAI-06.
- Zachary, W. (1977): An information flow model for conflict and fission in small groups. *Journal of Anthropological Research* **33**, 452-473.

A New Evaluation Measure for Imbalanced Datasets

Cheng G. Weng

Josiah Poon

School of Information Technologies,
J12, University of Sydney,
Sydney, NSW, Australia 2006,
Email: {cheng, josiah}@it.usyd.edu.au

Abstract

The area of imbalanced datasets is still relatively new, and it is known that the use of overall accuracy is not an appropriate evaluation measure for imbalanced datasets, because of the dominating effect of the majority class. Although, researchers have tried other existing measurements, but there is still no single evaluation measure that work well with imbalanced dataset. In this paper, we introduce a novel measure as a better alternative for evaluating imbalanced dataset. We provide a theoretical background for the new evaluation technique that is designed to cope with cost biases, which changes the previous view about class independent evaluation methods cannot deal with costs, such as ROC curves. We also provide a general guideline for the ideal baseline performance when building classifiers with a known misclassification cost.

Keywords: Evaluation, Imbalanced Datasets, ROC and Cost Sensitive Learning

1 Introduction

Since the workshop at AAAI 2000 (Provost (2000)), the imbalanced dataset problem has received an increasing attention for the past few years. This area of research focuses on datasets with skewed class distribution and that the minority class is the class of interest. Imbalanced datasets can occur in many domains, such as medical, information technology, biology, and finance. With imbalanced datasets, the conventional way of maximizing overall performance will often fail to learn anything useful about the minority class, because of the dominating effect of the majority class. Consider a problem where 99% of the data belongs to one class, and only 1% is rare class examples. A learner can probably achieve 99% accuracy with ease, but still fail to correctly classify any rare examples. Conventional approaches can produce misleading results on imbalanced dataset, so it is important to know that one needs to take a more localized approach at all levels when dealing with an imbalanced dataset.

Evaluation is the key to making advances in data mining, and it is especially important when the area is still at the early stage of its development. Imbalanced dataset community has criticized the use of non-class independent evaluation measures, such as the overall accuracy, for reporting experimental results on im-

Actual Class	Predicted class	
	+ve	-ve
	+ve	-ve
Actual Class	+ve	True Positive(TP)
	-ve	False Positive(FP)
Actual Class	+ve	False Negative(FN)
	-ve	True Negative(TN)

Table 1: Confusion matrix for a 2-class problem

balanced datasets. The non-class independent evaluation fails because the results only reflect the learning performance of the majority class, and the more skewed the class distribution is the worse the effect will be. Therefore, when we evaluate the performance on imbalanced datasets, we want to focus on individual classes.

There are many evaluation measures in data mining, some of the most relevant ones to imbalanced datasets are: precision, recall, F-measure, Receiver Operating Characteristic (ROC) curve, cost curve and precision-recall curve. The commonality shared between them is that they are all class independent measurements. In particular, ROC curve is well received in the imbalanced dataset community and it is becoming the standard evaluation method in the area. Provost et al. (1998) have argued that reporting accuracy can be misleading, but ROC curve can help explore different tradeoff among different classifiers over a range of operating conditions. However, using ROC curve is hard to compare different classifiers for different misclassification cost and class distributions.

In this paper, we demonstrate a generalize form, base on different cost bias, of computing the area under ROC curve (AUC), which we will refer to as weighted-AUC. We will describe the related evaluation measurements for imbalanced dataset in related works and present the details of the weighted-AUC in section 3. We show that weighted-AUC is a better alternative when evaluating imbalanced datasets. In section 4, we will define what the ideal baseline performance should be when the misclassification cost is given. Next, we present some experimental results comparing the normal AUC and weighted AUC. Finally, we will discuss some issues related to weighted-AUC in the discussion section.

In the rest of the paper, minority or positive may be used to refer to the rare class and majority or negative for reference to the common class. The examples in this paper will be restricted to two-class problems.

2 Related Works

In imbalanced datasets, not only is the class distribution is skewed, the misclassification cost is often uneven too. The minority class examples are often more important than the majority class examples. The cost of misclassify a minority class example is

2(a)	Predicted class			2(b)	Predicted class		
		+ve	-ve			+ve	-ve
Actual	+ve	0	1	Actual	+ve	0	5
Class	-ve	1	0	Class	-ve	1	0
(a) Equal cost case				(b) Uneven cost case			

Table 2: Cost Matrix Examples

far greater than misclassify a majority class example, for example, fraud detection or cancer diagnosis. We will briefly discuss some relevant evaluation measurements, starting with confusion matrix, which is closely related to many evaluation techniques and can be found in most data mining textbooks. Table 1 shows a confusion matrix for outcome of a two class problem. As an example, using this table, we can define the overall accuracy as $\frac{TP+TN}{TP+FP+TN+FN}$. Confusion matrix is useful when accessing the performance without taking cost into consideration. It is used as a basis for various measures, such as precision and recall.

Precision, recall and F-measure In information retrieval, if there is Z number of relevant documents in the database, and a system can returns X number of documents in which only Y number of them are relevant. We can compare different system performance with precision and recall, which are defined as eq.1 and 2. They can also be defined using confusion matrix terms.

$$Precision = \frac{X}{Y} \text{ or } \frac{TP}{TP + FP} \quad (1)$$

$$Recall = \frac{Y}{Z} \text{ or } \frac{TP}{TP + FN} \quad (2)$$

F-measure is a common evaluation metrics that combines precision and recall into a single value, usually with equal weighting on both measures.

$$F - measure = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (3)$$

Cost matrix Sometimes, the costs are known for the problem at hand, i.e. the misclassification cost of a positive or negative example. In this case, we can use the known cost to penalize the resulting confusion matrix to arrive at a meaningful performance assessment. A cost matrix looks the same as a confusion matrix, except it show the cost of misclassification. Table 2(a) shown an general equal cost case, where 2(b) shows an uneven cost matrix of 1 to 5, which is quite normal for imbalanced datasets. When evaluating, the cost is multiplied on top of a confusion matrix to reflect the true performance.

The disadvantage of using confusion matrix-based evaluation is that they are only looking at the performance on a “spot”, which means we cannot tell how different class distribution or different cost will affect the performance. So researchers may prefer to visually see the performance over a range of situations using one of the graphical evaluation tools, such as a ROC curve.

ROC and AUC Receiver Operating Characteristic (ROC) curve is a common evaluation technique, originated from radio signal analysis (Green and Swets (1966)) and introduced to machine learning by Spackman (1989) and made popular in data mining by Provost and Fawcett (1997). ROC curve presents the tradeoff between the true positive rate

and the false positive rate; as a learner captures more positive example, it will generally misclassify more negative examples as positive examples. If we have a two class problem, a ROC curve can be plotted by varying the probability threshold, from 0 to 1, for predicting positive examples. The true positive rate is the same as recall and the false positive rate equals to $\frac{FP}{FP+TN}$.

A ROC curve is consider to be good if it is closer to the top left corner, and the straight line connecting (0,0) and (1,1) represents a random classifier with even odds. The advantage of use ROC is that one can visually see for what region a model is more superior compare to another.

The area under the ROC curve (AUC) is often used to summaries a learner’s performance into a single quantity, which represents the performance of a learner in general across different prediction cost, and the larger the AUC the better. However, ROC-based measures still lack support for tasks involving uneven cost matrix. In addition, it is interesting to know that all the confusion matrix-based measures can be seen as a single point on the ROC curve.

Cost curve Cost curve was introduced by Drummond and Holte (2000), and they have also provided a detailed comparison between ROC curve and cost curve in Drummond and Holte (2004). Basically, cost curve looks at how classifiers perform across a range of different misclassification cost. It can be seen as different slope line tangent to the ROC curve, therefore every ROC curve has a corresponding cost curve. This view of a slope line bears similarity to the discussion about baseline performance in section 4.

Precision and recall (PR) curve Information retrieval experts use PR curve in similar fashion as ROC curve – except that because the axes are different and the ideal classifier is toward the upper right. An example is shown in figure 7. Davis and Goadrich (2006) provided a comparison between PR curve and ROC curve.

All the graphical evaluation tools provide different perspectives to access a learner’s performance, and the advantage is to have a better understanding of the learner’s behavior under a range of circumstances.

However, the measurements described in this section are designed to work well for different purpose; for example, ROC is well suited for ranking problems with a goal to achieve best resources utilization, whereas cost curve is suitable to locate best method for certain operating cost constraint. Therefore, we realized there is a need to design an appropriate evaluation method for dealing with imbalanced dataset, hence we propose a general approach that can take cost bias into account at evaluation

3 Weighted AUC

An imbalanced dataset often has different cost matrix than a balanced dataset, this is generally due to the nature of the imbalanced dataset, e.g. detection oil spill on satellite images, scanning for scam websites, or detecting cancer. The misclassification of a rare

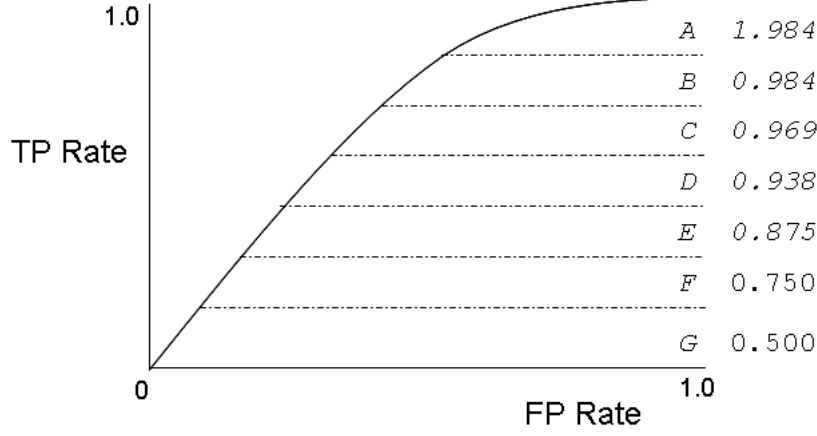


Figure 2: Weighted-AUC example: new weight vector after 50% weight transfer

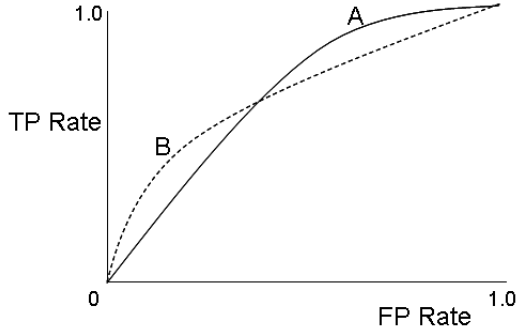


Figure 1: Compare classifiers with ROC curve when conventional AUC is the same.

occurring case in these situations can be very costly, much more expensive than a false alarm. Hence, there is a tendency for the imbalanced datasets to bias its cost towards positive examples. Therefore, one may require the learners to achieve a certain performance level by presetting the cost matrix. As an example, say detecting cancers from x-ray scans, the learners built must able to achieve a standard recall rate in order to be declare useful, otherwise it will not be used no matter how high the precision is. So if the target recall is 90%, then comparison of performances should only consider the area above the 90% TP rate on the ROC curve. In section 4, we provide more detail discussion about the relationship between cost matrix, class distribution and baseline performance.

The rationale for the weighted-AUC measurement is based on the notion that when one is dealing with imbalanced datasets, a learner that performs well in the higher TP rate region is preferred over ones that does not. In another word, a false negative is worse than a false positive. So, in the ideal case, the learner should be able to catch every positive example, i.e. 100% recall/TP rate. With this ideal in mind, if one wants a 100% recall learner, then the best choice will be the one that has the lowest FP rate at the 100% TP rate line.

Generally, where the cost is all equal and two learners have the same AUC, as in figure 1, one can not say classifier A is better than classifier B. However, in an imbalanced dataset situation, one can visually see that classifier A is more appropriate, because at higher TP rate region classifier A has smaller FP rate than classifier B. So, classifier A is preferred over classifier B even though they have the same AUC value. Therefore, the problem with conventional AUC is that it does not consider cost bias, because we sum

up the areas with equal weights of 1, which is a fair assumption when we have equal cost.

We propose a skewed weight distribution method that allows one to compute AUC with a cost bias. We refer to this approach as weighted-AUC. When the cost is uneven and biased towards the rare class, instead of summing up areas with equal weights, we want to give more weights to the areas near the top of the graph. So we create a skew weight vector by distributing more weights towards the top of the ROC curve, while keeping the total weights unchanged. The idea is to pass certain percentage of weights from the bottom areas towards the upper areas of the ROC curve. In figure 2, we present the resulting weight vector after a 50% weight shift is performed. Originally all weights were 1, but now 50% of area G is passed to area F, and 50% of the new area F's weights is again passed to area E and so on. The weight shifting process stops at the top, area A, which has the most weight. We can define the new weights more formally with eq 4. If we have N number of areas to sum:

$$W(x) = \begin{cases} \alpha, & x = 0 \\ W(x-1) \times \alpha + (1-\alpha), & 0 < x < N \\ \frac{W(x-1) \times \alpha + (1-\alpha)}{1-\alpha}, & x = N \end{cases} \quad (4)$$

Where α is the percentage of weight to transfer to the next area towards the top. α ranges from 0, no weight transfer, to 1, a total weight transfer. When α is 0, the resulting weighted-AUC is equal to the conventional AUC; when α is 1, only the area at the top is considered. $W(0)$ is the weight for the bottom area. The new weight of an area is defined as a recursive formula using the weight of the previous area. This new weight vector is used to compute a new AUC value by adding up the areas times their corresponding new weight.

$$\text{weighted-AUC} = \sum_{i=0}^N \text{area}(i) \times W(i)$$

If the cost is known, we can use the cost ratio between the positive and negative classes to set the weight transfer rate, α .

$$\alpha = 1 - \text{cost ratio}$$

The maximum and minimum of weighted-AUC is the same as original AUC, 1 and 0. The advantage and disadvantages of weighted-AUC is similar to the

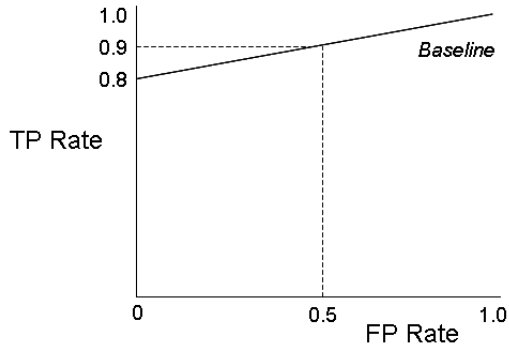


Figure 3: The ideal baseline performance

conventional AUC, except weighted-AUC is enhanced with the ability to adjust to different cost bias. This challenges the traditional view of ROC and AUC, where they were considered to be cost inconsiderate evaluation measures (Drummond and Holte (2004)).

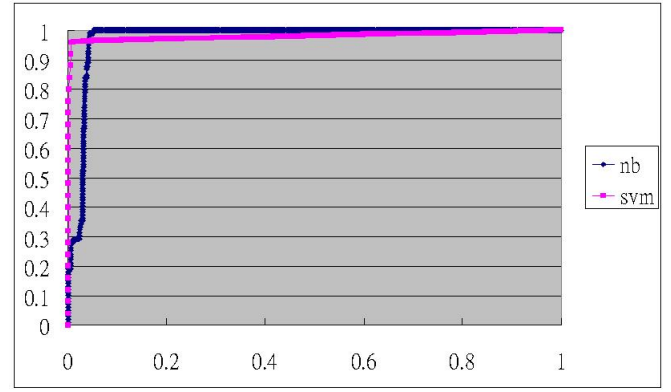
4 Ideal Baseline Performance

If the misclassification cost is known then one can define the baseline performance for the learners. Suppose the cost is same as in table 2(b), and if we assume the future class distribution is balanced, then the baseline performance is shown in figure.d. For example, if we need to classify 100 positive and 100 negative examples, with the cost of 1 to 5, when we classify everything as positive, the cost is 100. In order to match the same cost of 100, we can only afford to miss out 20 positive cases ($1/5 = 20\%$), while we correctly classify all negative examples, meaning zero FP rate. Following this reasoning, we can set the baseline performance by drawing a straight line connecting (80% TP rate, 0% FP rate) and (100% TP rate, 100% FP rate). A classifier is worth considering if it can achieve a performance above this line.

We have assumed equal class distribution in the above case for simplicity sake, but when we have an imbalanced class distribution, the baseline performance may need to be adjusted because the future distribution could also be skewed. If we use the training data class distribution as an estimate for the future class distribution, then the adjustment to the baseline performance is done by dividing the cost ratio over class distribution ratio. For example, if we have an imbalanced dataset with 100 positive and 400 negative examples, then with the cost of 1 to 5, the adjusted baseline should equal to $(1/5)/(100/400) = 0.8$. This means if we correctly classify every negative examples, then we can afford to miss 80 positive examples. The adjusted baseline performance will then go through (20%,0%) and (100%,100%).

It is important to know that when setting the misclassification cost, one should take the estimated class distribution into consideration, because they are closely related. If the class distribution ratio were to be less than the cost ratio, then the adjusted baseline performance will actually bias towards the negative class. Therefore, when one tries to set the cost ratio for an imbalanced dataset, where the rare class example is more important, one should generally consider have the smaller cost ratio than the estimated class distribution ratio.

It is interesting to note that the concept of cost curve is also about creating slopes, but instead of drawing slopes based on cost, the cost curve draws the slope lines tangent to a ROC curve to reflect the performance of a particular operating point.

Figure 4: ROC curve for *nb* and *svm* learning on 'anneal-2' dataset

We can see that *nb* touches the 100% TP rate much early than *svm* does, even though *svm* performs better at lower FP rate, and when working with imbalanced dataset, it is more desirable to have a learner that touches 100% TP rate at that lowest FP rate. So *nb* should be considered as having a better ROC curve for imbalanced datasets with cost bias towards the minority class.

5 Experiments

After introducing the background theoretical motivations for a weighted AUC, it would not be complete without looking at some real numbers from experiments. So, we have conducted an experiment to compare the normal AUC value and the weighted-AUC values. Four different learners were used in our experiment: Naïve Bayes(*nb*), Decision tree(*j48*), Support vector machines(*svm*), and k-nearest neighbour(*3nn*, we set the k to 3). The datasets are from UCI data repository (A. Asuncion, 2007) and we took multi-class problems and turned them into binary classification problems by treating one of the class as the positive class and use the rest of the classes as negative class. The process created 98 datasets.

When we compare the AUC value along side with weighted-AUC, it is clear that the more superior learner under AUC evaluation is not necessary better under weighted-AUC assessment. Table.3 shows both AUC and weighted-AUC values for a sample of 20 datasets out of 98 datasets across 4 different learners. We only shown 20 because it is sufficient to point out the difference between normal and weighted AUC. We have used 0.1 for the α , meaning we will transfer 10% of the weights.

The datasets in bold are where learners' performance ranking differs when evaluated under different metrics. To demonstrate, we look at the first occurrence of conflicting performance ranking, which happened with 'anneal-2' dataset. For this dataset, when ranking the learning performance by the original AUC values, the learners' rank were *j48*, *nb* and *svm*, and follow by *3nn*. However, when we look at weighted-AUC for the same dataset and learners, the ranking changed to *nb*, *j48*, *svm*, and *3nn*. So *nb* becomes the best learner out of the four. If we take a closer look at the ROC curves, as shown in figure.4, where we try to compare *nb* and *svm*, since they has the same normal AUC value, but different weighted-AUC value. We can see that *nb* does have a much better ROC curve for imbalanced dataset than *svm*, because it touches 100% TP rate much earlier than *svm*, which is not shown by the normal AUC.

Dataset	normal AUC				weighted-AUC			
	nb	j48	smo	3nn	nb	j48	smo	3nn
anneal-1	0.99	0.64	0.94	0.82	0.9	0.61	0.85	0.75
anneal-2	0.97	1	0.97	0.89	0.97	0.9	0.88	0.86
anneal-5	1	0.99	1	1	0.98	0.89	0.9	0.9
audiology-age	0.98	0.98	0.95	0.92	0.96	0.89	0.86	0.89
audiology-age_and_noise	0.99	0.96	0.92	0.63	0.94	0.87	0.84	0.6
audiology-cochlear_unknown	0.91	0.92	0.89	0.81	0.89	0.88	0.82	0.78
audiology-poss-noise	0.98	0.78	0.9	0.87	0.94	0.73	0.83	0.81
autos-2	0.65	0.81	0.63	0.81	0.64	0.77	0.63	0.78
balance-scale-B	0.33	0.5	0.5	0.35	0.33	0.5	0.5	0.35
balance-scale-L	0.99	0.84	0.92	0.98	0.99	0.81	0.83	0.97
balance-scale-R	0.99	0.84	0.92	0.98	0.99	0.81	0.83	0.97
breast-cancer	0.7	0.61	0.58	0.64	0.69	0.6	0.58	0.63
cleveland-heart-50_1	0.91	0.77	0.83	0.85	0.9	0.73	0.77	0.81
credit-rating	0.9	0.89	0.86	0.75	0.89	0.88	0.78	0.72
german_credit	0.79	0.65	0.67	0.61	0.78	0.63	0.65	0.6
Glass-buildwindfloat	0.76	0.81	0.57	0.87	0.75	0.78	0.58	0.82
Glass-buildwindnonfloat	0.7	0.76	0.5	0.84	0.69	0.73	0.5	0.8
heart-statlog	0.9	0.79	0.83	0.84	0.89	0.76	0.77	0.79
hepatitis	0.86	0.67	0.77	0.76	0.83	0.64	0.73	0.74
horse-colic.ORIG	0.79	0.5	0.71	0.61	0.78	0.5	0.68	0.59

Table 3: Compare normal AUC with weighted-AUC for UCI datasets.

We show a sample of the 98 datasets we have, and we highlight the datasets in bold to show where learners' performance ranking differs under different evaluation methods, i.e. normal and weighted AUCs.

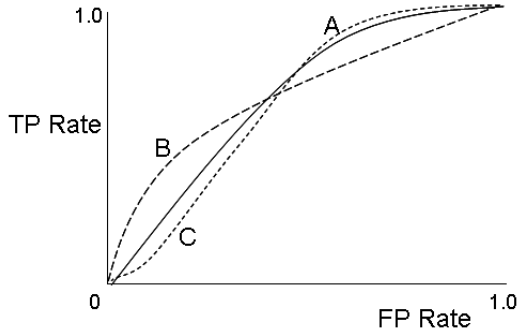


Figure 5: Same weighted-AUC example

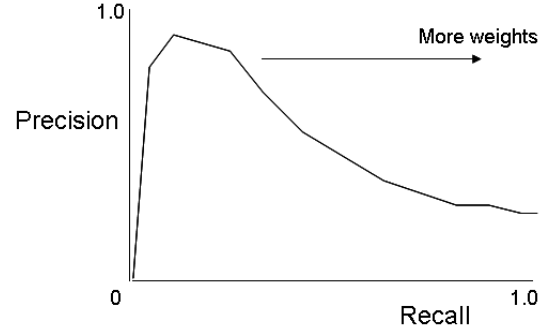


Figure 7: Precision-Recall curve

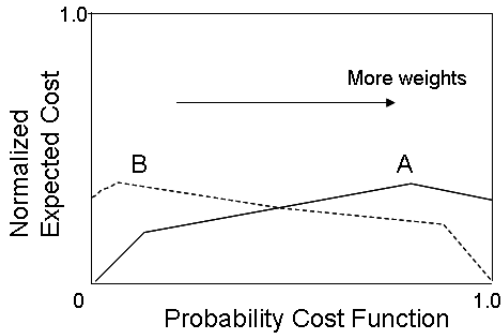


Figure 6: Cost curve

6 Discussion

There is an interesting similarity between cost curve and weighted-AUC, in which they are both designed to work with cost. However, cost curve is for visualizing performance over a range of costs, whereas the purpose of weighted-AUC is to use a biased weight vector to give an appropriate summarized quantity without sacrifice the advantage of graphical evaluation methods. Weighted-AUC is a compromise between a single value and a graphical view. The same weight vector idea can be applied on different graphical evaluation methods, such as cost curve and

precision-recall curve. For a cost curve, as shown in figure 6, the weight shifts towards the right side of the graph if the cost bias is towards the positive class. For the same cost preference, the weight also shifts towards the right side for a precision-recall curve, as shown in figure 7.

Some may argue, weighted-AUC will still have the same problem as the conventional AUC when the learners have equal weighted-AUC value, but have different ROC curves. This phenomena is illustrated in figure 5, where classifier A and classifier B have the same weighted-AUC. However, the purpose of weighted-AUC is to achieve cost bias evaluation, which means we can separate classifier A and B. It is alright to have two different learners both perform equally well under the same cost constraint.

When using weighted-AUC, one needs to set the α for the percentage of weight to transfer. This adjustable parameter allows the flexibility of adapting to different cost bias when the cost is known. In the case of unknown cost, one can always use the class distribution ratio as an estimate for the cost ratio. Weighted-AUC can be considered as a generalized form of AUC in terms of different cost biases.

Based on the concept of weighted-AUC, it can provide new insights for previous researches. For example, in a study by Weiss and Provost (2001), where they experimentally show that when conventional AUC is used as performance measure, then a balanced

class distribution should be used for training. This finding conforms with the concept of weighted-AUC, which equates that the conventional AUC as having zero weight shift; this translate to assuming zero cost bias or a balanced class distribution. Therefore, because the conventional AUC was used in the evaluation for their experiment, so only balanced class distribution will give the best results under the conventional AUC. It is like having a normally distributed graph and you want to find another bell-shaped graph that will yield the maximum overlap between the two, which will inevitably land you on another normally distributed graph with the same

7 Conclusion

We have introduced a new evaluation method for imbalanced datasets, called weighted-AUC. One can think of it as an enhanced version of AUC, a measure that is already gaining popularity in the imbalanced dataset community. We have shown why weighted-AUC is a better alternative under cost biased situations. A discussion for setting misclassification cost, baseline performance for imbalanced datasets is provided as a general guideline when dealing with imbalanced datasets. In the end of discussion section, we have also presented an example of how weighted-AUC can give new sights for researches in the data mining area. In the future, when one is dealing with imbalanced datasets, we recommend the use of weighted-AUC in place of conventional AUC in order to give a better cost-biased comparison.

References

- A. Asuncion, D. N. (2007), ‘UCI machine learning repository’.
URL: <http://www.ics.uci.edu/~mllearn/MLRepository.html>
- Davis, J. and Goadrich, M. (2006), The relationship between precision-recall and roc curves, *in* ‘In ICML06: Proceedings of the 23rd international conference on Machine learning’, pp. 233–240.
- Drummond, C. and Holte, R. C. (2000), Exploiting the Cost (In)sensitivity of Decision Tree Splitting Criteria, *in* ‘Proceedings of the Seventeenth International Conference on Machine Learning’, pp. 239–249.
- Drummond, C. and Holte, R. C. (2004), What roc curves can’t do (and cost curves can), *in* ‘ROCAI’, pp. 19–26.
- Green, D. and Swets, J. (1966), *Signal detection theory and psychophysics*, John Wiley and Sons Inc.
- Provost, F. (2000), Machine Learning from Imbalanced Data Sets 101, *in* ‘AAAI Workshop on Learning from Imbalanced Data Sets’, AAAI Press.
- Provost, F., Fawcett, T. and Kohavi, R. (1998), The case against accuracy estimation for comparing induction algorithms, *in* ‘Proceedings of the Fifteenth International Conference on Machine Learning’, pp. 43–48.
- Provost, F. J. and Fawcett, T. (1997), Analysis and visualization of classifier performance: comparison under imprecise class and cost distributions, *in* ‘Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining’, pp. 43–48.
- Spackman, K. A. (1989), Signal detection theory: Valuable tools for evaluating inductive learning, *in* ‘Proceedings of the Sixth International Workshop on Machine Learning’, pp. 160–163.
- Weiss, G. M. and Provost, F. (2001), The Effect of Class Distribution on Classifier Learning: An Empirical Study, Technical report, Department of Computer Science, Rutgers University. Technical Report ML-TR-44.

LBR-Meta: An Efficient Algorithm for Lazy Bayesian Rules

Zhipeng Xie

School of Computer Science

Fudan University

220 Handan Road, Shanghai 200433, PR. China

xiezp@fudan.edu.cn

Abstract

LBR is a highly accurate classification algorithm, which lazily constructs a single Bayesian rule for each test instance at classification time. However, its computational complexity of attribute-value pair selection is quadratic to the number of attributes. This fact incurs high computational costs, especially for datasets of high dimensionality. To solve the problem, this paper proposes an efficient algorithm LBR-Meta to construct lazy Bayesian rules in a heuristic way. It starts with the global classifier trained on the whole instance space. At each step, the attribute-value pair that best differentiates the performance of the current local classifier is selected and used to reduce the current subspace to a further smaller subspace for the next step. The selection strategy used has a linear computational complexity with respect to the number of attributes, in contrast to the quadratic complexity in LBR. Experimental results manifest that LBR-Meta has achieved comparable accuracy with LBR, but at a much lower computational cost.

Keywords: naïve Bayesian classifiers, lazy Bayesian rules, classification, decision trees.

1 Introduction

Classification is a core problem in machine learning and data mining. A variety of approaches such as naïve Bayes, nearest neighbours, decision trees, and neural networks, have been proposed to deal with it. However, none of these approaches can beat the others on all domains. Each approach has its own strong and weak points. Recently, a hotspot is to combine different learning paradigms together for higher overall predictability.

Among these numerous existing classification methods, the naïve Bayesian classifier (NB) (Duda & Hart 1973) is simplest and computationally most efficient. It is also robust to noise and irrelevant attributes, and has outperformed many complicated methods in varied application domains. However, all these advantages are obtained through its strong (or sometimes impractical) independence assumption that the attributes are conditionally independent given the class label. As a result, its performance may degrade significantly in the domains where the assumption does not hold. To alleviate this

problem, Kohavi (1996) proposed a hybrid algorithm NBTree to integrate decision tree algorithm and naïve Bayesian classifier together. Although NBTree has achieved higher accuracies than either naïve Bayesian classification method or decision tree learning algorithm in most datasets, it suffers from the small disjunct problem due to the tree structure. To deal with it, Zheng and Webb (2000) proposed the lazy Bayesian rule learning algorithm (LBR) which lazily induces at classification time a single Bayesian rule for each instance to be classified. The antecedent of a Bayesian rule is a conjunction of conditions in the form of attribute-value pairs, while the consequent is a local naïve Bayesian classifier trained on an instance subspace which is used to make the decision for the test unlabelled example. The objective of LBR is to grow the antecedent of a Bayesian rule that ultimately decreases the errors of the local naïve Bayesian classifier in the consequent of the rule. The LBR algorithm adopts a hill-climbing strategy in determining which attribute-value pair should be added to the antecedent at each step. It always makes the choice such that the local Bayesian classifier trained on the reduced subspace can obtain the highest reduction in classification error. Thus, for each candidate (attribute-value) pair, LBR has to train a local Bayesian classifier on the corresponding reduced subspace, and then estimate its error rate. This process requires substantial computational cost whose complexity is quadratic to the number of attributes. It is not desirable, especially for the datasets of high dimensionalities.

This paper proposes a heuristic strategy for attribute-value pair selection, which is based on the meta-information about the performance of the current local naïve Bayesian classifier. The attribute-value pair selected partitions the current subspace into two reduced smaller subspaces such that the current local naïve Bayesian classifier has significantly different performances on these reduced subspaces. Based on this selection strategy, an algorithm LBR-Meta is designed and implemented. LBR-Meta selects an attribute-value pair and adds it to the antecedent of the current Bayesian rule at each step. This process terminates only when the size of the reduced training subset is smaller than a threshold parameter. This simple criterion guarantees that the final subspace generated by LBR-Meta is quite small, when compared with the one generated by LBR.

This paper is organized as follows: Section 2 introduces naïve Bayesian classifier and local accuracy estimation. Section 3 describes the LBR-Meta algorithm in details. The experimental results are shown in Section 4. Finally, section 5 concludes the whole paper and points out the future work.

2 Naïve Bayesian Classification and Local Accuracy Estimation

Consider a domain where instances are represented as instantiations of a vector $A=\{a_1, a_2, \dots, a_m\}$ of m nominal variables. Here, each instance x takes a value $a_i(x)$ from $domain(a_i)$ on each a_i . Further, an example (or instance) x is also described by a class label $c(x)$ from $domain(c)$. Let $D=\{(x_i, c(x_i)) \mid 1 \leq i \leq n\}$ denote the training dataset of size n . The task of classification is to construct a model (or classifier) from the training set D , which is a function that assigns a class label to a new unlabelled example.

The underlying assumption of naïve Bayesian classifiers (NB) is that attributes are conditionally mutually independent given the class label. According to the Bayes Theorem, the probability of a class label i for a given unlabelled instance $x=(v_1, \dots, v_m)$ consisting of m attribute values is given by

$$P(c=i \mid x) = \frac{P(c=i) \times P(x \mid c=i)}{P(x)}.$$

It follows from the independent assumption that $P(x \mid c=i) = \prod_{k=1}^m P(a_k = v_k \mid c=i)$ holds. Thus, the class label with the

highest probability given the instance x , is used as the predicted class. Note that we do not need to compute the value of $P(x)$. This is because $P(x)$ is a constant for a given x . To put it formally, the naïve Bayesian classifier trained on D can be expressed as

$$NB(x, D) = \arg \max_i \left(P(c=i) \times \prod_{k=1}^m P(a_k = a_k(x) \mid c=i) \right)$$

Hence, the construction of naïve Bayesian classifier on D is to estimate the probabilities $P(c=i)$ and conditional probabilities $P(a_k=v_k \mid c=i)$, which are done in the following way:

$$P(a_k = v_k \mid c=i) = \frac{|\{x \in D : a_k(x) = v_k \text{ and } c(x) = i\}| + 0.5}{|\{x \in D : c(x) = i\}| + 0.5 \times |domain(a_k)|}$$

and

$$P(c=i) = \frac{|\{x \in D : c(x) = i\}| + 0.5}{|D| + 0.5 \times |domain(c)|}.$$

Local Accuracy Estimation

To determine which classification method is more suitable for a given data set, we often need to estimate the prediction accuracy of a classifier. Popular techniques of accuracy estimation include hold-out, cross-validation, and leave-one-out. As naïve Bayesian classifiers have been adopted as base local classifiers in this paper, which is easy to be updated incrementally, leave-one-out can be implemented easily and efficiently for accuracy estimation.

For the naïve Bayesian classifier $NB(D_1)$ trained on a training set D_1 , leave-one-out method can be used to estimate its accuracy as follows:

$$ACC_G(NB(D_1)) = |\{x \in D_1 \mid NB(x, D_1 - \{x\}) = c(x)\}| / |D_1|.$$

However, the estimated accuracy above is actually a kind of global accuracy, it is averaged over the whole instance space (or the whole training set). A classifier usually

performs differently in different regions (or subspaces). For example,

For a subset D_2 of D_1 , the *local accuracy* of $NB(D_1)$ on D_2 is estimated by

$$ACC_L(NB(D_1), D_2) = |\{x \in D_2 \mid NB(x, D_1 - \{x\}) = c(x)\}| / |D_2|.$$

It is evident that $ACC_G(NB(D_1)) = ACC_L(NB(D_1), D_1)$. This local accuracy plays an important role in classifier selection, because what we are interested in is actually which classifier performs best in the local subspace surrounding the target test example.

3 LBR-Meta: A heuristic algorithm for lazy Bayesian rules based on meta-information

A Bayesian rule r takes the form of “*antecedent*(r) \rightarrow *consequent*(r)”, where the antecedent of r is a conjunction of attribute-value pairs, and the consequent of r is a local naïve Bayesian classifier. An instance x satisfies a attribute-value pair (a, v) if and only if the value of attribute a on x equals to v (that is, $a(x)=v$). The instance subspace defined by r consists of all the instances that satisfy all the attribute-value pairs in r ’s antecedent. The subset of all training examples that satisfy the antecedent of r is called the local training set of r . The original LBR algorithm requires that the local naïve Bayesian classifier as the consequent should be trained on the local training set of r . This requirement is relaxed by the LBR-Meta algorithm of this paper. It is only required that the training set of the local naïve Bayesian classifier should be no less than the local training set of the Bayesian rule.

The pseudo-code of LBR-Meta algorithm is listed in figure 1, which will be fully explained in this section. For any given unlabelled test example x_{test} , LBR-Meta starts from the global Bayesian rule where the antecedent is empty and the consequent is trained on the whole training set D . At each step, an attribute-value pair is selected and added into the antecedent to reduce the current instance subspace, and hence reduce the corresponding local training subset. There are two key problems to be solved in the LBR-Meta algorithm. The first key problem is how to select an attribute-value pair. After an attribute-value is added to the antecedent and the subspace is refined to a further smaller subspace, the second problem appears: how to determine which local classifier is best suited for this reduced subspace. The following two subsections are devoted to the detailed solutions to these two problems.

3.1 A Heuristic Criterion for Attribute-Value Pair Selection

The first problem to be dealt with is: which attribute-value pair should be selected to reduce the current instance subspace? The original LBR algorithm makes the decision so that the local classifier trained on the reduced subspace has the lowest estimated error rate. With this selection criterion, one classifier has to be induced for each attribute-value pair, which leads to high computational overhead. In order to improve computational efficiency, this paper takes a heuristic way: it selects the attribute-value pair that can differentiate the performance of the current local classifier on the current subspace. The details go as follows:

Let r be the current Bayesian rule, D_{local} be the current local training set, and $NB(D_{current})$ be the current local naïve Bayesian classifier associated with r . As stated above, this local naïve Bayesian classifier is trained on $D_{current}$ which does not necessarily equal to D_{local} . However, it is required that $D_{local} \subseteq D_{current}$. A boolean attribute c_{meta} is appended to each training example, whose value denotes whether or not the current local naïve Bayesian classifier $NB(D_{current})$ can classify the corresponding training example correctly with leave-one-out method. Put it formally, for each $x \in D_{local}$:

$$c_{meta}(x) = \begin{cases} true & \text{if } NB(x, D_{current} - \{x\}) = c(x); \\ false & \text{otherwise.} \end{cases}$$

According to the values of c_{meta} , the current local training set D_{local} is partitioned into two subsets:

$$(D_{local})_{true} = \{x \in D_{local} | c_{meta}(x) = true\},$$

and

$$(D_{local})_{false} = \{x \in D_{local} | c_{meta}(x) = false\}.$$

Furthermore, each attribute-value pair (a, v) can also partition the current local training set D_{local} into two subsets:

$$(D_{local})_0 = \{x \in train | a(x) = v\}$$

and

$$(D_{local})_1 = \{x \in train | a(x) \neq v\}.$$

The subset $(D_{local})_0$ consists of the training examples in D_{local} that take value v on attribute a ; while $(D_{local})_1$ consists of the training examples in D_{local} that do not take value v on a . The current local naïve Bayesian classifier may have different performance (or accuracies) on these two subsets. Its local accuracy on $(D_{local})_i$, $i \in \{0, 1\}$, is estimated by

$$ACC_L(NB(D_{current}), (D_{local})_i) = \frac{|(D_{local})_{i,true}|}{|(D_{local})_i|},$$

$$\text{where } (D_{local})_{i,true} = (D_{local})_i \cap (D_{local})_{true}.$$

The idea used for heuristic attribute-value pair selection is: the more the difference between the local accuracies of these two subsets is, the more likely can we expect to get good performance by zooming the local naïve Bayesian classifier into the corresponding subspace of $(D_{local})_0$. Using information gain, the goodness of an attribute pair (a, v) is measured by LBR-Meta as follows:

$$Goodness(a, v) = Info(|(D_{local})_{true}|, |(D_{local})_{false}|) + \sum_{i=0}^1 \frac{|(D_{local})_i|}{|D_{local}|} Info(|(D_{local})_{i,true}|, |(D_{local})_{i,false}|),$$

$$\text{where } (D_{local})_{i,false} = (D_{local})_i \cap (D_{local})_{false},$$

$$\text{and } Info(n_1, n_2) = -\log \frac{n_1}{n_1 + n_2} - \log \frac{n_2}{n_1 + n_2}.$$

The information gain has also been used for attribute selection in a famous decision tree algorithm ID3 (Quinlan 1986, Quinlan 1993). The difference is that it is concerned about the Boolean attribute a_{meta} , while it is concerned about the class attribute c in ID3.

Comparison with LBR algorithm: The LBR algorithm tries to add each candidate attribute-value pair to the antecedent of the current Bayesian rule, then reduces the

local training set, and then trains a local classifier accordingly. The total time spent is $O(|A| \times |A| \times |D_{local}|)$.

The LBR-Meta algorithm calculates the goodness of each candidate attribute-value pair. The time needed in total is $O(|A| \times |D_{local}|)$.

3.2 Local Naïve Bayesian Classifier Selection

After an attribute-value pair has been selected and added to the antecedent of the rule, the current local subspace is reduced to a further smaller subspace. The current local training set D_{local} is also reduced to a smaller local training subset D_{red} . The aim of local classifier selection is to select the local classifier that has the highest local accuracy on the reduced smaller subspace that is measured on the smaller local training subset D_{red} . LBR-Meta uses the variable *LocalClassifiers* to represent the set of all qualified local naïve Bayesian classifiers that have already been generated. The global naïve Bayesian classifier is assumed to be qualified and added into *LocalClassifiers* (line 2 in figure 1). Once a local naïve Bayesian classifier *SubLocalNB* is trained on the reduced training subset D_{red} at each step, it is compared with all the qualified local classifiers in *LocalClassifiers*. If *SubLocalNB* has the highest estimated local accuracy on D_{red} , it is qualified and added into *LocalClassifiers* (lines 17-18 in figure 1); otherwise, *SubLocalNB* will be discarded.

- If D_{red} contains too few training examples, that is, the size of D_{red} is less than a threshold *Thresh* (line 9 in figure 1), the local accuracy estimated on D_{red} can not provide reliable information about the accuracy on the corresponding subspace, and thus the current local naïve Bayesian classifier *CurrentLocalNB* is used directly to make the decision for x_{test} . Note: the default value of *Thresh* is set to 5.
- If the size of D_{red} is larger than or equal to the threshold *Thresh*, we first estimate the local accuracy on D_{red} for each local classifier in *LocalClassifiers* (lines 10-12 in figure 1). The one with the highest estimated local accuracy is selected and denoted by *BestSupNB* with the estimated local accuracy stored in the variable *BestSupAcc* (lines 13-14 in figure 1). Then we compare the size of D_{red} with another parameter *MinNumObj* which should be set greater than *Thresh* (Note that *MinNumObj* has default value 15 in this paper):

A. If the size of D_{red} is less than *MinNumObj*, the decision made by the local classifier *BestSupNB* is returned as the result (line 15 in figure 1).

B. If D_{red} contains enough training examples to train a local naïve Bayesian classifier *SubLocalNB* (line 16 in figure 1), or in another words, the size of D_{red} is no less than *MinNumObj* (line 15 in figure 1), the classifier *SubLocalNB* will compete with other local classifiers previously generated for the domination of the subspace corresponding to the training subset D_{red} . If *SubLocalNB* is more accurate on D_{red} than *BestSupNB*, the classifier *SubLocalNB* is added into *LocalClassifiers*, and the *CurrentLocalNB* is set to be *SubLocalNB* (lines 17-19 in figure 1); otherwise, the *CurrentLocalNB* is set to be *BestSupAcc* (lines 20-21 in figure 1).

LBR-Meta**Input:** A : a set of attributes D : a set of training examples described using A and class attribute c x_{test} : a test example described using A **Output:** a predicted class for x

```

1   $CurrentLocalNB := NB(D)$ ;
2  Add  $NB(D)$  into the set of local classifiers  $LocalClassifiers$ ;
3   $D_{local} := D$ ;  $A_{local} := \{a \in A \mid a(x_{test}) \text{ is not missing}\}$ ;
4  FOR each example  $x$  in  $D$  DO  $c_{meta}(x) := \begin{cases} true, & \text{if } NB(x, D - \{x\}) = c(x) \\ false, & \text{if } NB(x, D - \{x\}) \neq c(x) \end{cases}$ ; ENDFOR
5  WHILE ( $A_{local}$  is not empty) DO
6    Calculate  $Goodness(a, a(x_{test}))$  for each attribute  $a \in A_{local}$ , according to the equation ();
7     $att := \arg \max_{a \in A_{local}} Goodness(a, a(x_{test}))$ ;
8     $D_{red} := \{x \in D_{local} \mid att(x) = att(x_{test})\}$ ;  $A_{local} := A_{local} - \{att\}$ ;
9    IF  $|D_{red}| < Thresh$  THEN return  $CurrentLocalNB(x_{test})$ ; ENDIF
10   FOR each local classifier  $NB(D_i)$  in  $LocalClassifiers$  DO
11     estimate its local accuracy on  $D_{red}$ :  $ACC_L(NB(D_i), D_{red})$ ;
12   ENDFOR
13    $BestSupNB := \arg \max_{NB(D_i) \in LocalClassifiers} ACC_L(NB(D_i), D_{red})$ ;
14    $BestSupAcc :=$  the estimated local accuracy of  $BestSupNB$  on  $D_{red}$ ;
15   IF  $|D_{red}| < MinObjNum$  THEN return  $BestSupNB(x_{test})$ ; ENDIF
16    $SubLocalNB := NB(D_{sub})$ ;
17   IF  $ACC_G(SubLocalNB) \geq BestSupAcc$  THEN
18     add  $SubLocalNB$  into  $LocalClassifiers$ ;
19      $CurrentLocalNB := SubLocalNB$ ;
20   ELSE
21      $CurrentLocalNB := BestSupNB$ ;
22   ENDIF
23    $c_{meta}(x) := \begin{cases} true, & \text{if } NB(x, Tr - \{x\}) = c(x) \\ false, & \text{if } NB(x, Tr - \{x\}) \neq c(x) \end{cases}$  for each  $x$  in  $D_{red}$  where  $NB(Tr) = CurrentLocalNB$ ;
24    $D_{local} := D_{red}$ ;
25 ENDWHILE

```

Figure 1. The LBR-Meta Algorithm

Finally, due to the fact that the current local naïve Bayesian classifier has possibly been changed, we need to update the attribute values of c_{meta} for all training examples in the reduced subspace (line 23 in figure 1).

3.3 Analysis of LBR-Meta

LBR uses a statistical sign-test to control tradeoff between the decreasing error by removing harmful attribute-value pair and increasing error as a result of reducing the accuracy of the probability estimations of the local naïve Bayesian classifier due to decreases in the size of the available training set. In addition, this statistical sign-test is also used as the termination condition for the repetitive process. However, the strategies adopted by LBR-Meta are different from LBR, which are described as follows.

Firstly, even if the accuracy of the probability estimations of the local naïve Bayesian classifier is decreasing with a smaller training set, it has already been reflected in its estimated local accuracy. That is to say, it is expected that the classifier should get low estimated accuracy if the probability estimations inside it are not reliable. Or if the estimated accuracy of the classifier is high, it means that the accuracy of the inside probability estimations is acceptable.

Secondly, LBR algorithm tries to add one attribute-value pair to the antecedent at each step. This process is repeated until no attribute-value pair could lead to a local naïve Bayesian classifier with statistically higher accuracy on the reduced subspace. However, in LBR-Meta, even if the attribute-value pair that is selected cannot lead to a qualified local naïve Bayesian classifier (that is, the classifier trained on the reduced training subset has lower accuracy), the process is not terminated. This strategy does not suffer from local maxima. The final subspace produced by LBR-Meta for a given test example is much smaller than that produced by LBR.

In spite of the fact that the individual attribute-value pair selected by LBR algorithm may be better than that selected by LBR-Meta, the above two strategies taken by LBR-Meta have compensated for this shortcoming in some degree. It will be shown in the experimental part that LBR-Meta has also yielded high accuracies comparable with LBR.

Furthermore, our actual objective is to maximize the local accuracy at the point of the test example, which is usually approximately by the accuracy on a subspace around the test example. There is also a tradeoff between variance and bias. If the subspace is too large, the accuracy estimated may differ significantly from the accuracy at the point of the test example. On the contrary, if the subspace is too small, the accuracy estimated is not reliable (with large variance).

4 Experimental Results

To evaluate the performance of the proposed LBR-Meta algorithm, we compare it with two other closely related algorithms: the naïve Bayesian classifier (NB), and the lazy Bayesian rule algorithm (LBR). Twenty-three data sets from UCI machine learning repository are used for the comparison, with detailed information listed in table 1. The datasets are drawn randomly, with the requirement

that each dataset should contain at least 300 examples. The numbers of attributes vary from 8 to 36, and the numbers of examples from 303 to 12960. Ten-fold cross validation is conducted on each data set, such that each fold has at least 30 examples. For LBR can only deal with discrete attributes, the continuous attributes are discretized by an entropy-based discretization algorithm (Fayyad & Irani 1993) as a preprocess.

	# examples	# attributes	# classes
Australian	690	14	2
Breast	699	10	2
Chess	3196	36	2
Cleve	303	13	2
Crx	690	15	2
Diabetes	768	8	2
German	1000	20	2
Horse-Colic	368	22	2
Hypothyroid	3163	25	2
Ionosphere	351	34	2
Mushroom	8124	22	2
Nursery	12960	8	5
Pendigits	10992	16	10
Pima	768	8	2
Satimage	6435	36	6
Segment	2310	19	7
Shuttle-Small	5800	9	7
Sick	2800	29	2
Solar	323	12	6
Soybean-Large	683	35	19
Tic-Tac-Toe	958	9	2
Vote	435	16	2
Waveform-21	5000	21	3

Table 1: Datasets used for comparison

The error rates of these algorithms over all datasets are listed in table 2. The final row shows the mean error rates across all the datasets. Among the three algorithms, LBR-Meta gets the best result, which is slightly better than LBR, and greatly better than NB. When we look at individual datasets, it is found that LBR-Meta has lower error rates than LBR on 10 datasets and higher than LBR on 11 datasets.

The mean error rate is only a naive measurement of a classification method over these datasets. To evaluate the relative error rate reduction, we present a new measurement called the *relative difference* (*rdiff*). For a given a dataset, assume e_1 be the error rate of the first method, and e_2 be the error rate of the second method. The relative difference from e_2 to e_1 is defined as:

$$rdiff(e_2, e_1) = \frac{e_2 - e_1}{\max\{e_1, e_2\}}$$

When both e_1 and e_2 are zero, the value is defined to be 0. It can be seen that if e_2 equals to e_1 , the relative difference is zero; if e_2 is larger than e_1 , the relative difference is positive; and if e_2 is less than e_1 , the relative difference is negative. Therefore, the smaller the relative difference

from e_2 to e_1 is, the better is the second method relatively than the first method on the dataset. Furthermore, it is evident from the definition that $rdiff(e_2, e_1) = -rdiff(e_1, e_2)$. The relative difference may be better than the error rate ratio used by Zheng & Webb (2000), in that the error rate ratio will lead to an infinite value when some error rate appears (or approaches) zero, and in that the $rdiff$ value lies in the interval $[-1, 1]$, while the error rate ratio lies in the interval $(0, \infty]$. The last row in Table 2 gives out the mean relative difference of each algorithm to LBR-Meta over all datasets. From the fact that the mean relative difference from NB to LBR-Meta is 25.5%, conclusion can be drawn that LBR-Meta has substantially reduced the error rates of NB.

In addition, the one-tailed pairwise t -test (with significance level set at 5%) shows that LBR-Meta wins on 3 datasets (Chess, Nursery, and Tic-Tac-Toe), and loses on 1 dataset (Waveform-21) when compared with LBR. LBR-Meta also wins significantly on 11 datasets and loses on 0 when compared with NB. When comparing LBR with NB, we find that LBR significantly wins on 13 datasets and loses on 0 datasets.

These results have shown that the algorithm LBR-Meta is comparable to LBR on the experimental datasets, but the computational cost of LBR-Meta is far less than that of LBR, which is shown next.

	LBRMeta	LBR	NB
Australian	14.3%	14.2%	14.3%
Breast	3.4%	3.0%	3.0%
Chess	1.3%	2.6%	12.0%
Cleve	17.5%	16.5%	16.5%
Crx	14.3%	14.8%	14.2%
Diabetes	25.7%	25.4%	25.4%
German	26.5%	25.1%	25.2%
Horse-Colic	20.4%	17.4%	21.2%
Hypothyroid	1.1%	1.2%	1.5%
Ionosphere	10.2%	10.5%	10.0%
Mushroom	0%	0%	3.5%
Nursery	1.7%	2.3%	9.7%
Pendigits	4.3%	3.9%	12.3%
Pima	24.0%	25.5%	25.3%
Satimage	13.8%	13.6%	17.9%
Segment	6.1%	5.8%	8.8%
Shuttle-Small	0.3%	0.3%	0.7%
Sick	2.1%	2.6%	2.9%
Solar	29.5%	30.7%	30.7%
Soybean-Large	7.5%	6.7%	7.0%
Tic-Tac-Toe	11.2%	14.4%	29.8%
Vote	5.5%	6.4%	9.7%
Waveform-21	19.0%	16.2%	18.9%
Mean	11.27%	11.28%	13.93%
Mean $rdiff$ to LBR-Meta	0	0.022	0.255

Table 2: Error rate comparison

The information about the runtimes is shown in table 3. For each dataset (in the first column), the second column records the running time (in seconds) of LBRMeta, the third column records the running time of LBR, and the fourth column is the ratio of LBR's runtime to LBRMeta's runtime. The runtime values are averaged over the ten folds. If the value in the fourth column is large than 1, it means that LBR-Meta runs faster on the corresponding dataset. From the table, LBR-Meta runs about 10.4 times faster than LBR on Soybean-Large (with 35 attributes), 7.45 times faster on Chess (with 36 attributes).

	LBR-Meta	LBR	Runtime Ratio of LBR to LBR-Meta
Australian	0.5922	0.4798	0.81
Breast	0.2986	0.1955	0.65
Chess	28.9234	215.403	7.45
Cleve	0.0764	0.0627	0.82
Crx	0.6391	0.5564	0.87
Diabetes	0.2422	0.1235	0.51
German	1.0595	1.1486	1.08
Horse-Colic	0.1124	0.2388	2.12
Hypothyroid	27.9467	36.6264	1.31
Ionosphere	0.250	0.9625	3.85
Mushroom	86.9904	188.848	2.17
Nursery	69.501	71.2436	1.03
Pendigits	62.7794	183.488	2.92
Pima	0.2499	0.1173	0.47
Satimage	46.1528	193.495	4.19
Segment	5.8579	10.8002	1.84
Shuttle-Small	16.386	18.2967	1.11
Sick	24.9702	49.5563	1.98
Solar	0.1062	0.1345	1.27
Soybean-Large	1.4609	15.2015	10.41
Tic-Tac-Toe	0.4281	0.4157	0.97
Vote	0.2048	0.3374	1.65
Waveform-21	17.5639	32.7342	1.86

Table 3: Runtime Comparison (in seconds)

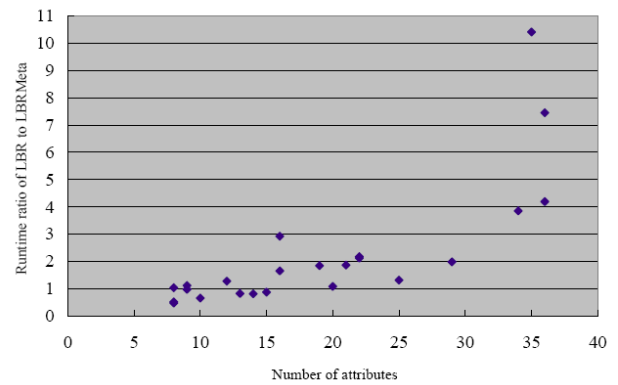


Figure 2: Scatter Plot

Furthermore, we use a scatter plot in Figure 2 to check the relationship between the runtime ratio of LBR to LBR-meta and the number of attributes. In Figure 2, each dot represents a dataset whose x coordination is the

number of attributes, and whose y coordination is the runtime ratio of LBR to LBR-Meta. A dot above the line $y=1$ means that the algorithm LBR-Meta is faster than LBR on the corresponding dataset. The higher is the y coordination of a dot, the faster and the better LBR-Meta is. Clearly, it can be observed from the figure that LBR-Meta runs faster than LBR for all the datasets with more than 15 attributes, and that LBR-Meta is much faster than LBR with increasing number of attributes in the datasets.

Finally, an ensemble technique, Bagging (Breiman 1996), is applied to LBR-Meta, LBR-Bag, and NB to check its effect in reducing error rate. As has been pointed out by Bauer & Kohavi (1999), naive Bayesian classifiers are not sensitive to the small change caused by resampling or replication of training examples. However, due to the facts that the final naive Bayesian classifiers generated by LBR-Meta and LBR are trained on a local training subset, and that the small change may change the attribute-value selected in the repetitive process, it is conjectured that LBR-Meta and LBR be more sensitive to Bagging. The experimental results are shown in Table 4, where the ensemble size is set at 10.

	LBRMeta-Bag	LBR-Bag	NB-Bag
Australian	14.1%	14.5%	14.3%
Breast	3.6%	2.9%	2.9%
Chess	0.8%	1.7%	13.2%
Cleve	16.5%	16.5%	16.5%
Crx	13.8%	13.9%	14.5%
Diabetes	25.1%	24.6%	25.0%
German	26.2%	24.8%	25.0%
Horse-Colic	17.1%	17.1%	20.9%
Hypothyroid	1.0%	1.1%	1.5%
Ionosphere	10.0%	9.4%	10.2%
Mushroom	0%	0%	3.6%
Nursery	1.2%	2.1%	9.8%
Pendigits	3.2%	2.9%	12.1%
Pima	23.3%	24.2%	24.6%
Satimage	12.0%	12.5%	17.8%
Segment	5.5%	4.9%	8.6%
Shuttle-Small	0.2%	0.3%	0.7%
Sick	2.2%	2.6%	3.1%
Solar	28.8%	28.2%	30.1%
Soybean-Large	6.7%	6.7%	7.6%
Tic-Tac-Toe	6.6%	10.1%	29.8%
Vote	5.3%	4.4%	9.9%
Waveform-21	16.8%	16.1%	18.9%
Mean	10.43%	10.50%	13.94%

Table 4: Bagging on LBR and LBR-Meta

From the results listed in Table 4, it can be seen that Bagging technique totally has no effect on naive Bayesian (NB) method, as NB-Bag has almost the same mean error rate as NB. By applying Bagging to LBR-Meta, the resulting LBRMeta-Bag gets 20 wins, 1 draws and only 2 loses when compared with the base LBR-Meta algorithm, while the mean error rate decreases from 11.27% to 10.43%. Applying Bagging to LBR has similar effects too.

5 Conclusion and Future Work

This paper proposes an algorithm LBR-Meta which makes use of a heuristic criterion for attribute-value pair selection in the lazy construction of a Bayesian rule. This criterion can be calculated in linear time with respect to the number of attributes, which has greatly improved the efficiency of the resulting algorithm. Experimental results have also shown that this algorithm also achieves high accuracy comparable to the classical LBR algorithm.

The future work about LBR-Meta is to enhance it with the ability to handle continuous attributes directly. This may be done as follows: Given a continuous attribute, the information gain with respect to the c_{meta} attribute is calculated for each possible split point. The best split point with the highest information gain value is selected, which can partition the current subspace into two reduced smaller subspaces. This continuous attribute together with the best split point competes with all the other attributes in the attribute-value pair selection. This straightforward process can endows the LBR-Meta with continuous-attribute handling ability.

Acknowledgements: This work was funded in part by National Natural Science Foundation of China under grant number 60503025

6 References

- Bauer, E., & Kohavi, R. (1999): An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36: 105-142
- Blake, C., Keogh, E., & Merz, C.J. (1998): UCI Repository of Machine Learning Databases. <http://www.ics.uci.edu/~mllearn/MLRepository.html>. Irvine, CA: University of California, Department of Information and Computer Science.
- Breiman, L. (1996): Bagging predictors. *Machine Learning* 24: 123-140
- Duda, R. O., & Hart, P. E. (1973): Pattern classification and scene analysis. New York: John Wiley
- Fayyad, U.M., & Irani, K.B. (1993): Multi-interval discretization of continuous-valued attributes for classification learning. *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence* (pp. 1022-1027), San Mateo, CA: Morgan Kaufmann.
- Kohavi, R. (1996): Scaling up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (pp. 202-207), Menlo Park, CA: AAAI Press.
- Quinlan, J. R. (1986): Induction of decision trees. *Machine Learning* 1: 81-106
- Quinlan, J. R. (1993): C4.5: Programs for machine learning. Morgan Kaufmann
- Zheng, Z., Webb, G. I. (2000): Lazy learning of Bayesian Rules. *Machine Learning* 41: 53-87

Exploratory Mining over Organisational Communications Data

Alan Allwright¹

John F. Roddick²

¹ Defence Science and Technology Organisation
PO Box 1500, Edinburgh, South Australia 5111,
Email: alan.allwright@dsto.defence.gov.au

² School of Computer Science, Engineering and Mathematics
Flinders University,
PO Box 2100, Adelaide, South Australia 5001,
Email: roddick@csem.flinders.edu.au

Abstract

Exploratory data mining is fundamental to fostering an appreciation of complex datasets. For large and continuously growing datasets, such as obtained by regular sampling of an organisation's communications, the exploratory phase may never finish. This paper describes a methodology for exploratory data mining within an organisational communications dataset. A model of support for knowledge discovery is described in conjunction with a communications based concept hierarchy. This is then used as the basis for a set of visualisations. The intention of supporting visualisations in this way is to establish a sound set of requirements for the representation of communications data. The visualisations provide several interconnected representations of the data, as well as support query and drill-down into a dataset. It is suggested that this interaction with the dataset facilitates an appreciation of the data which precedes and shapes knowledge discovery. A communications analysis example is developed using the visualisations within the context of exploratory data mining.

Keywords: Exploratory Data Mining, Visualisation, Communications Analysis.

1 Introduction

Organisational communications studies¹ are often conducted in order to improve existing, or introduce new, services to users. In many organisations such services include email, WWW access, and access to distributed repositories, in addition to telephone and fax.

In order to provision the facilities for services an assessment of demand is necessary. In communications analysis demand is typically time and/or topology based. In time based studies the information traffic flow characteristics such as arrival rate, message length and service time are assessed (q.v. Jain 1991), while topology based studies tend to investigate characteristics of network connectivity such as

link density and distribution (q.v. Albert & Barabási 2002).

An analysis of demand does not necessarily provide an adequate appreciation of where, in an organisational sense, people (or other business components) are creating and using applications. An assessment for the provision of services must look beyond technical measures. Neither time nor topology based analysis necessarily provide insight into the relationships between the various data dimensions. The multidimensional nature of communications data, the interconnectedness of communications components, and the sheer volume of data collected can complicate the analysis of communications systems².

Data mining supports communications analysis by providing insight gleaned from the total communications dataset and the provision of techniques to detect patterns and isolate data of interest. Overviews of data mining studies in the area of communications networks (Garofalakis & Rastogi 2001, Hulst et al. 2001, Julisch & Dacier 2002) offer some insight into current efforts to address these problems³.

The approach discussed in this paper is to support an investigation focussed on a large and diverse collection of communications data. Our objective is to provide analysis in a general sense, with the proposed data mining capability placed between low level datasets and more specialised higher level mining and analysis tools. A set of interrelated visualisations are developed for this approach. The visualisations are described in the context of communications analysis, and the potential application of data mining and knowledge discovery (DMKD) algorithms. An important aspect of the visualisations is to blend communications analysis with data mining to offer new perspectives through explorations. We also describe a graphical user interface (GUI) designed to assist in exploratory data mining within a communications dataset. The objective of the GUI is to assist in representing the multidimensional aspects of communications data, and to ultimately support knowledge discovery.

The structure of this paper is as follows. A discussion on exploratory data mining is provided first in

¹Organisational communications studies may cover the full spectrum of human communications, however, in this paper the term communications is used to represent electronic communications between organisational components. Components are used generically and include business units, people, or other uniquely identifiable parts of an organisation.

Copyright ©2008, Commonwealth of Australia. This paper appeared at the Australasian Data Mining Conference (AusDM 2008), Glenelg, South Australia, Australia. November 2008. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 87, John F. Roddick, Jiuyong Li, Peter Christen and Paul Kennedy, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

²Overviews of the types of networks, the range of problems, a variety of visualisations, and the general complexity of current communications networks research and analysis are provided by Keller & Keller (1993) and Dodge & Kitchin (2001a,b).

³The Association for Computing Machinery (ACM) has held workshops in mining network (MineNet) data in 2005 and 2006. Prior to this the related ACM workshops were in the more general area of data mining and knowledge discovery (DMKD). The principal topics of interest in the MineNet workshops are Collection, storage and access infrastructure, Network data analytics techniques and tools, and Applications to network operations and management. Within the DMKD community these relate closely with the data understanding and data preparation, modelling, and evaluation and deployment classifications of the CRISP-DM knowledge discovery process (Chapman et al. 1999).

Section 2. In order to establish requirements for the proposed GUI, we establish, in Section 3, what we mean by communications in the context of an organisation. In Section 4, a set of visualisations and their usage is described. Conclusions and further work are discussed in Section 5.

2 Exploratory Data Mining

Similar to mining in the general sense, the objective of data mining is to extract valuable or interesting information from a data resource. A discussion of interestingness⁴ is outside the scope of this paper. However, the DMKD supporting processes of directing, filtering and isolating data are, as discussed in this paper, necessary to provision of contemporary communications analysis.

Data mining has directed and undirected (or exploratory) aspects. Directed data mining has specific goals to efficiently and deliberately extract information from the dataset that has a greater than average prospect of being of interest to an analyst. An exploratory perspective implies a more loosely framed goal. Consequently, there is greater scope for ad-hoc interaction with the dataset. What is learned through the exploratory activities can be used to inform and guide future mining activities. In either case, directed or exploratory, the resource (i.e. the dataset), and the tools (e.g. *association mining*, *clustering* and *classification*) can be the same.

Ceglar, Roddick, Mooney & Calder (2003) propose a human-centric, tightly-coupled knowledge discovery process. The process is proposed in support of the assertion that only a human can direct their enquiries to derive meaning from their interaction with the dataset. Such a directed process could only occur if conducted within a sound contextual appreciation of the dataset. This appreciation would necessarily include an understanding of the discipline and methods through which the data was obtained, knowledge of the design and organisation of the data repository, and an understanding of the range and patterns within the data.

This paper suggests a bootstrap approach where familiarity with the dataset, facilitated through human-centric exploratory data mining, fosters the development of a contextual appreciation that in-turn supports a directed knowledge discovery process. The approach attempts to presume a minimum *a priori* knowledge of the dataset. Consequently, it is suggested that the appreciation precedes, facilitates, and shapes the knowledge discovery process. However, the presumption of minimal prior knowledge of the dataset is leveraged against an expectation of a good understanding of the discipline (in this case organisational communications). A good knowledge of the discipline, in-turn, assists the analyst in framing appropriate queries within the context of the dataset. This duality forms the basis for differentiating implicit from explicit information, as discussed later in this section.

Importantly, for human-centric explorations, and in particular during their early stages, the ability to form ad-hoc queries, to filter and isolate data are critical. The issue of flexibility in the formation of queries is recognised in general in DMKD. To quote Han (Ankerst 2002):

Data selection and viewing of mining results should be fully interactive, the mining process should be more interactive than the cur-

rent state of the art and embedded applications should be fairly automated.

This quote also highlights a combination of capabilities very common in DMKD, namely data selection, visualisation, and interaction. These capabilities, in particular visualisation, are central to the approach proposed in this paper.

Many visualisations in the area of DMKD, and indeed within the general area of visualisation, are based upon predominantly intrinsic information. This information is essentially self evident within the context of a dataset. However, extrinsic (that is, outside of the dataset) information can be important. As an example, in association mining often visualisations are constructed to represent rules and associated quality metrics such as confidence and support (qv. Hao et al. 2001, Hofman et al. 2000, Ong et al. 2002, Rainsford & Roddick 2000). The rules and associated quality information can be derived entirely from the dataset. This tendency to consider only intrinsic information potentially overlooks valuable support from the discipline to which the rules apply. With regard to the typical market basket example, such intrinsic information includes the individual shopping items, quantity and cost. Sources of extrinsic information include taxonomies/ontologies and process models. With regard to the market basket example, extrinsic information may be derived from classes of items (e.g. cleaning products, cereals, etc.), rules for how the items satisfy the user requirements (e.g. for cleaning, for eating, etc.) and models of how the shopping is conducted (e.g. through mail-order, Internet, or visited physically). Extrinsic information considered in this paper is developed on the basis of technical characteristics of communications. Additionally, the mining process may be informed by the functions of the organisation and its communications systems.

3 Organisational Communications

All organisations are dependant upon communications for their day-to-day operation. People are increasingly becoming skilled in an ever growing range of communications technology. The ways in which people and collectively their organisations communicate are as individual as the people and the roles they conduct (El-Shinnawy & Markus 1997, Michailidis & Rada 1997). In this section a model for organisational communications data mining is developed. The model is based upon extrinsic communications factors in order to overcome limitations noted in the previous section. The model is independent of any particular dataset. Specific functions of the model are to clarify the data types and support the design of the visualisations. The model helps to scope explorations and to define what is available in the dataset. In addition, the model provides assistance in the interpretation of exploratory outcomes.

The analysis of communications within and between organisations must take into account many modes of communication. The ability to analyse predominant, or conversely unusual, modes of communication requires insight into the total range of communications for a given situation. Additionally, insight into a range of communications supports a balanced investigation across all represented communication and avoids being prematurely drawn to a conclusion. There are many techniques supporting the analysis of specific communication domains (Card et al. 1999, Keller & Keller 1993), including computer, radio or social communications. Our objective is to provide a consistent framework to collect, associate and interact with communications data from a broad

⁴A valuable survey of this topic is provided by Geng & Hamilton (2006).

range of domains. We are unaware of similar systematic approaches to support organisational communications combining the individual domain analyses. In order to achieve this, we first establish what we mean by information and develop a communications based concept hierarchy and associated characteristics.

3.1 Communications Analysis Requirements

Communications analysis can be complicated by the wide range of issues of interest to users of communication systems, the diversity of technologies and the variety of analysis tools available.

Often there are multiple users, each having requirements involving a composite of information amassed in a combination that is unique to a particular situation, such as that expressed in Figure 1.

As an example, a user, suggested by the icon centre left in Figure 1, will use technology in what they consider appropriate to a given task. A user with a requirement to collect information may correspond with other users through email, or search news and WWW repositories. The selection may be based upon an application perspective (e.g. email, news or the WWW), or perhaps a search perspective (e.g. bookmarks or the WWW). In either case, the user is presented with a relatively simple choice. However, often users do not fully appreciate that the communications resources are supported by diverse and complex communications systems. The decision as to which technology to use is moderated by a wide variety of factors including cultural and personal preferences, experience and training, corporate rules and processes, as well as the availability and applicability of technology. This situation is likely to be reflected by users through a broad mixture of requirements including what they need, what they want, what they think will do the job, and what they know about technology and services. Technically, these requirements are many, varied, and not well known or understood. Requirements from the users may be ambiguous and conflicting, and potentially users are either unaware, or unable to communicate their complete set of requirements (Gause & Weinberg 1989, Thayer & Dorfman 1995).

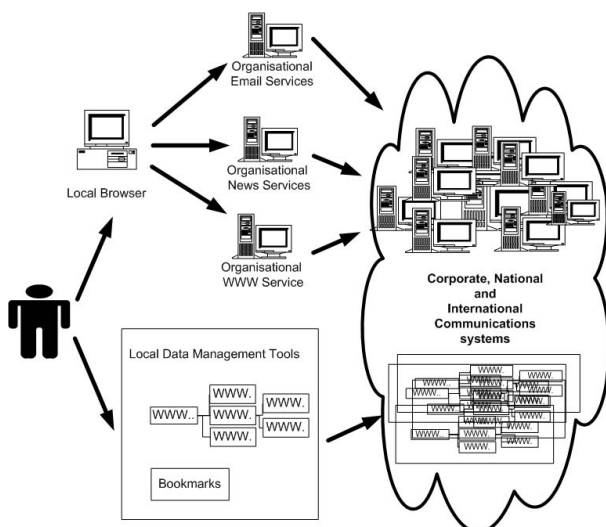


Figure 1: Computer Communications Application Diversity

In order to work within this uncertainty, we suggest a three step approach. First, we resolve what we mean by information, and then provide a set of communications systems characteristics suitable for the

development of a concept hierarchy. Finally we suggest how these characteristics can support the communications analyst.

3.2 Information

A basic problem in establishing a concept hierarchy for communications is finding an adequate description of information. On the one hand, the description must cover the broad range of communication related issues encountered within organisations. On the other hand it must be technically precise enough to specify a hierarchy.

Communication studies is a broad subject covering a multitude of modes of human communication from a largely social/psychological perspective. This field is particularly useful for our work by providing a context within which a range of communications can be described within a common framework.

Within communications studies, the foundation concepts in information theory (Fiske 1990) provide a useful starting point. Unfortunately, these concepts are fundamentally contentious because they focus on the characteristics of information systems and not the consequence of the exchange. The following quote highlights this issue:

Shannon and Weaver's engineering and mathematical background shows in their emphasis. In the design of a telephone system, the critical factor is the amount of signals it can carry. What people actually say is irrelevant.

(Fiske 1990, p.10)

In essence, their theory is considered as a technical example within the group of fundamental communications theories. Within this paper, we accept the technical emphasis. In fact, it is the technical aspects that form the basis of our communications model. We also suggest that analysis based on technical measurements may help to frame investigations into other, less tangible communications occurring within an organisation.

In summary, the communications system proposed as the basis for Shannon's mathematical description of communications (Shannon 2001) is useful in the broader analysis of communications and is also technically specific enough to develop a foundation set of communications related characteristics. The system supports a type of information that is tangible, it flows through channels (which are carried by bearers), it has a source and a destination. A relevant communications system is shown in Figure 2. The source and destination are identifiable components within an organisation (e.g. people, organisational groups and various forms of technology). The bearer represents a medium capable of carrying information. Within the figure, the perspective, and thereby the type of query (shown as ?), is dictated by the type of object (user, bearer, technology). This includes the type of information associated with an object, how the query should be constructed, and the types of responses.

3.3 Concept Hierarchy

Given the above communications system model, basic types of questions are as follows:

- what are the information bearers?
- how are source and destination represented? and
- what constitutes the tangibility of the information?

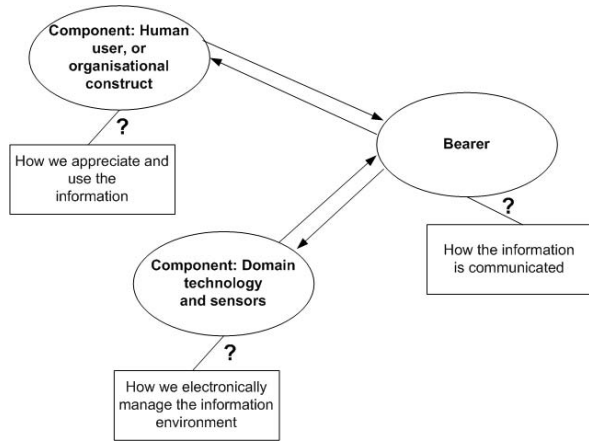


Figure 2: Communications System

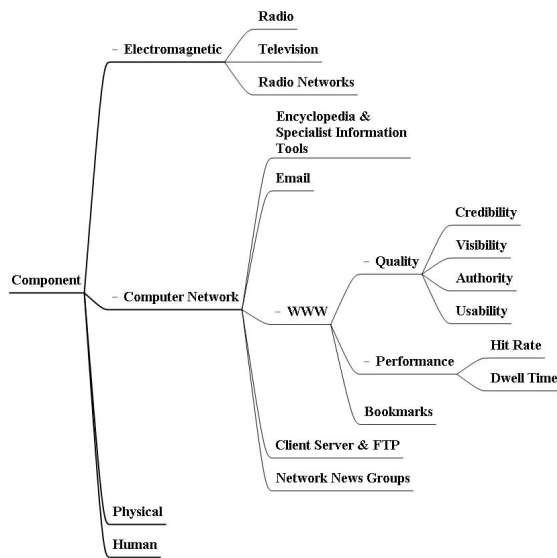


Figure 3: Example Concept Hierarchy

This information can be drawn from the communications models of Shannon (1948) or Lasswell (1948), from technical descriptions of the various technologies (such as computer networking, television or radio), and from the users requirements. Below we provide a description of the information related characteristics: bearers, addressing, and tangibility. We use the measures volume and time to represent the concept of tangibility.

3.3.1 Bearers

A detailed study of users and their target environment would normally be required to fully consider all the communications options. In this paper, a subset of characteristics regarding the bearers associated with a given user are considered, these are shown in Figure 3. The principal information bearer types are the computer network, physical, human and electromagnetic. Instances of bearers within these types (such as television, email or the WWW) would be expected to have representative systems within many organisations.

Measures, such as the availability or relevance of the WWW or television, are likely to be dependent on particular aspects of an organisation. Detailed metrics, including the quality or performance of a particular medium, are very likely to be situation de-

pendant. Consequently, representation for these measures is based upon intrinsic information.

The example concept hierarchy provides a model for the relationship between the bearer types and a subset of classes of characteristics (namely addressing, volume and time). At the level of the bearers, all share these classes of characteristics. At lower levels (e.g. specific channels), this is no longer necessarily the case. The important issue is that a communications analyst is provided with a representative class of bearers and associated characteristics for the particular aspects of a study. In the guided knowledge model of Ceglar & Roddick (2007), this would also have an interactive nature where the class of characteristics (i.e. guidance) would be generated in response to the current model of the underlying dataset (i.e. streaming).

The visualisations in the next section build upon the extrinsic information by including example intrinsic data.

3.3.2 Addressing

Within our communications model, information is treated as if it is contained within a discrete parcel while being transferred across the bearer. The characteristics of interest are either explicit (such as the parcel addressing) or evident from the bearer (such as a *television* program). In either case, the parcel has a defined source and destination. Various types of source and destination may be considered, and the communication may be point-to-point, multi-cast or broadcast. The model does not prescribe an address format, only that components are uniquely addressable.

3.3.3 Volume

The information parcel exists within a discrete space. Volume is used in the sense of the amount of information that may be contained within the parcel, and also to highlight the multidimensional nature of the communications data. There are various measures associated with the volume of information, ranging from information theoretic to simply registering the length of transmission. A parcel could have multiple measures, for example, a physical package, a number of bits, a length of transmission, or an information density. A data recording tape has all of these.

3.3.4 Time

The parcel exists in time and may have various time-stamps (e.g. when it was created, sent or detected). In addition to a time-stamp, the parcel has a duration indicating the time over which the parcel occupied a bearer.

3.4 Organisational Communications Data

Lower level computer communications information is often stored in logs at rates that present a substantial burden on automated computer based analysis. For example, WWW site contact logs (kept for caching or audit purposes) tend to be stored for days, or even weeks on machines at service providers. Each contact may include: the source and destination identifiers, page identifiers, page metrics, time stamps etc. For a busy server, such as might be found in a commercial WWW service provider or a large enterprise, this may account for a number of megabytes of text per day⁵. Longitudinal analysis may require the data to

⁵Garofalakis & Rastogi (2001), Cáceres et al. (2000) and others, discuss the network data volume issue further.

be held for extended periods of time. As an example, to determine whether to cache or mirror a group of sites, it is beneficial to monitor the traffic to the site for long enough to capture not only the steady state patterns but also transients (due to business hours, public holidays and special events).

Table 1 shows a snippet of the type of information that may be represented in such a communications dataset - information source (Src) and destination (Dest), classification details of the Bearer, and the time stamps (Time) associated with the communication. In addition, the dataset may be linked to supporting information such as the name and address (e.g. physical, geographic, business) associated with the source or destination.

Src	Dest	Bearer	Time	
			Start	End
C1	C2	Email	0900	0910
C1	XX	WWW	0905	1020
C3	C5	Phone	0915	0917
C1	C3	Email	0920	0925
⋮	⋮	⋮	⋮	⋮

Table 1: Communication Dataset Snippet

It is highlighted at this point, that even given the broad range of communications options (Email, WWW, Phone) the characteristic types (source, destination, bearer, time) are all the same. The remainder of this paper considers whether such a composite dataset can support communications analysis.

3.5 Communications Analysis

The analysis considered in this paper is largely transitive. Exploratory data mining provides an intermediate step to more specialised analysis. Important functions include, highlighting potentially valuable data, helping to weed out non-useful information, and facilitating the identification of outliers. Numerical measures include traffic aggregates, higher level metrics supporting mean value analysis (qv. Jain 1991), as well as support and confidence.

The ability to substantiate results and search for supporting data is important in the analytical process. Similarly, it is valuable to visualise a volume of information while at the same time provide insight into the detailed events that support the observation. A single transaction of considerable importance may be buried within a mass of day-to-day communications. The ability to drill down from multiple perspectives into a dataset may help to detect such occurrences. A set of interrelated visualisations, described in detail in Section 4, are designed to facilitate the observation of relationships in the gross system whilst allowing inspection of the underlying data. This is analogous to navigating from an aggregate measure (such as support or confidence) into the raw data, facilitating detailed inspection and verification.

As noted above, the ability to interactively query and form visualisations is important to DMKD. Similarly, visualisations generated on the basis of selected samples of a dataset can provide insight into a communications system. Due to the significant rate of change, and the wide variety of data types, the ability to form samples must be flexible. Experience with the dataset (such as through exploration), and a good understanding of the communications discipline should guide the formation of queries.

4 Visualisation

The transactional representation of communications records (see Table 1), in conjunction with information volume and collection frequency issues resonates strongly with areas of support from DMKD. Equally important, as discussed above, these factors highlight the need to support an exploratory perspective.

A composite exploratory data mining and visualisation based query model is described in this section. We term this composite model the *exploration map*. The objectives of the model are to:

- provide a multi-perspective representation of communications data,
- support exploration and navigation,
- focus on particular aspects of communications systems, and
- support the query of datasets.

The individual visualisations (called *organisation*, *component* and *bearer*), are described following the exploration map.

4.1 Exploration Map for Data Mining in Organisational Communications

Figure 4 shows the exploration map. This is an overview of the inter-relations between the visualisations and a dataset. The dataset comprises a list of communications related data in transactional format (as shown in Table 1). Views are generated automatically from the dataset. Interaction with the views is intended to support a comparative assessment across the communications records, through graphically guided query and navigation.

An example scenario would involve :

1. generating the views from the dataset,
2. sighting a level or type of relationship (such as highly clustered) between components in either the component or organisation views,
3. selecting the individual components to query the dataset, and
4. generating and presenting the bearer view.

The bearer view is then used to further query the dataset. In addition, the dataset may be queried directly to filter and focus attention on a subset of the data. This reduced dataset can then be used for further visual explorations.

Visualisations are presented in Figures 5 through 7. Figure 5 provides an organisation centric view, Figure 6 presents a component centric view, and Figure 7 provides a bearer centric view. Aspects of association mining, clustering and classification are described in conjunction with these visualisations. A description of additional diagram attributes that may be associated with the visualisations is also provided, at the end of this section. Figure 8 shows an overall scenario for interaction between the views and the dataset.

4.2 Organisation and Component Centric Visualisation

Figures 5 and 6 provide graph based visualisations of connections (arcs) between components. These views focus on clustering and association for records within a dataset. Components are identified by the source (C_i) or destination (C_j) address. An arc (C_i to C_j) is drawn if the pair exist on a single record.

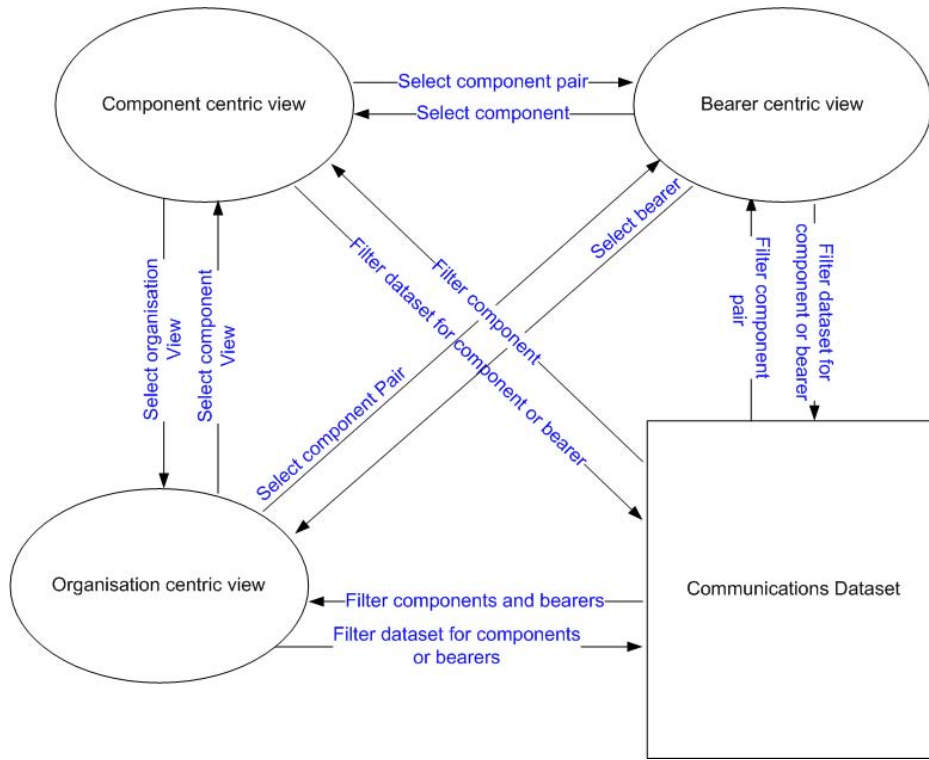


Figure 4: Exploration Map

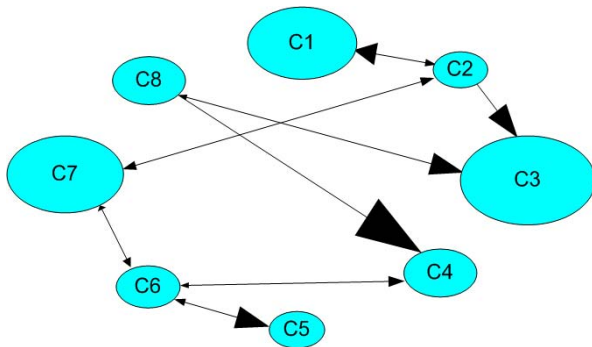


Figure 5: Organisation Centric View

lights aspects of clustering, concentration and isolation. This capability is important as the number of arcs can increase exponentially with the number of components, and may quickly obscure details of the visualisation.

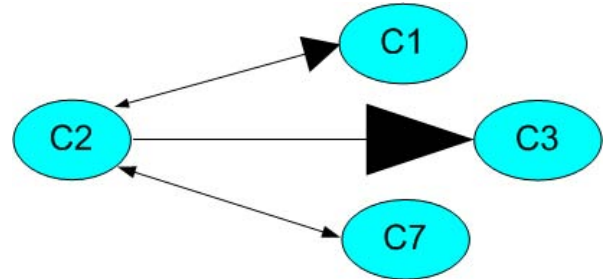


Figure 6: Component Centric View

Figure 5 shows the relationships between communicating components in an organisational view. The key features of this view are that it captures the complete set of components and their interrelationships as represented in the dataset. In particular, it highlights areas of concentration (many to 1) and isolation (none to 1). Data mining measures, including support and confidence, assist in substantiating any inferences.

Arrows are shown on the arcs. The direction is from source to destination, as observed in the dataset. The size of the arrows represents the relative frequency of the occurrence of the directed tuple. This provides a visual indication of the support for the two relations (i.e. $C_i \rightarrow C_j$ and $C_j \rightarrow C_i$) implicit in the connection.

The selection of an individual component (e.g. C2) queries the dataset for a specific information. The selection of an arc between two components (e.g. C1 and C2) queries the dataset and generates the bearer centric view (Figure 7).

Figure 6 provides a component centric view of a subset of the components shown in Figure 5. This view provides a means to reduce clutter and focus on specific components. It provides filtering and high-

These graph based visualisations convey a mixture of communications and data mining measures. Specific data mining tools, namely, association mining, clustering and classification, are described next. These tools are described with regard to how they support communications analysis within the context of the visualisations. Areas where additional support may be gained from the tools are also noted. In this sense, this section highlights areas of future work. An example, introducing scope for additional measures, is outlined in Section 4.4. Similarly, data mining tools are again noted with regard to the bearer centric visualisation (Section 4.3).

Association Mining : A measure of relative support is presented with relationships by varying the size of the arrows. Similarly, the thickness of the arcs could be used to represent confidence. In this sense the layout could be transformed into a Rule Graph (such as presented that by Ceglar, Roddick & Calder (2003)). Care must

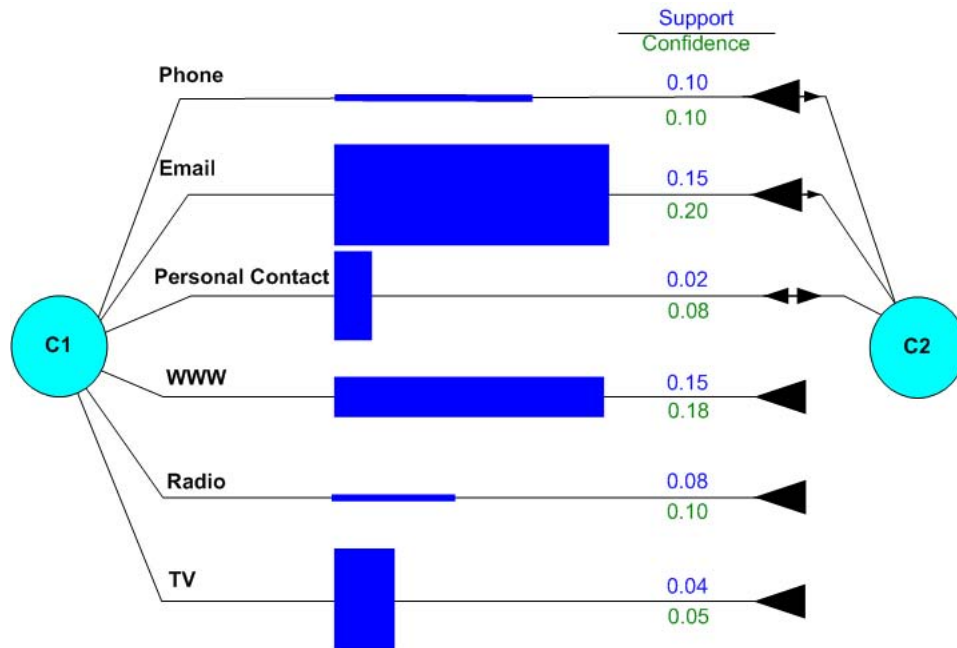


Figure 7: Bearer Centric View - showing the most significant 6 of 50 bearers

be exercised in interpreting the graph. In this instance, the length of a path (hop count) or size of an itemset cannot be assumed to be any greater than two. Inferring longer paths (or larger itemsets) could be problematic due to difficulties in adequately expressing where a given path (or itemset) starts and finishes. Sequential pattern mining applications such as INTEM (INTERacting Episode Miner and viewer) may be valuable in constructing longer path lengths (Mooney & Roddick 2004, 2006).

Clustering : Figures 5 and 6 show characteristics of connection clustering, where the number of connections attached to individual components potentially infers relationships. Other aspects of clustering can be represented within the dataset. The physical location of components could be used to construct 2D spatial clusters. The organisational or component views could also be layered over the clusters⁶. A visualisation comprising the logical clustering (based on connectivity, shown in Figures 5 and 6), overlaid upon the physical clustering (based on location) could provide important information about the efficiency of a current communications system design. For example, an overlaid perspective would support an analyst in considering whether communications are generally within or between components. Such knowledge may support network or business reengineering to more effectively utilise local services.

Classification : A high level classification scheme based on the concept hierarchy (see Figure 3) is implicit in the organisational and component views. In conjunction with clustering, a classification scheme could also be applied to highlight the organisational structure or logical business units associated with components. To extend the above example, a classification scheme in conjunction with physical and logical clustering

could provide information on whether communication services are efficiently deployed to the relevant business units. An analyst, for example, could use a composite representation to assess whether business processes (such as accounting) are appropriately physically distributed.

The characteristics of organisations and components are generic, essentially only requiring that an organisation is composed of uniquely identifiable (and therefore addressable) components. The visualisations could be extended to show relationships between higher level organisations. As an example, an analyst may investigate areas of concentration and isolation between organisations in order to better appreciate preferred modes of inter-organisational communications.

4.3 Bearer Centric Visualisation

The representations in Figures 5 and 6 can easily become cluttered, and do not provide insight into the different bearers and the volume of the information. In order to overcome these limitations, the bearer centric view (Figure 7) provides an alternative perspective. The visualisation is primarily intended to provide a two dimensional histogram of the volume of information flowing between two components.

The components (C1 and C2) are the source and destination of the information. The net set of communications is partitioned across the bearers: Phone, Email, Personal Contact, etc.

The individual bars in the histogram present the relative volume of information in the dataset. The horizontal length of the bars represents the relative frequency of communication; the longer the bar, the higher the relative frequency. The thickness of the bars represents a measure of the relative quantity of information, such as the length of an email, or the time spent at a WWW site. Together, these may suggest a relative preference of a bearer by a component.

The view also shows the support and confidence results for the association *source, destination* \rightarrow *bearer*, (where \rightarrow represents *coincidental with*), for each bearer.

⁶Geographic Information Systems (GIS) systems such as ArcView are designed to support this type of overlay presentation and analysis. This could be developed to provide a geographic DMKD capability as described in Miller & Han (2001).

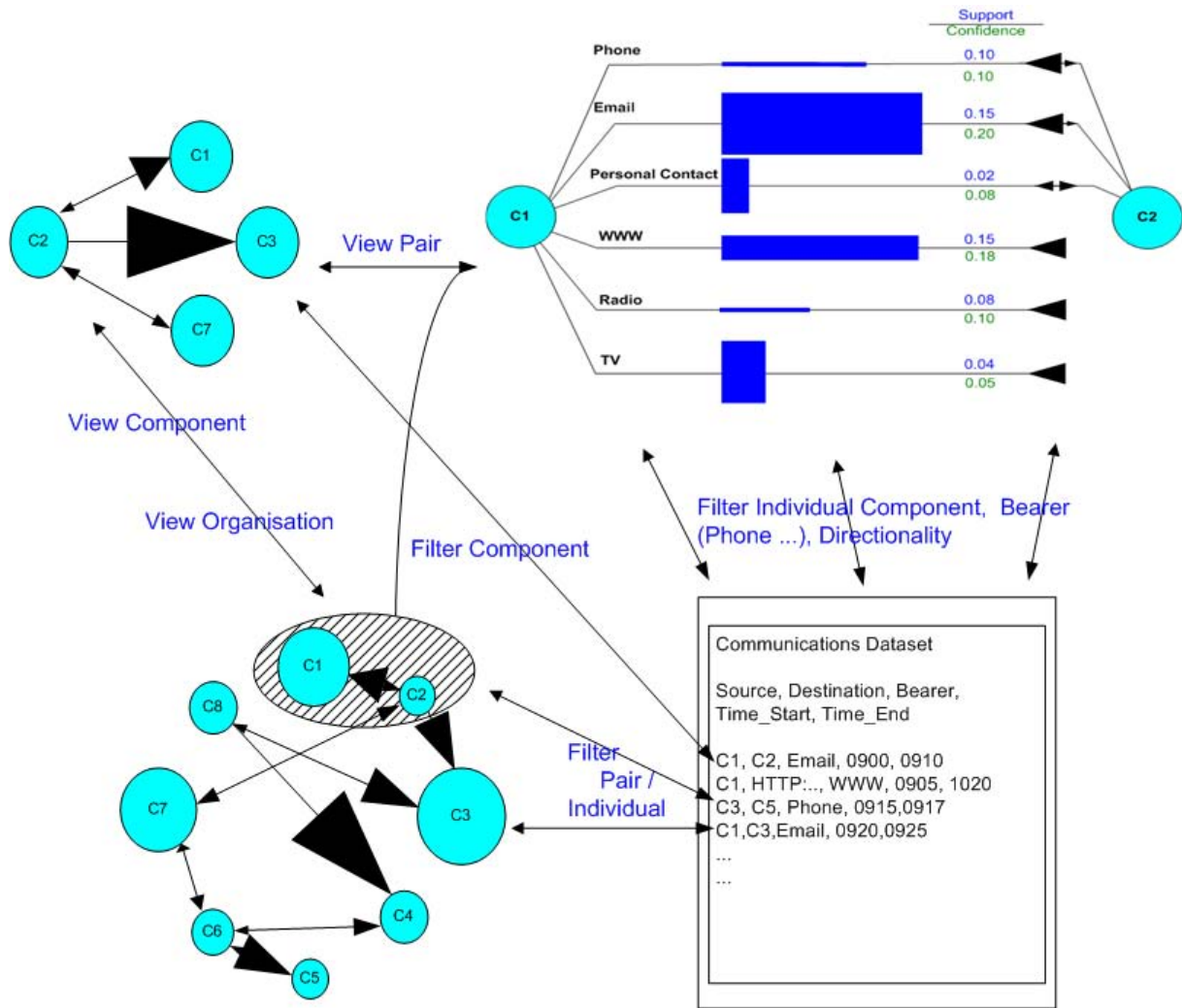


Figure 8: Integrated View

Finally, the arrows represent the overall directionality of the communications, C1 to C2, or C2 to C1. Note that broadcast media such as television, radio, and the WWW are unidirectional, and only involve the principle component (C1) as the destination. The size of the arrows represents the relative frequency of the occurrence of the directed tuple. As with the organisation and component views, the comparative size of the arrows is indicative of both the relative support in the dataset and the demand on the bearer.

The principal queries from this visualisation are the source and destination, the bearers, and the information volume. The selection of a single component (e.g. C1 or C2) or a bearer (e.g. Phone) raises either the organisation or component centric views. The component view is focussed on the selected component. If a specific bearer is selected, the dataset is filtered for the bearer before the organisation or components views are raised. Consequently, these views display only components associated with the selected bearer.

The measures, support and confidence, are introduced to blend the communications with the data mining perspectives. Further support from data mining (through the association mining, clustering and classification tools), as described next, includes additional confidence measures and higher order clustering and classification of bearers.

Association Mining : The bearer centric view is essentially an association representing the type,

volume and direction of information flowing between two components.

The support measure, in combination with the volume, represents the frequency with which the combination occurs in the dataset. From a communications perspective, this provides an indication of the communications load associated with a specific bearer. From a data mining perspective, this measure provides an indication of how well the relation is supported within the dataset. An additional measure of interest would be support and confidence for the directionality. This could be valuable in locating data servers and mirror sites.

Clustering : Clustering can be applied to the bearer set, the directionality, and the importance. Within the context of components (C_i and C_j), clusters represent preferred modes of communications. In contrast to the implicit assumption of independent bearers in the association mining, a cluster of bearers may represent a preferred grouping of communications methods.

Classification : Hierarchical classification could be applied to bearers, such that sub-classes of bearers are represented. This may lessen the current restriction of only having bearers associated through components. This is equivalent to including more branches of the concept hierarchy in the bearer view. For example, rather than

Example Mapping	
Visual Attributes	Knowledge Attributes
Size of the nodes	Relative support for the component
Colour of nodes	Importance/priority of component
Size of Arrows	Relative frequency of the direction of distribution
Colour of arrows	Relative importance of communications

Table 2: Example Attribute Mapping

Email and WWW as two separate classes, a super class of *computer networks* could be used. Sub-classes of the computer network would include not only Email and WWW but also other forms of Internet applications, such as network news, and corporate TCP/IP applications.

4.4 Knowledge Visualisation Attributes

The size of the nodes, thickness of the lines, arrows and colours are, and could be further, used to indicate different characteristics of the relationships within the visualisations. An example mapping is provided in Table 2. Visualisation attributes are those associated with the shapes, such as lines and ellipses, presented in Figures 4 through 7. Knowledge attributes are associated with the visualisations through the concept hierarchy. Not all the attributes are relevant for all the different visualisations, either because an attribute does not contribute to the relationship, or because the data type is not available in the dataset.

The visualisations currently indicate relative directionality for communications through arrows. From a communications perspective the visualisations could be made more instructive if they included the priority or importance of components or their relationships. The relative importance of a component, as may be ascertained from an organisational chart, could be encoded into the colour or size of the ellipses. Similarly, the colour arrows could be used to represent the importance or priorities associated with relationships.

5 Conclusions and Further Work

This paper has presented a discussion on exploratory data mining within the context of organisational communications. Analysis of communications data is complicated by a variety of factors ranging from ambiguity over the meanings of communications and information, to issues with collection rates and complex data interrelations. In order to work within this complexity we have developed a model incorporating problem specification, visualisation and data mining. Developing a problem specification incorporating a model for organisational communications and concept hierarchy has allowed us to form a basis for comparison across a diverse set of communications bearers. Visualisation has supported the integrated representation of communications characteristics from a variety of perspectives. This technique also appears to blend well with core data mining tools and measures. The value of an exploratory perspective in the analysis of communications data has been discussed. These elements work together to improve the potential to better appreciate organisational communications datasets.

An exploratory data mining GUI which combines visualisation with data mining tools has been described. The strengths of our proposal are the ability to use communications data from a broad range of sources, and to work efficiently through potentially very large datasets. The weakness is the inability to directly support specialised domain analysis.

The overall objective of communications analysis is to answer particular questions, or focus on a small set of issues, such as which is the preferred service, phone or email, or which WWW sites should be preferentially supported. Exploratory mining may lead an analyst to appreciate where the answer to these questions lie. Additional tools may be required for the development and presentation of results.

Several areas of further work may be considered. Enhancements to the current visualisations have been suggested, either associated with specific data mining tools or as attributes. Classification based views, such as organisational charts and product/work breakdown structures, may be necessary to provide a more complete picture. These would be included as essentially components within the exploration map toolkit.

Overall, we envisage, as further work, implementing the GUI as an interface between the mass of basic communications data and higher-level analysis tools. As an example, the GUI as described in this paper provides visually based association, clustering, and classification of representations of communications data. Tools such as ArcView may better support the final presentation. The objective is to provide mechanisms to export from a dataset into ArcView. In this way the data mining supports the context based discovery, selection and isolation of relevant data, the application of communication domain specific requirements to the mining activity, and the translation of data from communications datasets into a format for third party tools.

References

- Albert, R. & Barabási, A. (2002), 'Statistical mechanics of complex networks', *Review of Modern Physics* 74(1), 47–97.
- Ankerst, M. (2002), The perfect data mining tool: Automated or interactive?, in 'Panel at ACM SIGKDD02', ACM, Edmonton, Canada.
- Bryson, L. (1948), *The communication of ideas: Religion and civilization series*, Harper and Row, New York.
- Cáceres, R., Duffield, N., Feldmann, A., Friedmann, J., Greenberg, A., Greer, R., Johnson, T., Kalmanek, C., Krishnamurthy, B., Lavelle, D., Mishra, P. P., Ramakrishnan, K., Rexford, J., True, F. & van der Merwe, J. (2000), 'Measurement and analysis of IP network usage and behavior', *IEEE Communications* 38(5), 144–151.
- Card, S. K., Mackinlay, J. D. & Schneiderman, B. (1999), *Readings in Information Visualization: Using Vision to Think*, Morgan Kaufmann.
- Ceglar, A. & Roddick, J. F. (2007), 'GAM - a guidance enabled association mining environment', *International Journal of Business Intelligence and Data Mining* 2(1), 3–28.
- Ceglar, A., Roddick, J. F. & Calder, P. (2003), Guiding knowledge discovery through interactive data mining, in P. Pendharkar, ed., 'Managing Data

- Mining Technologies in Organisations: Techniques and Applications', Idea Group Pub., Hershey, PA, pp. 45–87. Ch. 4.
- Ceglar, A., Roddick, J. F., Mooney, C. H. & Calder, P. (2003), From rule visualisation to guided knowledge discovery, in S. Simoff, G. Williams & M. Hegland, eds, '2nd Australasian Data Mining Workshop (AusDM'03)', UTS, Canberra, pp. 59–94.
- Chapman, P., Kerber, R., Clinton, J., Khabaza, T., Reinartz, T. & Wirth, R. (1999), The CRISP-DM process model, Discussion paper, CRISP-DM Consortium.
- Dodge, M. & Kitchin, D. R. (2001a), *Mapping Cyberspace*, Routledge.
- Dodge, M. & Kitchin, R. (2001b), *Atlas of cyberspace*, Addison-Wesley, New York.
- El-Shinnawy, M. & Markus, M. (1997), 'The poverty of media richness theory: explaining people's choice of electronic mail vs. voice mail', *International Journal of Human-Computer Studies* **46**(4), 443–467.
- Fiske, J. (1990), *Introduction to Communication Studies*, Routledge.
- Garofalakis, M. & Rastogi, R. (2001), 'Data mining meets network management: The nemesis project', *ACM SIGMOD International Workshop on Research Issues in Data Mining and Knowledge Discovery*.
- Gause, D. C. & Weinberg, G. M. (1989), *Exploring Requirements: Quality Before Design*, Dorset House.
- Geng, L. & Hamilton, H. J. (2006), 'Interestingness measures for data mining: A survey', *ACM Computing Surveys* **38**(3).
- Hao, M. C., Dayal, U., Hsu, M., Sprenger, T. & Gross, M. H. (2001), Visualization of directed associations in e-commerce transaction data, in 'VisSym'01, Joint Eurographics - IEEE TCVG Symposium on Visualization', IEEE Press, Ascona, Switzerland, pp. 185–192.
- Hofman, H., Siebes, A. P. & Wilhelm, A. F. (2000), Visualizing association rules with interactive mosaic plots, in '6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', ACM, Boston, MA, USA, pp. 227–235.
- Hulten, G., Spencer, L. & Domingos, P. (2001), Mining time-changing data streams, in '7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'01)', ACM Press, San Francisco, CA, USA, pp. 97–106.
- Jain, R. (1991), *The art of computer systems performance analysis*, Wiley.
- Julisch, K. & Dacier, M. (2002), Mining intrusion detection alarms for actionable knowledge, in '8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', ACM Press, Edmonton, Alberta, Canada, pp. 366–375.
- Keller, P. R. & Keller, M. M. (1993), *Visual cues: practical data visualization*, IEEE Computer Society Press.
- Lasswell, H. D. (1948), The structure and function of communication in society, in '(Bryson 1948)', pp. 37–51.
- Michailidis, A. & Rada, R. (1997), 'Activities and communication modes', *International Journal of Human-Computer Studies* **46**(4), 469–483.
- Miller, H. & Han, J., eds (2001), *Geographic Data Mining and Knowledge Discovery*, Research Monographs in Geographic Information Systems, Taylor and Francis, London.
- Mooney, C. H. & Roddick, J. F. (2004), Mining relationships between interacting episodes, in M. Berry, U. Dayal, C. Kamath & D. Skillicorn, eds, '4th SIAM International Conference on Data Mining (SDM'04)', SIAM, Lake Buena Vista, Florida.
- Mooney, C. H. & Roddick, J. F. (2006), Marking time in sequence mining, in P. Christen, P. Kennedy, J. Li, S. Simoff & G. Williams, eds, 'Australasian Data Mining Conference (AusDM 2006)', Vol. 61 of *CRPIT*, ACS, Sydney, NSW, pp. 129–134.
- Ong, K. H., Ong, K. L., Ng, W. K. & Lim, E. P. (2002), Crystalclear: Active visualization of association rules, in 'International Workshop on Active Mining (AM-2002) in Conjunction with the IEEE International Conference on Data Mining (ICDM'02)', IEEE Press, Maebashi City, Japan.
- Rainsford, C. P. & Roddick, J. F. (2000), Visualisation of temporal interval association rules, in '2nd International Conference on Intelligent Data Engineering and Automated Learning, (IDEAL 2000)', Vol. 1983 of *LNCIS*, Springer, Shatin, N.T., Hong Kong, pp. 91–96.
- Shannon, C. (1948), 'A mathematical theory of communication', *Bell System Technical Journal* **27**, 379–423, 623–656.
- Shannon, C. E. (2001), 'A mathematical theory of communication', *ACM SIGMOBILE Mobile Computing and Communications Review* **5**(1), 3–55. Reprinted (with corrections) from (Shannon 1948).
- Stone, G. C., Singletary, M. W. & Richmond, V. P. (1999), *Clarifying Communications Theories: A Hands-on Approach*, Iowa State University Press, Ames, Iowa.
- Thayer, R. H. & Dorfman, M. (1995), *System and software requirements engineering*, IEEE Computer Society Press, Los Alamitos, CA, USA.

Towards Scalable Real-Time Entity Resolution using a Similarity-Aware Inverted Index Approach

Peter Christen¹

Ross Gayler²

¹ Department of Computer Science,
The Australian National University,
Canberra ACT 0200, Australia
Email: peter.christen@anu.edu.au

² Veda Advantage,
Melbourne VIC 3000, Australia
Email: Ross.Gayler@VedaAdvantage.com

Abstract

Most research into entity resolution (also known as record linkage or data matching) has concentrated on the quality of the matching results. In this paper, we focus on matching time and scalability, with the aim to achieve large-scale real-time entity resolution.

Traditional entity resolution techniques have assumed the matching of two static databases. In our networked and online world, however, it is becoming increasingly important for many organisations to be able to conduct entity resolution between a collection of often very large databases and a stream of query or update records. The matching should be done in (near) real-time, and be as automatic and accurate as possible, returning a ranked list of matched records for each given query record. This task therefore becomes similar to querying large document collections, as done for example by Web search engines, however based on a different type of documents: structured database records that, for example, contain personal information, such as names and addresses.

In this paper, we investigate inverted indexing techniques, as commonly used in Web search engines, and employ them for real-time entity resolution. We present two variations of the traditional inverted index approach, aimed at facilitating fast approximate matching. We show encouraging initial results on large real-world data sets, with the inverted index approaches being up-to one hundred times faster than the traditionally used standard blocking approach. However, this improved matching speed currently comes at a cost, in that matching quality for larger data sets can be lower compared to when standard blocking is used, and thus more work is required.

Keywords: Record linkage, data matching, scalability, approximate string comparisons, similarity measures.

1 Introduction

In many application areas, data from different sources often needs to be matched and aggregated before it can be used for further analysis or data mining. The objective of entity resolution is to match all records that relate to the same entity. These entities can, for example, be customers, patients, business names, bibliographic citations, or genome sequences.

Copyright ©2008, Australian Computer Society, Inc. This paper appeared at the Seventh Australasian Data Mining Conference (AusDM 2008), Glenelg, Australia. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 87, John F. Roddick, Jiuyong Li, Peter Christen and Paul Kennedy, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

Techniques for finding and matching records that correspond to the same entity have traditionally been used in the health sector and within the census (Fellegi & Sunter 1969, Winkler 2006). These techniques assume that no unique entity identifiers are available in the databases to be matched. They compare pairs of records using the available partially identifying attributes (such as names, addresses, and dates of birth) and calculate a similarity score for each compared record pair. These similarity scores are then used to classify record pairs as matches, non-matches, or possible matches, in which case manual clerical review is required to decide the final match status (Christen & Goiser 2007, Gu & Baxter 2006). The matching process is usually quite challenging, because real world databases contain dirty data, such as missing or out-of-date attribute values, variations and (typographical) errors, or even data that is coded differently.

Entity resolution techniques are increasingly being used within and between many organisations in the public and private sectors to improve data management, processing and analysis. Example applications include finding duplicates in business mailing lists and bibliographic databases (online libraries); crime and fraud detection within finance and insurance companies, as well as government agencies; compilation of longitudinal data for research; and assembly of terrorism watch lists for national security. And while statisticians and epidemiologists speak of record or data linkage, computer scientists and the database and business oriented IT communities name the same task as entity resolution, data or field matching, data cleansing, data integration, duplicate detection, ETL (extraction, transformation and loading), object identification, or merge/purge processing.

Entity resolution is particularly important in consumer financial services, especially when the services are remotely delivered. After an account is established, the consumer is normally required to use an unambiguous identity token, like an account number. However, initially establishing the consumer's identity is difficult even in countries with national identity tokens. The usual approach taken is entity resolution of identifying information, as provided by the consumer, against one or more databases of related identifying information. The information provided by a consumer is often subject to variability and error, so the matching process must be approximate.

An example of this type of application of entity resolution is accessing a consumer's credit history at a credit bureau. The financial institution forwards the identifying attributes supplied by the consumer, and the credit bureau resolves the identity to a pre-existing record on its database, and returns the matching credit history. In a developed economy,

most of the population will have a credit history, so a credit bureau could easily have tens to hundreds of millions of records, and tens to hundreds of thousands of enquiries per day. Most of these enquiries will be driven by automated systems that require a sub-second response from the credit bureau. The major technical challenges for such a system are therefore automated and accurate matching, scalability, and real-time entity resolution.

In recent years, research into entity resolution has been conducted in various fields, including machine learning, data mining, information retrieval, artificial intelligence, and the database community (Christen & Goiser 2007, Elmagarmid et al. 2007). The newly developed techniques can be classified into learning based approaches (Bhattacharya & Getoor 2007, Cohen & Richman 2002, Cohen et al. 2003, Elfeky et al. 2002), or database and graph-based methods (Dong et al. 2005, Kalashnikov & Mehrotra 2006, Weis & Naumann 2007, Yin et al. 2006). Most research has focussed on the quality of the entity resolution results, i.e. the accuracy of classifying record pairs into matches and non-matches, but not on scalability, nor on automated or real-time matching.

Matching two databases potentially requires each record from one database to be compared with all records from the other database. Thus, comparing all pairs is computationally not feasible for very large databases (Christen & Goiser 2007). Indexing techniques, also known as *blocking*, are applied to reduce the number of record pair comparisons (Baxter et al. 2003), at the cost of potentially missing some true matches. These techniques work by splitting the databases into blocks according to some criteria, and only comparing the records within each block with each other. A blocking criterion, also called *blocking key*, could simply be a record attribute that contains values of high quality (for example postcodes), it could be the concatenation of several attribute values, or even be phonetically encoded attribute values, in order to group similar sounding values into the same block (Christen 2006).

Most blocking techniques have two drawbacks. First, the size of the generated blocks depends upon the frequency distribution of the record values used as blocking keys. For example, using a 'surname' attribute will usually generate a very large block containing the surname 'Smith', resulting in a very large number of comparisons to be done for this block. Second, if a value in a record attribute used as blocking key contains errors or variations that result in a differently encoded blocking key value, then the corresponding record will be inserted into a different block and true matches will be missed. This problem can be overcome by having two or more different blocking keys based on different record attributes.

Most publications in the field of entity resolution present experimental results based on only small to medium sized data sets. The computational complexity of many of the recently developed advanced entity resolution approaches currently makes them unsuitable for very large databases containing many millions of records. Additionally, with the exception of one approach (Bhattacharya & Getoor 2007), all techniques developed so far assume that the databases to be matched are static, and that the matching process is done off-line in batch mode. The following list shows experimental timing results of four recent state-of-the-art entity resolution approaches:

- (Bhattacharya & Getoor 2007): 831,991 records, 31 seconds matching time (for one query record). This approach will be discussed in more details in Section 2.

- (Kalashnikov & Mehrotra 2006): 75,000 records, 180 seconds matching time (for the complete data set).
- (Weis & Naumann 2007): 1,000,000 records, 24,433 seconds matching time (for the complete data set).
- (Yin et al. 2006): 100,000 records, 1,534 seconds matching time (for the complete data set).

As can be seen, none of these approaches will allow scalable real-time entity resolution of large databases. Thus, there is a need for research into the development of such techniques, and this paper is a first step in this direction. Specifically, we investigate the use of inverted index techniques, as commonly used in the field of information retrieval for large-scale Web search engines (Zobel & Moffat 2006). With the popularity and commercial success of such search engines in the past decade, there has been a large amount of research on optimisation techniques for such applications (Bayardo et al. 2007). Our work is aimed at applying such optimisations to the task of real-time entity resolution of very large databases.

The main objective of real-time entity resolution is to process a stream of query records as quickly as possible, and to match them to one or several (large) databases that contain existing entities, and possibly to a range of external data sources that contain additional information that can be used for verification of the matched entities. The response time for matching a single query record has to be as short as possible (ideally sub-second), and the matching technique must scale-up efficiently to very large databases containing many millions of records. In addition, such techniques should generate a match score that indicates the probability that a matched record in the database refers to the same entity as the query record.

We are only aware of one publication that discusses query-time entity resolution (Bhattacharya & Getoor 2007). However, as the above list shows, the matching time of this approach for a single query record is more than 30 seconds on a medium sized database, making the approach impractical on very large databases. Scalability and real-time deduplication (i.e. entity resolution within one database) have been investigated by the information retrieval community in the context of search engines operating on very large document collections (Conrad et al. 2003). The objective in these applications is to remove duplicate documents returned by a search query. The basic idea is to calculate condensed document representations (called 'signatures' or 'fingerprints'), for example on the most and least common document features. Documents that have the same signatures are then assumed to be duplicates of each other.

The rest of this paper is structured as follows. Work related to our research is discussed in the following section. In Section 3, the three indexing techniques under investigation are presented in detail, and they are evaluated experimentally in Section 4 using a collection of real-world Australian data sets. The results of these experiments are then discussed in Section 5, and the paper is concluded in Section 6 with an outlook to future work.

2 Related Work

In this section, we present in more detail recent research in the area of indexing and blocking for entity resolution, discuss the query-time entity resolution approach described in (Bhattacharya & Getoor 2007) in more detail, and we give a short overview of research on inverted indexing techniques as developed by the information retrieval community.

Research in indexing and blocking can be classified into two categories. The first considers the development of new and improved indexing techniques aimed at making entity resolution more scalable and more accurate at the same time. Besides the traditional standard blocking approach, to be presented in Section 3 below, new indexing techniques recently developed include:

- Sorted neighbourhood approach (Hernandez & Stolfo 1995)
The idea behind this technique is to sort a database according to the values in the blocking key, and to then slide a window of a certain size over the database and compare all records within the current window with each other. An adaptive sorted neighbourhood approach has recently been proposed that dynamically adjusts the size of the window according to the values in the record attribute used as blocking key (Yan et al. 2007). This technique produced matching results of better quality than both the standard blocking and the basic sorted neighbourhood techniques.
- Q -gram based indexing (Baxter et al. 2003)
This technique aims to allow for ‘fuzzy’ blocking, by converting the blocking key values into lists of q -grams (sub-strings of length q), and, based on sub-lists of these q -gram lists, each record is inserted into several blocks according to a Jaccard-based similarity threshold. While this technique improves entity resolution for data that contains a large proportion of errors and modifications, its computational complexity makes it unsuitable for large databases.
- Canopy clustering (Cohen & Richman 2002)
The idea behind this technique is to use a computationally efficient similarity measure to generate high-dimensional, overlapping clusters (called ‘canopies’), and to then extract blocks of records from these clusters. Each record is inserted into several clusters, again overcoming the problem of (typographical) errors and modifications in the record values.
- String map based indexing (Jin et al. 2003)
Another approach is to map the blocking key values (assumed to be strings) into a high-dimensional Euclidean space in such a way that the distances between all pairs of strings are preserved; followed by finding pairs of objects in this space that are similar to each other. Any multi-dimensional index data structure, such as an R-tree, can be used to efficiently retrieve similar pairs of objects in this high-dimensional space. However, as the dimensionality of this space increases, the efficiency of many index data structures decreases rapidly, and with around 15 to 20 dimensions, in most tree based index structures all objects in the index will be accessed when similarity searches are performed (Aggarwal & Yu 2000).
- Suffix-array indexing (Aizawa & Oyama 2005)
The basic idea of this technique is to insert the blocking key values and their suffixes into a *suffix array* based inverted index. A suffix array contains strings or sequences and their suffixes in an alphabetically sorted order. Similar to canopy clustering, each record might be inserted into several blocks, depending upon the length of their blocking key values. Record pairs will then be formed from all pairs that are in the same inverted index list.

The second area of research into indexing for entity resolution is the development of techniques that learn how to optimally choose the blocking keys, i.e. the record attributes used for indexing. Traditionally, the choice is made manually by domain and entity resolution experts. Recently, two supervised learning based approaches have been developed. They either use predicate-based formulations of learnable indexing functions (Bilenko et al. 2006), or apply the sequential covering algorithm to discover disjunctive sets of rules (Michelson & Knoblock 2006). Both approaches aim to find blocking criteria that maximise the number of true matches while minimising the total number of candidate record pairs generated.

As previously mentioned, we are only aware of one approach that addresses query-time entity resolution (Bhattacharya & Getoor 2007). It is based on an unsupervised relational clustering technique, and assumes that the data contains relational information that links different types of entities. For example, in a census database this could be a ‘household’ attribute that contains values such as ‘father of’ or ‘married to’, that explicitly link two records. Similarities between entities can then be calculated by not just using their record attributes, but also through their connectivity between records. At query time, a graph is built that connects the query record with potential matches in the database, and these matches are then iteratively refined using links as well as attribute similarities between the connected database records. This iterative clustering approach is computationally very expensive, and while it produces better matching results compared to other approaches, it is not scalable to large databases, requiring around 30 seconds for one query record on a database containing around 800,000 records (Bhattacharya & Getoor 2007). It also needs to be clarified that, compared to the other timing results shown in the dot list on the previous page, this query-time entity resolution approach only finds the database records that match to the query entity, but otherwise it leaves the database untouched (i.e. un-resolved). Thus, for each query record, this approach needs to start from scratch.

A large body of work is available on inverted index techniques, mainly in the information retrieval community (Witten et al. 1999, Zobel & Moffat 2006). With the increasing size of document collections, especially the World Wide Web, as well as the intense competition between commercial Web search engines, there has been tremendous interest in developing scalable and fast indexing methods for massive data collections. While not all of the commercially developed techniques are being published, some optimisation techniques have been made publicly available (Bayardo et al. 2007). These techniques are based on ideas such as exploiting similarity thresholds to quickly filter out candidate matches that cannot make it into the final result set, or by exploiting the order of the weighted entries in the inverted index lists in order to avoid having to add new candidates. In our experiments, we have so far only implemented a simple threshold based optimisation, as will be described in the following section.

3 Indexing for Real-Time Entity Resolution

In this section we describe the three indexing methods under investigation. Figures 1 and 2 provide a simple illustrative example of these methods using only one attribute. In real world applications, and in the experiments discussed in Sections 4 and 5 below, normally several attributes would be used, and individual index data structures would be built for each attribute, as will be discussed in more detail below.

Record ID	Surname	Soundex encoding
r1	smith	s530
r2	miller	m460
r3	peter	p360
r4	myler	m460
r5	smyth	s530
r6	millar	m460
r7	smith	s530
r8	miller	m460

Figure 1: Example records with surname values and their Soundex encodings, used to illustrate the three indexing methods in Figure 2.

There are two components that are used in all three indexing methods. First, the similarity between attribute values can be calculated using any similarity function appropriate for the content of an attribute. While we assume that the attribute values are strings and an approximate string comparison function is used to calculate similarities (Christen 2006, Cohen et al. 2003), other possible attribute types could be dates, times, numerical values, or geographic locations. Specific similarity functions are available for all these attribute types (Christen 2008). We assume that all these functions return a normalised numerical similarity value, with 1.0 indicating exact similarity and 0.0 total dissimilarity, and values in-between indicating somewhat similar attribute values.

A second component, commonly used in standard blocking, is a phonetic encoding function (such as Soundex, NYSIIS or Double-Metaphone) (Christen 2006), that groups similar sounding values into the same block. We use such an encoding function in all three indexing methods, and only calculate similarities between values in the same block. In Figure 2 (a), this can be seen explicitly, as the Soundex encoded surname values are the keys of the blocks, while in Figures 2 (b) and (c) this ‘blocking’ is illustrated via the dotted lines that show the similarities between the surname values in the same block.

There are two phases involved in real-time entity resolution. In the first phase, an index is built using a static database containing a large number of records. This database is assumed to be cleaned and deduplicated, such that it contains only one record per real-world entity. In the second phase, we assume a built index is queried by a stream of incoming records, and the aim is to retrieve a ranked list of matches from the index for each query record. If there is a record that refers to the query entity stored in the index, then ideally the top ranked record returned by the index should refer to this entity (i.e. this would be a true match). In the following three sections we describe how both phases are implemented in the three indexing methods, and in Section 3.4 we then describe an optimisation technique that can be applied to these methods at query time.

3.1 Standard Blocking

This method is commonly used in traditional entity resolution when one static database is being deduplicated or two databases are being linked (Baxter et al. 2003). Each record in a database is inserted into one block according to the value of its blocking key. In order to overcome the problem of variations and (typographical) errors in the record attribute values used as blocking keys, and to put similar sounding blocking key values into the same block, phonetic encoding functions are commonly used (Christen 2006). All records that have the same (encoded) blocking key value are inserted into the same block, as shown

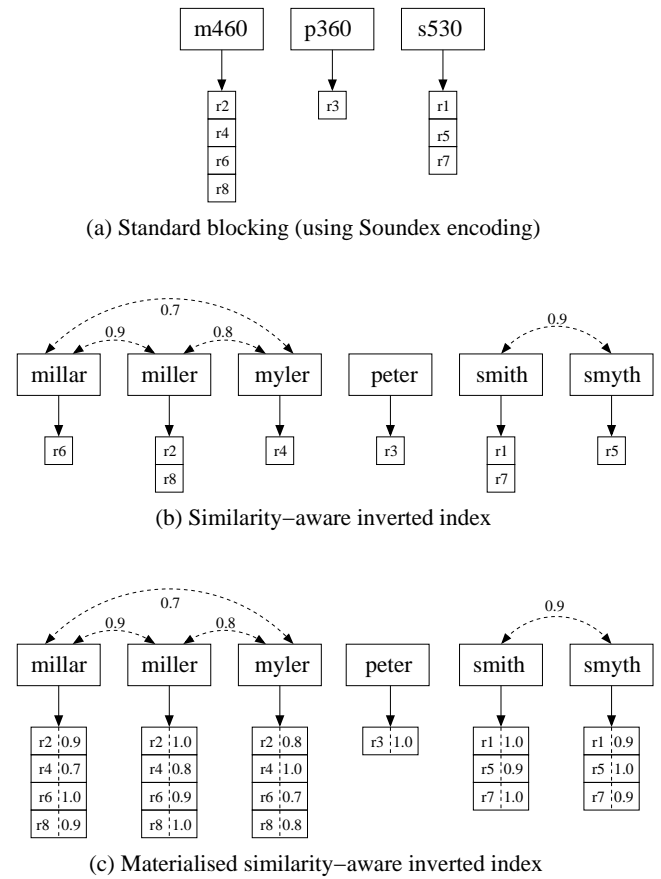


Figure 2: Example indexing methods resulting from the records given in Figure 1.

in Figure 2 (a). Each block can be implemented as an inverted index list, with the key being the (encoded) blocking key value, and the values in the corresponding list being the identifiers of the records in that block. No similarity calculations are performed during the build phase of this indexing method.

At query time, all blocking keys (assuming several attributes are used for the index) are generated from the relevant attribute values taken from the query record, and the record identifiers of all records stored in the same blocks as the query record are retrieved. For example, assuming a query record has the surname value ‘smythe’ (with Soundex encoding ‘s530’), then the list of record identifiers from the ‘s530’ list from Figure 2 (a) will be retrieved from the index, and the query record surname value ‘smythe’ will be compared to the surname values ‘smith’ (‘r1’), ‘smyth’ (‘r5’), and ‘smith’ again (‘r7’). If several indices are built using different attributes as blocking keys (in the experiments in Section 4 three indices were built, one each on the ‘Postcode’, ‘Surname’ and ‘Suburb’ attributes), then for each index the list of record identifiers from the corresponding block is retrieved, the union of these lists is generated, and comparisons are done between the query record and all records in this unified list. This approach can result in a large number of similarity calculations that need to be conducted for each query record.

What is additionally required for the standard blocking index, besides the inverted index based data structures, is that all records in the database can be accessed fast and efficiently via their record identifiers. This is because the actual record values are needed at query time when the similarities are calculated between the query record and the database records that are in the same block as the query record.

Ideally, this would require that the complete database can be stored in main memory, or at least that a fast index can be generated on the record identifier attribute.

3.2 Similarity-Aware Inverted Index

The basic idea of this indexing method is to reduce the number of similarity calculations to be done at query time by storing similarity information in the index data structure, and facilitating efficient access to these similarity values at query time.

With this indexing method, the actual record values are used as keys of the inverted index, rather than the encoded blocking keys. When the index is built using the database records, a standard inverted index is generated for each record attribute that is being used as a blocking key. As illustrated in Figure 2 (b), for each unique value in an attribute, the identifiers of all records that have this value are inserted into one list. Additionally, information about the blocks and the similarities between the values in each block are also stored, as illustrated in Figure 2 (b) using the dotted lines (the values connected by a dotted line are in the same block). Similar to the standard blocking approach, a phonetic encoding method can be used to assign record values into blocks. For example, in Figure 2 (b) the surname values 'millar', 'miller' and 'myler' are in one block, 'smith' and 'smyth' are in a second block, and 'peter' is in a block by itself.

This differs to standard blocking in that each unique record value is stored only once, similar to the way it is stored as a key in the inverted index, and similarity calculations between record values only have to be done once when a new unique value is loaded and processed for the first time from a database record when the inverted index is generated. In this case, its encoding value is generated, the value is inserted into its corresponding block, and the similarities between this new value and all other values in the same block are calculated. At the end of the build phase, the similarities between all values in a block are therefore known and available for quick retrieval at query time.

At query time, the similarity values of possible matches, i.e. the record identifiers of candidates retrieved from an index, are added into an 'accumulator' (Bayardo et al. 2007, Witten et al. 1999, Zobel & Moffat 2006), a list data structure that contains record identifiers and their (partial) similarity values with the query record. If more than one index has been built over several blocking keys, then the similarity values of the candidate record identifiers retrieved from each matched inverted index list are added into the accumulator, and at the end of this process the accumulator is sorted according to the overall similarity values. The elements at the beginning of the accumulator with the largest similarity values are then returned. When a query record is to be matched, the following two cases can happen.

1. A query record value used as blocking key is available as a key in the corresponding inverted index. In this case, all record identifiers from this inverted index list are retrieved first, and inserted into the accumulator with a similarity value of 1.0 (as they correspond to exact matches). Next, the record identifiers of all other values in this block, i.e. the values connected via dotted lists in Figure 2 (b), are retrieved and inserted into the accumulator with their corresponding similarity value (which will be less than 1.0).

For example, using the index from Figure 2 (b), if a query record has a surname value of 'miller',

then first the two record identifiers 'r2' and 'r8' are added into the accumulator with similarity value 1.0, and then the approximate matches would be added: 'r6' with similarity value 0.9, and 'r4' with similarity 0.8. Thus the (unsorted) accumulator would look like this:

$$accu = \begin{array}{|c|c|c|c|} \hline r2 & r4 & r6 & r8 \\ \hline 1.0 & 0.8 & 0.9 & 1.0 \\ \hline \end{array}$$

2. The second case happens when a value in the query record is not available as a key in the inverted index (thus there is no exact matching record for that attribute). In this case, the phonetic encoding of the query record value needs to be calculated first, in order to determine the block where this value belongs to. Then, similar to the process described above, for each record value in this block, the similarity between this value and the query value needs to be calculated, and the corresponding record identifiers of these values will be added into the accumulator with the calculated similarities.

Following the above example, if we assume the query record surname value is 'smithe' (with Soundex encoding 's530'), then the values 'smith' and 'smythe' from the 's530' block are retrieved, the similarities between these two values and 'smithe' are calculated (let us assume they are 0.9 between 'smithe' and 'smith', and 0.7 between 'smithe' and 'smyth') and added into the accumulator, which then looks like this:

$$accu = \begin{array}{|c|c|c|} \hline r1 & r5 & r7 \\ \hline 0.9 & 0.7 & 0.9 \\ \hline \end{array}$$

To summarise the query process, for each attribute that has been used as a blocking key and for which an inverted index has been built, candidate record identifiers are retrieved and their similarity values are inserted into the accumulator, or summed to already existing entries in the accumulator for a candidate record. Thus, in order for a candidate record to have a high overall similarity value it needs to be in the same block as the query record values in all attributes that are used as blocking keys. This approach therefore corresponds to an intersection of the candidate record identifier lists, which is different from standard blocking, where candidate record identifiers are retrieved if at least one of their attribute values is in the same block as the query record value (union of lists). Thus, it is possible that standard blocking retrieves more candidate records than this inverted index approach, and this could possibly result in an increased matching rate for standard blocking.

Finally, the accumulator is sorted according to the total similarity values in it, and the top ranked candidate record identifiers and their similarity values are returned as a ranked list of possible matches.

3.3 Materialised Similarity-Aware Inverted Index

This method is a variation of the similarity-aware inverted index presented in the previous section. The idea behind it is to reduce the number of retrievals of similarity values from different inverted index lists, by inserting them directly into the inverted index lists at build time. This is similar to what is done in information retrieval, where weights, such as term frequencies, are commonly stored in the inverted index lists themselves. As a result, the amount of memory used by the inverted index lists increases significantly,

because redundant information is being stored. However, as this is a more standard approach, it will be better suited to the various optimisation techniques that have been developed for inverted index techniques (Bayardo et al. 2007, Zobel & Moffat 2006).

At build time, the identifier of each record is not only inserted into the inverted index lists of its values (i.e. corresponding to exact matches), but also into the inverted lists of all other record values in the same blocks with their corresponding similarity values, as can be seen in Figure 2 (c).

At query time, there are again two cases that can happen. First, if the value from a query record is available as key in an inverted index, then its list containing record identifiers and similarity values can be retrieved directly and added into the accumulator. In the second case, where a query record value is not available as an inverted index key, the encoding of the query value needs to be generated first, then all values in the same block as the encoding are retrieved, the corresponding similarities are calculated, and then the record identifiers from the relevant inverted index lists are retrieved and added into the accumulator with the calculated similarity values. Apart from this, the query-time process of retrieving, sorting and returning matches and their similarity values is the same as with the basic similarity-aware inverted index presented in Section 3.2 above.

3.4 Optimisations

Various optimisation techniques have been developed for inverted index methods to improve querying an index (Bayardo et al. 2007, Witten et al. 1999, Zobel & Moffat 2006). They are based on ideas such as compression of the index after it has been built, or filtering or sorting of candidates at query time to reduce the amount of computation required.

In our indexing methods, we have implemented a similarity threshold based filtering approach that works as follows. Assume n indices are built (on n different attributes used as blocking keys). If we assume that the similarity calculations are normalised between 1.0 (exact match) and 0.0 (totally different), then the maximum similarity between two records can be $(n \times 1.0) = n$, corresponding to exact matches on all n record attributes that are being compared.

For the inverted index based methods that use an accumulator to store attribute similarities of candidate matches, a minimum total similarity threshold t (with $t < n$) can be used to filter out candidate records that will not reach a total minimum similarity of t . For example, assume three attributes are used and three indices are built (as in the experiments in the following section), so $n = 3$. Let us assume the minimum total similarity threshold has been set to $t = 2.4$. When candidate record identifiers from the first attribute are retrieved from the first index, all candidates with a similarity value $s_1 < (t - 2) = 0.4$ do not need to be inserted into the accumulator, because even if they have exact similarities in the other two attribute values (i.e. $s_2 = 1$ and $s_3 = 1$), they will not be able to reach the total minimum similarity threshold, because $(s_1 + 1 + 1) < 2.4$. Similarly, when candidate record identifiers for the second attribute are retrieved from the second index, we can check if their accumulated similarity $(s_1 + s_2) < (t - 1) = 1.4$, and if so, these candidates can also be removed from the accumulator. Finally, when adding candidate record identifiers from the third index, all candidates with a total similarity of $(s_1 + s_2 + s_3) < 2.4$ can be removed. Throughout this process, once a candidate record identifier has been removed, it is inserted into a list of ‘do not consider’ record identifiers, so that it will not be added into the accumulator later on.

Australian state/territory	Number of records	Number of unique values		
		Postcodes	Suburbs	Surnames
NT	48,754	28	171	15,887
ACT	115,558	31	132	28,599
TAS	184,158	118	868	20,430
SA	544,562	342	1,304	63,288
WA	653,167	394	1,395	77,325
QLD	1,309,744	432	2,945	110,028
VIC	1,738,216	708	3,030	175,045
NSW	2,323,355	624	4,223	207,403

Table 1: Characteristics of the *Australia on Disk* data sets (sorted according to number of records).

While this optimisation can improve query time performance by a large amount, setting the total minimum similarity t too high will mean that potentially good matches are not being retrieved, resulting in a lower matching accuracy.

Many other optimisations are possible (Bayardo et al. 2007), but are left for future work. For example, within the query phase, the computationally expensive approximate string comparisons can be cached, so future comparisons of the same pair of string values only require a cache look-up. This would reduce the query time for all three investigated indexing methods, at the expense of additional memory needed.

4 Experimental Evaluation

We used a collection of real-world Australian data sets to conduct a series of experiments with the aim of assessing the timing performance, memory usage and matching accuracy of the three indexing methods described in the previous section. All experiments were conducted on an otherwise unloaded Linux server with two Intel Xeon quad-core 64-bit CPUs with 2.33 GHz clock frequency, 8 Gigabytes of main memory, and two SAS drives (446 Gigabytes in total).

For the experiments presented in this paper, we used postcode, suburb name and surname values from all eight Australian states and territories, extracted from a 2002 edition of the *Australia On Disc*¹ data collection. This data corresponds to the entries in the Australian telephone books in late 2002, and thus has characteristics similar to many real world data collections used by Australian organisations. Table 1 shows the size of these data sets, as well as the number of unique values in each of them, and Figure 3 illustrates the sorted distribution of these values.

As can be seen, and as one would expect, the number of unique postcodes and suburb names in each data set is much smaller than the number of unique surnames. Still, the larger states with a bigger population do contain more postcode and suburb areas, and also have a larger variety of surnames. The distribution of surname values is very skewed, which means a small number of surnames are very common, while the majority of surnames are very rare. For example, the four most common surnames in New South Wales (NSW) are: ‘Smith’ (10.6% of population), ‘Jones’ (5.1%), ‘Brown’ (5.1%), ‘Williams’ (4.9%), and ‘Wilson’ (4.3%), together accounting for 30% of the population. On the other hand, there are more than 100,000 surnames that appear only once in NSW. Suburb names and postcodes are more evenly distributed, with most postcode values having a frequency count of between 50 and 10,000. Interestingly, every postcode in the state of Tasmania (TAS) has at least eight records, while in all other states and territories there are postcodes that contain only one record in the *Australia On Disc* data collection.

¹<http://www.australiaondisc.com>

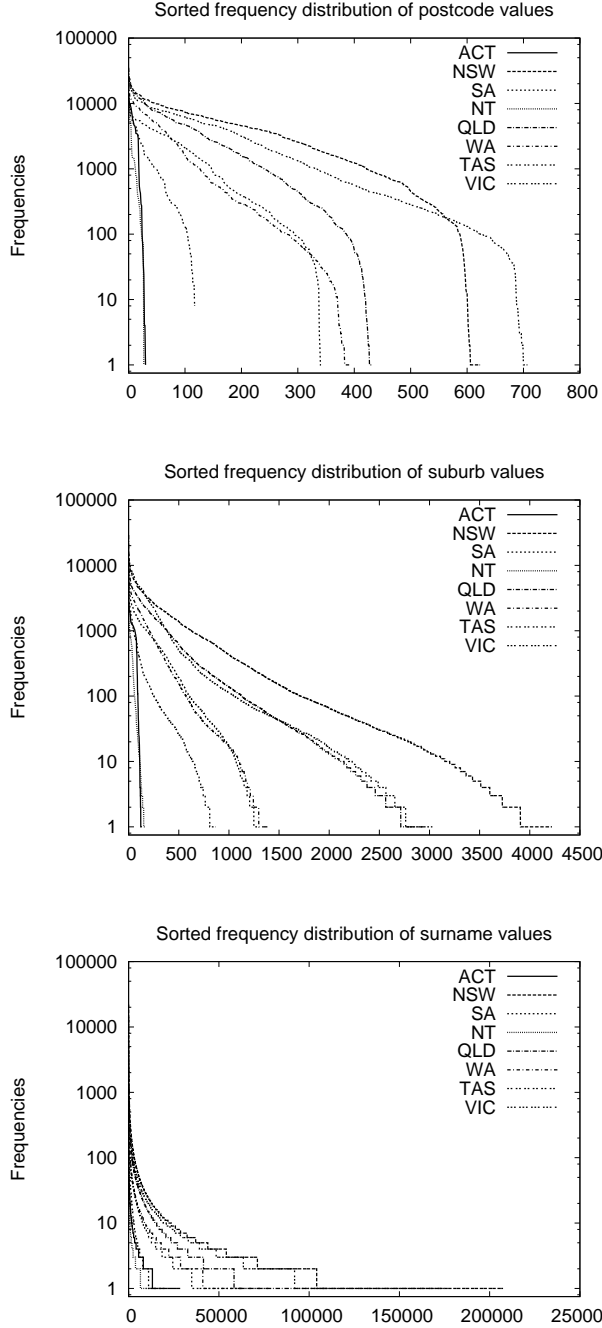


Figure 3: Sorted frequency distributions of values in *Australia on Disk* data sets.

The three indexing methods were implemented in Python, with version 2.5.1 used for the experiments. The Double-Metaphone (Christen 2006) encoding was used for the surname and suburb name blocking, while for postcodes simply the last three digits were used (i.e. all postcodes that have the same last three digits were put into the same block). For attribute value comparisons, the Winkler (Christen 2006) approximate string comparison was used for surnames and suburb names, while 1-gram based digit comparison was used for postcodes (Christen 2006).

In order to evaluate the query times required by the three indexing methods presented in Section 3, we manually generated two series of query sets. For each of the eight state or territory data sets, we randomly selected two sets of 100 records to be used as query records. In the first eight query sets (one for each state or territory), we manually applied one sin-

Australian state/territory	Standard-Blocking	Sim-Aware-Inv-Index	Mat-Sim-Aware-Inv-Index
NT	0.7	5.0	7.2
ACT	1.6	14.3	34.6
TAS	2.0	9.4	40.1
SA	6.6	78.9	1,090.0
WA	7.5	125.7	2,214.0
QLD	14.1	334.3	—
VIC	19.2	904.1	—
NSW	25.7	1,509.0	—

Table 2: Time used to build the index in memory (all values in seconds).

Australian state/territory	Standard-Blocking	Sim-Aware-Inv-Index	Mat-Sim-Aware-Inv-Index
NT	80	130	238
ACT	116	237	571
TAS	151	285	704
SA	336	824	3,243
WA	386	1,022	4,467
QLD	721	1,982	—
VIC	973	3,101	—
NSW	1,243	4,068	—

Table 3: Memory usage for different indexing methods (all values in Megabytes).

gle modification to only one of the three attributes (i.e. either the surname, suburb name, or postcode value). These modifications corresponded to possible typographical errors for surnames and suburb names (for example, ‘dickson’ was modified in one record into ‘dixon’), while for postcodes we only changed one digit (for example, ‘2607’ into ‘2601’). As a result, these 100 query records will not exactly match with their corresponding original (unchanged) records in the index, and thus an approximate match will have to be found. This allows us to measure both the time required to find and rank the approximate matches for a given query record, as well as the accuracy of the approximate matches being returned.

For the second series of query sets, we modified all three attribute values for all 100 query records, sometimes making several changes to a value (for example, ‘wollongong’ into ‘wolonnonggg’). This makes the matching process much harder, as no exact match will be found in any of the three attributes, and the modified values possibly even end up in different blocks, resulting in missed matches for the inverted index approaches, as will be discussed below. These data sets also allowed us to evaluate the times needed for matching query records that will have no exact matching values, and the impact of this on the time required for matching query records.

For each of the eight data sets described above, and for each of the three indexing methods presented in Section 3, we built the index in memory, and then queried it using the corresponding two query sets. We recorded the time used to build an index data structure in memory, the amount of memory used by an index, the times used for matching the 100 query records, and the accuracy of the resulting matches. For the experiments with optimisation enabled, we set the minimum threshold value as $t = 2.0$ (for $n = 3$ attributes and indices). The results of these experiments are shown in Tables 2 to 5, and Figure 4.

5 Results and Discussion

As the build timing results in Table 2 show, the standard blocking approach is fastest to build, mainly because it only inserts record identifiers into blocks but doesn’t calculate similarities between record val-

ues. For the similarity-aware inverted index, the build time increases more than linearly with the number of records in the database, which is due to the fact that all inverted index lists are becoming longer, and inserting new elements into them takes more time. Additionally, the blocks of record values also become larger, and thus more similarity calculations need to be done. The time used by the materialised similarity-aware inverted index increases even faster, because each record will be inserted into several inverted index lists, and the more records are already in the index, the more often a new value will need to be added into other index lists of similar values. Both inverted index based approaches are therefore currently not fully scalable to very large databases, and more work needs to be done to improve upon this. Specifically, additional optimisation techniques (Baryardo et al. 2007) will need to be investigated.

As can be seen in Tables 2 and 3, the materialised similarity-aware inverted index required more than the 8 Gigabytes of main memory available on our server for the larger data sets, and we therefore had to abandon these experiments. A disk-based inverted index approach would be required in these cases, making both the build and the query time much slower. The other memory usage results in Figure 3 show that standard blocking requires less than half of the memory of the similarity-aware inverted index. Compression techniques for inverted indices (Witten et al. 1999, Zobel & Moffat 2006) could be implemented to reduce the memory requirements for the inverted indexing methods, making them more scalable.

Looking at the matching accuracy results shown in Tables 4 and 5, the major obvious difference is that the results for the query sets with only one modification are much better than the ones with three modifications per record, as one would expect. Additionally, the matching accuracy becomes lower as the data sets become larger. This is likely because the larger data sets will have more records that contain values that are similar to the values in a query record. Thus, the likelihood that the values of a modified query record match better to the values of a different database record, rather than the values of its original record in the database, is increased.

For both inverted index approaches, the results for the query data sets with three modifications are significantly below the results for the standard blocking approach. As already discussed in Section 3.2, the problem with our inverted index approaches is that they only add similarity values into the accumulator of records that are in the same block as the query record. A query record that has a modification that puts a value into a different block will therefore lose the corresponding similarity values. Improving upon this deficiency will be one of our major future research tasks. Also of interest is that the optimisation step with both inverted index methods can improve the matching accuracy, in certain cases significantly, by removing some possible matches from the accumulator. Whether this is specific to our data and experimental setup, or a more general property of this optimisation technique, needs to be investigated with more experiments on different data sets and using different values of the minimum similarity threshold.

Looking at the query timing results presented in Figure 4, one can clearly see that both inverted index based methods outperform standard blocking significantly. The best improvement we measured was the similarity-aware inverted index being 100 times faster than standard blocking (on the ‘NT’ data set, with optimisation enabled, and the query set with one modification per record). It is clear that the current implementation of the materialised similarity-aware inverted index is quite slower than its non-

Australian state/territory	Standard-Blocking	Sim-Aware-Inv-Index	Mat-Sim-Aware-Inv-Index
NT	97 / 97	99 / 99	97 / 99
ACT	92 / 92	95 / 95	95 / 95
TAS	94 / 94	93 / 93	93 / 93
SA	95 / 95	97 / 97	97 / 97
WA	96 / 96	95 / 95	95 / 95
QLD	98 / 98	94 / 94	–
VIC	95 / 95	92 / 92	–
NSW	91 / 91	87 / 87	–

Table 4: Matching accuracy as percentage values of correctly top-ranked true matches for the query data sets with one modification only per record. Each pair of accuracy results corresponds to the tests without / with optimisation.

Australian state/territory	Standard-Blocking	Sim-Aware-Inv-Index	Mat-Sim-Aware-Inv-Index
NT	85 / 85	67 / 66	67 / 66
ACT	78 / 78	60 / 65	60 / 65
TAS	75 / 75	55 / 54	55 / 54
SA	78 / 78	39 / 52	39 / 52
WA	73 / 73	48 / 54	48 / 54
QLD	69 / 69	30 / 41	–
VIC	72 / 72	36 / 56	–
NSW	79 / 79	45 / 65	–

Table 5: Matching accuracy for the query data sets with three modifications per record. The same format as in Table 4 is used.

materialised version. It is also clearly visible in Figure 4 that the timing advantage of the inverted index based methods gets smaller as the index data structures are getting bigger. The smallest improvement we measured was that the similarity-aware inverted index was 30% faster than standard blocking (on the ‘NSW’ data set, without optimisation and one modification per query record).

6 Conclusions and Future Work

In this paper, we have investigated three indexing methods aimed at large-scale real-time entity resolution. We have compared the standard blocking approach with two variations of a similarity-aware inverted index approach. These two approaches significantly outperformed standard blocking with regard to the average time required to match a query record, being between 1.3 and 27 times faster in our experiments without optimisations, and between 2.6 to 100 times faster with a simple threshold-based optimisation technique enabled.

While the accuracy of the matching results of the inverted index approaches is comparable to the standard blocking approach when the query records are similar to the records stored in the index, their accuracy drops significantly when the query records mainly contain values that are not stored in the inverted index. This drawback is one of the main avenues for future work. Additionally, we aim to implement other optimisation techniques for the inverted index approaches, based on the various techniques used for large-scale Web search engines (Baryardo et al. 2007, Zobel & Moffat 2006), and conduct further experiments on other large data sets.

To the best of our knowledge, nobody has applied techniques developed for large-scale Web search engines to the problem of real-time entity resolution. Our work is a first step in this direction, and we plan to continue this work with the aim to combine information retrieval and machine learning approaches to develop a new generation of scalable, accurate, automatic and real-time entity resolution techniques.

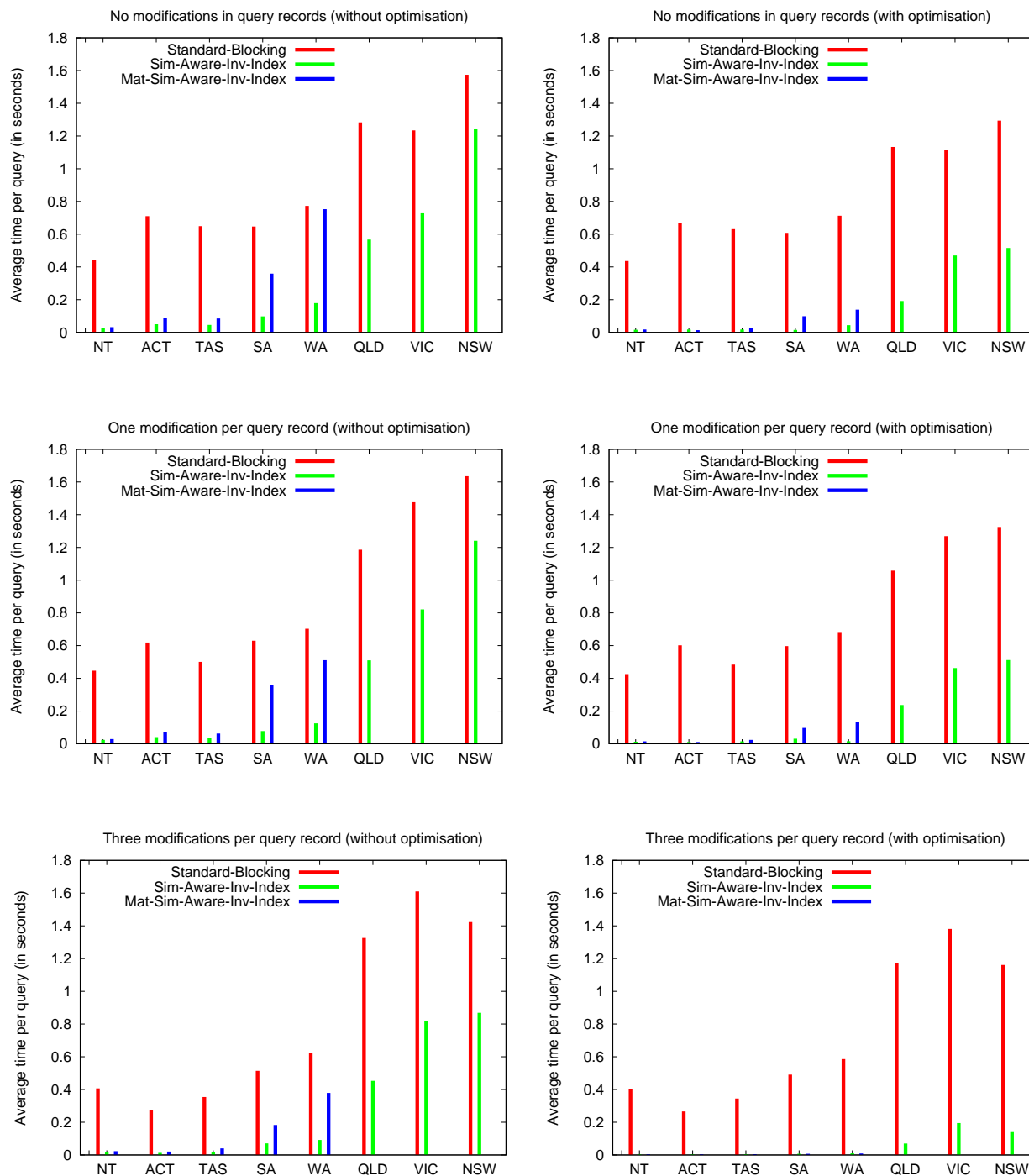


Figure 4: Query timing results for the three indexing methods. The graphs on the left side show the results without optimisation, while the right side graphs show the results with optimisation enabled.

References

- Aggarwal, C.C. & Yu, P.S. (2000), The IGrid index: Reversing the dimensionality curse for similarity indexing in high dimensional space, *in* 'ACM International Conference on Knowledge Discovery and Data Mining' (SIGKDD'00), Boston, pp. 119–129.
- Aizawa, A. & Oyama, K. (2005), A fast linkage detection scheme for multi-source information integration, *in* 'Web Information Retrieval and Integration' (WIRI'05), Tokyo, pp. 30–39.
- Baxter, R., Christen, P. & Churches, T. (2003), A comparison of fast blocking methods for record linkage, *in* 'ACM SIGKDD'03 Workshop on Data Cleaning, Record Linkage and Object Consolidation', Washington DC, pp. 25–27.
- Bayardo, R.J., Ma, Y. & Srikant, R. (2007), Scaling up all pairs similarity search, *in* 'International Conference on World Wide Web' (WWW'07), Banff, Canada, pp. 131–140.
- Bhattacharya, I. & Getoor, L. (2007), 'Query-time entity resolution', *Journal of Artificial Intelligence Research*, **30**, 621–657.
- Bilenko, M., Kamath, B. & Mooney, R.J. (2006), Adaptive blocking: Learning to scale up record linkage, *in* 'IEEE International Conference on Data Mining' (ICDM'06), Hong Kong, pp. 87–96.

- Christen, P. (2006), A comparison of personal name matching: Techniques and practical issues, in 'Workshop on Mining Complex Data' (MCD'06), held at IEEE ICDM'06, Hong Kong.
- Christen, P. (2008), Febrl – An open source data cleaning, deduplication and record linkage system with a graphical user interface, in 'ACM International Conference on Knowledge Discovery and Data Mining' (SIGKDD'08), Las Vegas, pp. 1065–1068.
- Christen, P. & Goiser, K. (2007), Quality and complexity measures for data linkage and deduplication, in F. Guillet & H. Hamilton, eds, 'Quality Measures in Data Mining', Springer Studies in Computational Intelligence, Vol. 43, pp. 127–151.
- Cohen W.W., Ravikumar P. & Fienberg S.E. (2003), A comparison of string distance metrics for name-matching tasks, in 'IJCAI'03 Workshop on Information Integration on the Web' (IIWeb'03), Aca-pulco, pp. 73–78.
- Cohen, W.W. & Richman, J. (2002), Learning to match and cluster large high-dimensional data sets for data integration, in 'ACM International Conference on Knowledge Discovery and Data Mining' (SIGKDD'02), Edmonton, pp. 475–480.
- Conrad, J.G., Guo, X.S. & Schriber, C.P. (2003), On-line duplicate detection: Signature reliability in a dynamic retrieval environment, in 'ACM Conference on Information and Knowledge Management' (CIKM'03), New Orleans, pp. 443–452.
- Dong, X., Halevy, A., & Madhavan, J. (2005), Reference reconciliation in complex information spaces, in 'ACM International Conference on Management of Data' (SIGMOD'05), Baltimore, pp. 85–96.
- Elfeky, M.G., Verykios, V.S. & Elmagarmid, A.K. (2002), TAILOR: A record linkage toolbox, in 'International Conference on Data Engineering' (ICDE'02), San Jose, pp. 17–28.
- Elmagarmid, A.K., Ipeirotis, P.G. & Verykios, V.S. (2007), 'Duplicate record detection: A survey', *IEEE Transactions on Knowledge and Data Engineering* (TKDE), **19**(1), 1–16.
- Fellegi, I.P. & Sunter, A.B. (1969), 'A theory for record linkage', *Journal of the American Statistical Society*, **64**(328), 1183–1210.
- Hernandez, M.A. & Stolfo, S.J. (1995), The merge/purge problem for large databases, in 'ACM International Conference on Management of Data' (SIGMOD'95), San Jose, pp. 127–138.
- Gu, L. & Baxter, R. (2006), Decision models for record linkage, in 'Selected Papers from AusDM', Springer LNCS 3755, pp. 146–160.
- Jin, L., Li, C. & Mehrotra, S. (2003), Efficient record linkage in large data sets, in 'International Conference on Database Systems for Advanced Applications' (DASFAA'03), Tokyo, pp. 137–146.
- Kalashnikov, D.V. & Mehrotra, S. (2006), 'Domain-independent data cleaning via analysis of entity-relationship graph', *ACM Transactions on Database Systems* (TODS), **31**(2), 716–767.
- Michelson, M. & Knoblock, C.A. (2006), Learning blocking schemes for record linkage, in 'National Conference on Artificial Intelligence' (AAAI'06), Boston.
- Weis, M. & Naumann, F. (2007), 'Space and time scalability of duplicate detection in graph data', Technical report, University of Potsdam, Germany.
- Winkler, W.E. (2006), 'Overview of record linkage and current research directions', Technical Report RR2006/02, US Bureau of the Census.
- Witten, I.H., Moffat, A & Bell, T.C (1999), *Managing Gigabytes: Compressing and Indexing Documents and Images*, 2nd Ed. Morgan Kaufmann.
- Yan, S., Lee, D., Kan, M.Y., & Giles, L.C. (2007), Adaptive sorted neighborhood methods for efficient record linkage, in 'ACM/IEEE-CS Joint Conference on Digital Libraries' (JCDL'07), Vancouver, pp. 185–194.
- Yin, X., Han, J. & Yu, P.S. (2006), LinkClus: Efficient clustering via heterogeneous semantic links, in 'International Conference on Very Large Data Bases' (VLDB'06), Seoul, pp. 427–438.
- Zobel, J. & Moffat, A. (2006), 'Inverted files for text search engines', *ACM Computing Surveys* (CSUR), **38**(2).

Customer Event Rate Estimation Using Particle Filters

Harsha Honnappa¹

¹ Applied Research Group,
Satyam Computer Services Ltd.,
Indian Institute of Science (IISc),
Bangalore, India.
Email: Harsha.Honnappa@satyam.com

Abstract

Estimating the rate at which events happen has been studied under various guises and in different settings. We are interested in the specific case of consumer-initiated events or transactions (credit/debit card transactions, mobile phone calls, online purchases, etc.), and the modeling of such behavior, in order to estimate the rate at which such transactions are made. In this paper, we detail a model of such events and a Bayesian approach, utilizing Sequential Monte Carlo technology, to online estimation of the event rate from event observations alone.

Keywords: Event Rate Estimation, Markov Jump Process, Particle Filter, Cox Process, Poisson Process.

1 Introduction

Modeling consumer behavior is advantageous for a multitude of business problems - targetted advertising/marketing, fraud detection, click-stream analysis, etc. There are a plethora of statistics that one can 'mine' or extract from this data¹. One statistic that is particularly significant, especially since (individual) consumer events are time-sequenced, is the rate at which such events occur. The rate is a particularly useful metric in detecting changing consumer behavior.

Indeed, there have been several approaches to estimating the (mean) rate at which events occur. In (Lambert et al., 2001), a very detailed algorithm for estimating the rate using a 'controlled' version of the exponentially weighted moving average (EWMA) time-series model is described. Scott et al, study click-rates with a Markov Modulated Poisson Process (MMPP), in (Scott and Smyth, 2003). In (Weinberg et al., 2006) a Markov Chain Monte Carlo (MCMC) approach to determining the rate of a doubly stochastic Poisson process is detailed, based on call center data. Similar approaches have been used in other proprietary systems.

In this paper, our aim is to detail a different approach to rate estimation. We assume that the rate itself is unobservable, since it is not exactly a physically manifested signal. Thus, we can view the

rate estimation problem as a latent state estimation or, in signal processing parlance, a filtering problem. As we shall see, in this case we are dealing with a non-linear filtering problem. Further, we want to estimate the rate, in an online fashion, without storing much data.

Latent state space models are widely used in many applications. The problem that this paper aims to solve is estimating this state using only observations and a knowledge of a *state space model*. This filtering problem has been solved in a few cases (and indeed optimally), and the most famous example of such a filter is the Kalman filter, which optimally estimates the state in the case of a linear, Gaussian model; i.e., when the state process is a Gaussian process and the relation between observations and states is a linear model, driven by Gaussian noise. As we shall see, in our model the underlying state process cannot be a Gaussian process. Instead, we assume that the rate process follows a piecewise deterministic Markov process (PDMP), see (Davis, 1984). More specifically, we assume that the rates follow a piecewise continuous Markov process, or a Markov jump process (MJP). Further, we assume a Poisson observation model, in which the number of events occurring within a given time interval is Poisson distributed, dependent upon the underlying rate process.

Given our modeling assumptions, the solution to the filtering problem requires sophisticated algorithms. Here, we consider a Bayesian approach to solving the problem - using Sequential Monte Carlo (SMC) methods. SMC methods, more commonly known as particle filtering (PF), estimate the posterior density of the (latent) states given the observations, by a cloud of weighted samples or *particles*. These particles are updated and propagated using sequential importance sampling (SIS) and MCMC methods. For a very good introduction to PF's see (Arulampalam et al., 2002) and (Cappé et al., 2007). Generic particle filtering approaches usually used assume that, for every observation, there is a corresponding underlying state; i.e., the state process evolves stochastically at every time instant. However, in our model, we assume that the underlying state changes at random instants of time and that it remains constant between these instants of time. Thus, the rate at which state changes occur is different from the rate at which events are observed. Put another way, we do not know the number of jumps in the state process. This requires a different paradigm of PF's.

Variable rate particle filters (VRPF), (Godsill and Vermaak, 2005) and (Godsill et al., 2007), are a specific type of particle filter that allows for different

Copyright ©2008, Australian Computer Society, Inc. This paper appeared at the Seventh Australasian Data Mining Conference (AusDM 2008), Glenelg, Australia. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 87, John F. Roddick, Jiuyong Li, Peter Christen and Paul Kennedy, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

¹We shall henceforth refer to the transactions or consumer-initiated events, noted above, as *events*.

rates of evolution for the observations and the state. VRPF's define a 'neighborhood' for each sample observed, and sample *stopping times*, or jump-times, and states from a predefined importance density until the neighborhood for a given sample is *complete*, in some well defined manner. This framework is exactly what is required for estimating event rates, given our modeling assumptions. Here we detail how the VRPF algorithm can be applied to this problem, with appropriate definition of neighborhoods and importance density. We make some enhancements to the basic algorithm laid out in (Godsill et al., 2007), by incorporating *moves* into the filter to improve particle diversity.

The rest of this paper is organized as follows. In section 2, we describe the MJP model we assume for the rate process and the corresponding state evolution density functions. We also describe the Poisson observation model, and give a brief mathematical treatment of the specific type of Poisson process we assume for this model. Next, in section 3, we describe the VRPF algorithm that we adopt in this paper. We also briefly discuss particle filters and the general idea behind them, for completeness. In section 4, we present results of experiments we conducted comparing the VRPF and EWMA algorithms and comparisons of the individual VRPF algorithms, incorporating moves and without moves. Finally, we conclude in section 5 with a listing of future research directions we intend to pursue and a brief discussion of the implications and importance of this model on estimating customer event rates.

2 The Model

There are numerous algorithms that aim to solve the latent state estimation problem. However, we need to first model the latent state process. There are a couple of dimensions to this general modeling problem and listing them will help in setting up our own problem. First, the state space itself can be discrete or continuous. Secondly, the event sequencing (or time-ordering) can be, again, discrete or continuous. As we shall see, it is useful to model the state process by a continuous-time, continuous state-space model, with a point process observation model, in our problem.

An example would help set the stage for a more formal explanation. Consider credit card usage. Generally, a histogram of the time of usage of the cards, across a portfolio of card users and over a 24 hour period, would be a bell curve. Clearly, there are periods of the day during which consumers tend to transact more, and other times during which they transact less. For an individual user, the time scale is probably much different, but even there, individual consumers would tend to use their cards more over the weekend (shopping, movies etc.) than during the working week. Thus, there are clear categories of usage. However, even though there maybe a finite number of usage categories, the quantification of this usage (in terms of a rate) does not necessarily have to be in a finite or discrete set of values. To illustrate, suppose that a cardholder uses her card more often over a weekend, implying a higher rate. Even though she exhibits a fairly stationary behavior, it is not true that she has the exact same rate *every* weekend. The rate quanta can vary over a continuous range of values. Further, one cannot assume that the cardholders state changes abruptly at a *set* time instance (even though it is assumed that the state can change at an instant). Thus, even though the

number of state changes over a given time period maybe denumerable, the instants of the changes and number of state changes, and the magnitude changes of the state process are random. Figure 1 is an illustration of this conceptual model.

We can see that there is a compelling case for modeling such consumer behavior with continuous time stochastic processes. Let us now look at a more formal presentation, and start with a description of our model of consumer behavior.

2.1 An Observation Model

As mentioned in the introduction to this section, generally there are a denumerable and random number of events during any given period of time. We shall model the event observations as a type of Markov process - a Poisson process, with rate $\lambda > 0$; see (Ross, 2007, Ch. 5) for a good introduction. Note that Poisson processes are a type of continuous-time Markov chain, where in the states of the chain are the number of events seen up to a given instant of time. These processes are also called as counting processes, and also as pure-birth processes. Poisson processes have some very useful properties -

- The number of events over a given time interval is Poisson distributed - $P(N(t+s) - N(s) = n) = e^{-\lambda t} \frac{(\lambda t)^n}{n!}$. Where, $n \in \{0, 1, 2, \dots\}$.
- The Poisson process has stationary and independent increments; i.e., $N(t+s) - N(s)$ is independent of $N(s+u) - N(u)$ where, $u < s < t$.
- The time between events is exponentially distributed.

Note that in the basic, or homogeneous, Poisson process the rate, λ , is assumed to be a constant. However, in our problem, the rate is assumed to change at random instants of time to random locations. Thus, we need to turn to a more sophisticated model, a doubly stochastic Poisson process or Cox process, (Daley and Vere-Jones, 2003), in which the rate is assumed to be some stochastic process. There are several interesting applications of Cox processes, including modeling insurance risk and securities risk, see (Dassios and Jang, 2003) and (Lando, 1997) for illustrative examples.

2.1.1 Cox Process

We shall follow Bremaud's definition of a Cox process, (Bremaud, 1981) (also used in (Dassios and Jang, 2003)). Let, (Ω, \mathcal{F}, P) be a probability space, where \mathcal{F} consists of all σ -algebras \mathcal{F}_t . Suppose N_t is a Poisson process adapted to \mathcal{F}_t . Let λ_t be a non-negative process that is \mathcal{F}_t adapted, and

$$\int_0^t \lambda_s ds < \infty \text{ a.s.}$$

If $\forall 0 \leq t_1 \leq t_2$ and $u \in \mathcal{R}$,

$$\begin{aligned} E \left\{ e^{iu(N_{t_2} - N_{t_1})} | \mathcal{F}_{t_2} \right\} \\ = \exp \left\{ (e^{iu} - 1) \int_{t_1}^{t_2} \lambda_s ds \right\} \end{aligned}$$

Then, the probability of n events in time interval $t_2 - t_1$ is Poisson distributed, given the rate process over

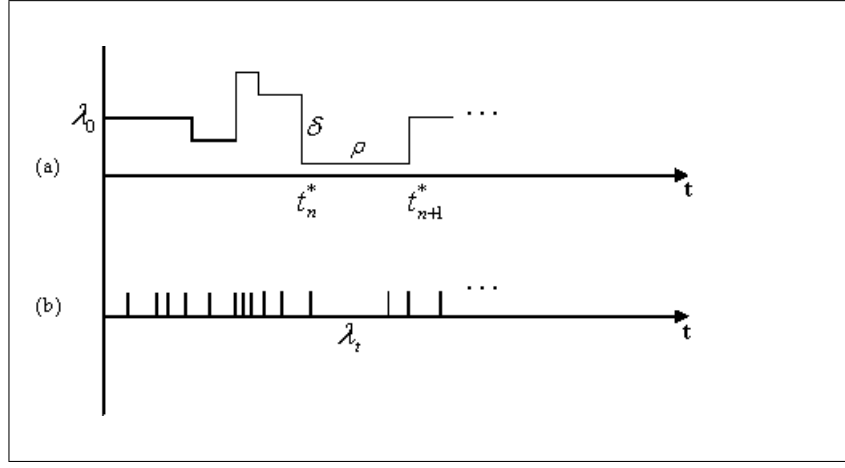


Figure 1: The conceptual model of a Poisson observation model and a Markov Jump Process rate process. (a) depicts the state process, where λ_0 is the initial value, ρ and δ are model parameters governing the jump-time Poisson process, and the state-magnitude Markov chain resp. and t_n^* and t_{n+1}^* are two jump times, (b) depicts the events sequence, where events are represented by marks on the time axis. The rate at which events occur is shown as λ_t which is the realization of the rate process at time t .

$$t_1 \leq s \leq t_2,$$

$$P\{N_{t_2} - N_{t_1} = n | \lambda_s, t_1 \leq s \leq t_2\} = \frac{e^{-\int_{t_1}^{t_2} \lambda_s ds} \left(\int_{t_1}^{t_2} \lambda_s ds \right)^n}{n!}$$

Thus, in order to employ the Cox process as a valid observation model, a model for the state evolution process, λ_t , needs to be chosen. In the next section we will consider an appropriate model for our problem.

2.2 The State Evolution Model

Markovian state evolution models are uniquely represented by a state transition probability - $P(Y|X)$. Where, crudely, P represents the probability of moving to state Y , given that the system is in state X at the previous step. This state transition probability is also represented by the use of a *transition function*, $k_{s,t}(dy, x)$, where $s(< t)$ are the times at which one considers the process evolution.

There are many choices of processes for describing the state evolution. A good example would be a *random walk*, where in the current state is a random perturbation of the state at the previous time instant-

$$\lambda_{t+1} = \lambda_t + \nu_{t+1} \quad (1)$$

Where, ν_{t+1} is the random perturbation or noise component, and λ is the state. It is quite simple to verify the Markovian nature of such a process. Further, if the noise is assumed to be standard normal, $\mathcal{N}(0, 1)$, and $\lambda_0 = 0$, then this process represents Brownian motion (of course subject to some conditions that allow its description in continuous time).

In (Lambert et al., 2001) the authors describe a multiplicative noise state evolution model -

$$\lambda_{t+1} = \nu_{t+1} \lambda_t \quad (2)$$

Where,

$$\nu \sim \Gamma(\alpha, \alpha)$$

And λ_0 is assumed to be known.

In (Weinberg et al., 2006), the authors describe the behavior of calls arriving at a call center at a large North American bank. From their analysis, it is clear that the rate at which calls arrive has different stages, and that these vary from day-to-day. The authors model this behavior with a doubly stochastic Poisson process, with the rate evolution governed by -

$$\lambda_{t+1} = w_{d_{t+1}} v + \epsilon_{t+1} \quad (3)$$

Where, $w_{d_{t+1}}$ is the proportion of calls occurring in the period of interest, v is an estimate of the number of daily volume on the day of interest and ϵ_{t+1} is a random perturbation. The most interesting aspect of this model is the fact that the model accommodates both inter- and intra-day variation in the rate, thus allowing for greater control of the model.

However, as described in the introduction to this section, it appears more plausible that the states of consumers undergo a slow evolution. Further, we assume that the state can change only a finite number of times in a given time period (say, a day). We model the state evolution with a Markov process, and thus introduce the assumption that the future state is conditionally independent of the past, given the present.

Given the hypotheses stated above it does not appear reasonable to assume that the state evolution follows some type of a *diffusion process*. A pure-diffusion process assumes that the state is continuously evolving at each time instant. This violates the assumption of a denumerable number of state changes in a given period of time. Jump-diffusion processes appear to be a more plausible model, since one can assume that the state changes are represented by the jumps in the process, and that the process then evolves around the magnitude jumped to. However, this precludes any notion of a slow evolution, since the state still changes at each time instant.

A more plausible model for the state evolution process is to assume it follows a piecewise deterministic Markov process (PDMP). Specifically, we assume that

the process is a Markov jump process (MJP). Note, however, we do not pursue the PDMP description directly, and instead merely point out the connection between MJP's and PDMP's. Describing PDMP's is outside the purview of this paper, and the interested reader is directed to (Davis, 1984). Figure 1 shows an example of how the state process may vary. We assume that the state process is a pure-jump process, with constant trajectory, or flow, between the jumps. Our assumption that the state of a user does not change significantly, once in a given state, means that MJP's are a good model for representing such behavior. A brief review of MJP's and their properties follows.

2.2.1 Markov Jump Process (MJP)

A jump process is a stochastic process that changes its magnitude only at random instants of time. Generally, any process with piecewise-constant trajectories between jumps is called as a jump process. Markov Jump Processes impose a Markovian structure on the states, by requiring that the future states are conditionally independent of the past, given the present. Linked to this is an important result that shows that a jump process is Markovian *iff* the time between jumps is exponentially distributed - $F_\lambda(t)$, $\forall \lambda \in S$, with parameter $\rho(\lambda(t))$.

The basic characteristics of MJP's include -

- A stochastic kernel $k(\lambda, A)$ on $S \times \mathcal{B}(S)$ ($A \in \mathcal{B}(S)$) that satisfies the condition, $k(\lambda, \lambda) = 0$. This ensures that there actually *is* a jump at each jump time.
- A bounded, non-negative function of the state, $\rho(\cdot)$, that controls the rate at which jumps occur. We assume that the rate at which jumps occur is much lower than the rate process magnitude, $\lambda(t)$.

Intuitively, a continuous-state MJP can be thought of as starting in some position λ_0 at time t_0 , and remains in that state till an exponentially distributed time t_1 , when it jumps to a new location λ_1 according to the transition kernel, $k(d\lambda_1, \lambda_0)$. It then remains in the state λ_1 until the next exponentially distributed jump time, t_2 , and again the process jumps to a new location, following the transition kernel $k(d\lambda_2, \lambda_1)$. This continues till the end of the period of interest, or ad infinitum.

We can think of the MJP as being the composition of a Poisson (point) process, N , and a Markov chain, M , $\Lambda = N \circ M$. Now, we can model the state by a tuple, $\lambda = (\tau, \theta)$, where, τ is the jump time and θ is the state-magnitude. Further, we assume that the state evolution density can be decomposed in the following manner -

$$f(\tau_t, \theta_t | \tau_{t-1}, \theta_{t-1}) = f(\tau_t | \tau_{t-1}, \theta_{t-1}) f(\theta_t | \theta_{t-1}) \quad (4)$$

We assume that the future jump time is conditionally independent of the future state-magnitude, given the present jump time and the present state-magnitude, and the future state-magnitude is conditionally independent of the present jump time. This assumption fits in nicely with our assumption that the state process is a MJP. This is a very general description of the process evolution. In order to adapt this description to the problem of event rate estimation, we make a few further assumptions that will fully describe the model -

- We assume that users generally have a fairly moderate rate of usage, with jumps being not too high or low.
- The high rate states tend to be of short duration (or 'bursty').
- High rate states are usually followed by a drop down to a more moderate rate.
- A low rate state is followed by a moderate state with high probability.

Based on these, still, rather general assumptions, we model user behavior by considering the following forms for the state evolution densities.

2.2.2 Jump Time Distribution

As stated earlier, one of the properties of MJP's is that the time between jumps is exponentially distributed. The parametrization of this distribution is assumed to dependent upon the present state-magnitude in the following manner.

$$\frac{1}{\rho(\theta_t)} \exp\left\{-\frac{\tau_{t+1} - \tau_t}{\rho(\theta_t)}\right\} \quad (5)$$

Where, the mean rate of the jumps, as a function of the state-magnitude is given as,

$$\rho(\theta_t) = \frac{\alpha}{\theta_t}$$

Here, $\alpha \in \mathbb{R}$. As stated earlier, we assume that the rate at which jumps occur is much lower than the event rate. α allows us to control the jump rate. Thus, one can see that a large rate value will produce a very low future jump time, and vice versa.

2.2.3 State-magnitude Distribution

In the description of a MJP, it was noted that the process is associated with a transition kernel which determines the evolution of the state-magnitude, θ . We use a random walk type model, (1), to describe the relation between the future and present state-magnitude. We define the random perturbation as -

$$\Delta\theta_{t+1} \triangleq \theta_{t+1} - \theta_t \quad (6)$$

Note that $\theta \in \mathbb{R}^+$. This implies that the difference between the present and future magnitudes is bounded below -

$$-\theta_t \leq \Delta\theta_{t+1} < \infty$$

We could model $\Delta\theta_{t+1}$ by a shifted Gamma distribution (the density function of which exists), where the shift parameter is $-\theta_t$. However, as defined, MJP's require the probability of jumping to the same location should be zero, $k(\lambda, \{\lambda\}) = 0$. That is, $\Delta\theta_{t+1} \neq 0$. We accommodate this by modeling the density function as a mixture of two shifted gamma distributions, such that the density of the mixture at $\Delta\theta_{t+1} = 0$ approaches zero, for a fixed θ_t . By a slight abuse of mathematical propriety, we will define the first mixture component density, c_1 , as that having most of its density in the region $[-\theta_t, 0)$, and the second component, c_2 , as that having support, $[0, \infty)$. While there is a small probability that $\Delta\theta_{t+1} = 0$, by carefully choosing the parameters of the gamma distribution and the

mixture probabilities, it is possible to minimize this.

Now, recalling the assumptions we made regarding the user behavior, we assume that high or low rate magnitudes will be followed by a return to a more 'moderate' rate. We model this behavior by considering an appropriate probability mass function for the components, which is dependent upon the present state-magnitude. In order to achieve this, we require an appropriate function, π , such that, $\pi : \mathbb{R}^+ \rightarrow [0, 1]$. We choose an exponential function, with a suitable parametrization incorporating θ_t , as this function -

$$\pi(1|\theta_t) = 1 - \exp\left\{-\frac{\theta_t}{\eta}\right\} \quad (7)$$

$$\pi(2|\theta_t) = 1 - \pi(1|\theta_t) \quad (8)$$

Where, η parametrizes the mean magnitude to which the process returns to. For example, consider a situation with a large η ($\eta \gg 1$). Then, if θ_t is small, there is a greater probability of jumping to a larger value, in the next jump (since $\pi(2|\theta_t) > \pi(1|\theta_t)$), and vice versa. We call this model as an *exponentially modulated Gamma mixture*. Finally, the state-magnitude transition density function is given by -

$$f(\theta_{t+1}|\theta_t) = \pi(1|\theta_t) \Gamma(\delta, \frac{1}{\delta\theta_t}) + \pi(2|\theta_t) \Gamma(\delta, \frac{1}{\delta\theta_t}, \theta_t) \quad (9)$$

Where, $\Gamma(\cdot, \cdot)$ is the gamma distribution and $\Gamma(\cdot, \cdot, \cdot)$ is the shifted gamma distribution, with the third parameter the shift. $\delta \in \mathbb{R}^+$ is the gamma distribution parametrization.

A final observation to be made is that the state-magnitude is assumed to have a Gamma prior distribution, with parametrization δ . Thus, we have a description of the the latent state process transition density, (4). We will use this description in designing an estimation procedure for the state from observations. As stated earlier, the state evolution process is not directly observable, but only through the Cox process observation model. Further, we would like to estimate this process (or at least the mean) in an online fashion. We accomplish this estimation with a Particle Filter. We describe the filter design and algorithm in the next section.

3 A Particle Filter Solution

Non-linear state space models are described by a state evolution model,

$$x_t = a(x_{t-1}, \nu_t) \Leftrightarrow f(x_t|x_{t-1})$$

And an observation model,

$$y_t = b(x_t, \iota_t) \Leftrightarrow f(y_t|x_t)$$

Where, $f(x_t|x_{t-1})$ and $f(y_t|x_t)$ are the transition density (a.k.a. the transition kernel) and the observation density, resp. $a(\cdot)$ and $b(\cdot)$ are some non-linear functions, and ν and ι are the driving noise processes. We have detailed our modeling assumptions in the previous section. Now, we are interested in estimating the *posterior density*², $f(x_{0:T}|y_{0:T})$. Using Bayes rule we get -

²We shall implicitly assume that the density function exists for all distributions of interest to us.

$$\begin{aligned} f(x_{0:T}|y_{0:T}) &= \frac{f(y_{0:T}|x_{0:T})f(x_{0:T})}{\int f(y_{0:T}|x_{0:T})f(x_{0:T})d\mathbf{x}} \\ &= \frac{f(x_0)f(y_0|x_0)\prod_{i=1}^T f(x_i|x_{i-1})f(y_i|x_i)}{\int f(y_{0:T}|x_{0:T})f(x_{0:T})d\mathbf{x}} \end{aligned} \quad (10)$$

However, evaluating this expression requires evaluating the integral, which is intractable in most cases. One can use a simulation based approach to estimating this density, using importance sampling (IS). IS uses a carefully designed *auxiliary* density, $q(\cdot)$, called as the importance function, whose support covers the support of the target density function. Then, the expectation, $E(g(X))$, of some function $g(\cdot)$ of random variable X , can be estimated by using N IID, weighted, samples drawn from the importance distribution -

$$\begin{aligned} E[g(X)] &= \int_{\mathcal{D}(x)} g(x)f_X(x)dx \\ &= \int_{\mathcal{D}(x)} g(x)\frac{f_X(x)}{q(x)}q(x)dx \\ &\approx \sum_{i=0}^{N-1} w_i g(x_i) \end{aligned} \quad (11)$$

Where, w_i are the normalized importance weights, N is the number of samples and x_i are samples drawn from the importance density. The un-normalized importance weights, \tilde{w}_i , are defined as -

$$\tilde{w}_i = \frac{f_X(x)}{q(x)} \quad (12)$$

These weights are then normalized before being used in (11) -

$$w_i = \frac{\tilde{w}_i}{\sum_{i=0}^{N-1} \tilde{w}_i}$$

It can be shown that the estimate (11) converges to the true expectation, $E[g(X)]$, in the limit, $N \rightarrow \infty$. The standard IS procedure can be extended to a sequential situation, where the weights are estimated and updated at each instant an observation is made. This is called as the sequential importance sampling (SIS) algorithm, and forms a subset of sequential Monte Carlo (SMC) methods. SIS forms the basis of particle filter (PF) algorithms.

3.1 Generic Particle Filters

PF's are a specific instance of SIS algorithms used to estimate the the posterior density (and associated expectation functions), using a set of samples, or particles, drawn from the importance density. Essentially one can think of drawing particles as drawing a sample path from the importance density. The importance weights are given by -

$$\tilde{w}_t^{(i)} = \frac{f(x_{0:t_n}^{(i)}|y_{0:t_n})}{q(x_{0:t_n}^{(i)}|y_{0:t_n})} \quad i = 1 \dots N \quad (13)$$

Where, $x^{(i)}$ is the i^{th} particle drawn from $q(\cdot)$. By assuming that the importance density can be decomposed as -

$$q(x_{0:t_n}|y_{0:t_n}) = q(x_{0:t-1}|y_{0:t-1})q(x_t|x_{t-1}, y_t) \quad (14)$$

i.e., the particle value at t is dependent only on the observation at t and the particle value at the previous time instant. Crudely, one can say that the sample path is extended by sampling from the second part of the importance function decomposition. From (10), and the Markovian nature of the state process, we can see that the posterior density function can be expressed as -

$$f(x_{0:t}|y_{0:t}) \propto f(y_t|x_t) f(x_t|x_{t-1}) f(x_{0:t-1}|y_{0:t-1}) \quad (15)$$

Now, substituting (14) and (15) into (13), and dropping the terms that do not depend upon the state sequence, we get -

$$w_t^{(i)} \propto w_{t-1}^{(i)} \frac{f(y_t|x_t^{(i)})f(x_t^{(i)}|x_{t-1}^{(i)})}{q(x_t^{(i)}|y_t, x_{t-1}^{(i)})} \quad (16)$$

Thus, we get a sequential update equation for the importance weight, which depends only upon the present particle, the current observation and the previous particle. At each observation instant, we draw a sample from the second part of the RHS of (14), and update the weight according to (16). However, this basic algorithm has problems with its performance. Specifically, it suffers from weight degeneracy, where most of the weight is concentrated on a few particles after a few iterations of the algorithm. This problem can be worked around by re-sampling the particles either on every iteration, or according to some criterion (described later on).

Note that in the generic particle filter, we know that the dimension of the sample space of the posterior probability distribution is strictly increasing; i.e., $\dim(S_{t-1}) < \dim(S_t)$. That is, we know that at each observation instance, there is exactly one latent state variable value. However, this is not the situation with our problem. Recall that we assume a Markov jump process (MJP) model for the state evolution process, that changes state at a rate much lower than the state-magnitude at a given instant of time. Thus, we have a situation where the number of state variable samples at each observation instance is unknown. Here, we must assume that at each instance, we have the same sample space, S , but the support of the posterior density function, E_t , is increasing. Thus, we require different algorithms to solve this problem.

There have a few attempts at solving this type of filtering problem. The Variable Rate Particle Filter (VRPF), described in (Godsill and Vermaak, 2005) and expanded in (Godsill et al., 2007), works by sampling stopping times, such that every observation has a *complete neighborhood*, where neighborhood is some well defined region around each observation. In (Del Moral et al., 2006), a general technique for sampling from a sequence of distributions that are defined on the same underlying sample space, called Sequential Monte Carlo (SMC) Samplers is defined. Applications of this algorithm to PDMP's and jumping processes is described in (Whiteley et al., 2007). We

adopt the VRPF algorithm in this paper, and describe some simple extensions to the basic algorithm described in (Godsill et al., 2007). We describe the VRPF algorithm next.

3.2 Variable Rate Particle Filter

For brevity, a brief description of the VRPF follows; details of the algorithm are available in (Godsill and Vermaak, 2005) and (Godsill et al., 2007). One of the assumptions made about the state evolution model is that the rate at which jumps occur is much lower than the state-magnitude itself. Thus, we require a way of redefining (10) and (15), to incorporate this fact. First, the posterior distribution that we are interested in is defined as -

$$f(x_{0:k_t^\square}|y_{0:t}) = f((\tau, \theta)_{0:k_t^\square}|y_{0:t}) \quad (17)$$

Where, k_t^\square is the index of the last jump time, τ_k , greater than the observation time, t ; i.e.,

$$k_t^\square = \min\{k : \tau_k > t\}$$

Recall from (4) that the state is a tuple, (τ, θ) , composed of the jump time and the state-magnitude resp.. This leads to a definition of the *neighborhood* of the t^{th} observation, y_t as -

$$\aleph_t = \{x_k : \tau_{k_{t-1}^\square+1} < \dots < t < \tau_{k_t^\square}\} \quad (18)$$

Accompanying the neighborhood structure, we also define a neighborhood function, $\hat{\phi}(t)$, that helps compute a neighborhood structure from the state-magnitude values. There are many possibilities for this function. In our case, we adopt the interpolation function defined in (Godsill et al., 2007), as it fits our problem well -

$$\hat{\phi}(t) = \frac{\theta_k(\tau_k - t) + \theta_{k-1}(t - \tau_{k-1})}{\tau_k - \tau_{k-1}}, \quad \tau_{k-1} \leq t < \tau_k$$

Now, using Bayes rule and the Markovian nature of our model, (17) can be expanded as -

$$\begin{aligned} f(x_{0:k_t^\square}|y_{0:t}) &= \frac{f(y_{0:t}|x_{0:k_t^\square}) f(x_{0:k_t^\square})}{f(y_{0:t})} \\ &\propto f(y_t|x_{\aleph_t}) f(x_{k_{t-1}^\square+1:k_t^\square}|x_{k_{t-1}^\square}) f(x_{0:k_{t-1}^\square}|y_{0:t-1}) \end{aligned} \quad (19)$$

(19) is the VRPF analogue of (15). Now, we can choose an appropriate importance density function, $q(x_{0:k_t^\square}|y_{0:t})$, such that it factorizes as in (14) -

$$q(x_{0:k_t^\square}|y_{0:t}) = q(x_{k_{t-1}^\square+1:k_t^\square}|x_{k_{t-1}^\square}, y_t) q(x_{0:k_{t-1}^\square}|y_{0:t-1}) \quad (20)$$

The importance weight update is then obtained by combining (19) and (20) in the equivalent of (13) -

$$\tilde{w}_t^{(i)} \propto \tilde{w}_{t-1}^{(i)} \frac{f(y_t|x_{\aleph_t}^{(i)}) f(x_{k_{t-1}^\square+1:k_t^\square}^{(i)}|x_{k_{t-1}^\square}^{(i)})}{q(x_{k_{t-1}^\square+1:k_t^\square}^{(i)}|x_{k_{t-1}^\square}^{(i)}, y_t)} \quad (21)$$

Where, the samples $x_k^{(i)}$ are drawn from the importance density -

$$x_{k_{t-1}^\square+1:k_t^\square}^{(i)} \sim q(x_{k_{t-1}^\square+1:k_t^\square}^{(i)} | x_{k_{t-1}^\square}^{(i)}, y_t)$$

From (19) we see that the posterior distribution depends upon the number of jumps, k_t^\square , made up to the t^{th} observation. This is a random variable itself, and this fact will have a bearing upon the design of the particle filter algorithm. Essentially the algorithm comes down to sampling jump time proposals from the importance density until the neighborhood of the current observation is *complete*. In our problem, the current observation y_t is basically the time between the latest observation and the last observation. Thus, if the last observation was at time instant v_{t-1} , then the instant of the current observation is $y_t + v_{t-1}$. Thus, completing the neighborhood requires drawing jump-times and corresponding state-magnitudes from the importance density till one of the jump-times is greater than $y_t + v_{t-1}$.

The algorithm performance hinges to a great extent on the choice of importance density, $q(\cdot)$. A simple and often used choice for the importance density is the state evolution density, (4). This type of a particle filter is called as a *bootstrap filter*. As noted in (Arulampalam et al., 2002), the state space is explored without any knowledge of the current observation. This could make the algorithm susceptible to perform poorly, when there are outliers in the data. However, since we assume that most consumers tend to have a fairly stable behavior, this might not cause a problem for us, and using the prior transition density as the importance density should not be a bad choice.

We will incorporate re-sampling into the algorithm. Re-sampling is performed when the Effective Sample Size (ESS) falls below some pre-defined threshold. ESS measures the number of samples that have significant weight, and is defined as -

$$N_{ess} = \frac{N_s}{1 + Var(w_k^*)}$$

Where, w^* is the *true* importance weight, and N_s is the number of particles. The ESS is approximated by,

$$\hat{N}_{ess} = \frac{1}{\sum (w_k^i)^2}$$

Particles are resampled using the sample-with-replacement regime, with the normalized weights as pseudo sample probabilities. Post re-sampling the weights of the sampled particles is set equal to $\frac{1}{N}$. The basic VRPF algorithm with re-sampling is listed in **Algorithm 1**.

Over time, the re-sampling regime suffers from a lack of sample diversity. As the re-sampling occurs, there is a greater tendency to sample the same few particles. Thus, even though the weight degeneracy is eliminated, another problem crops up. In order to correct for this problem, authors in the past have suggested adding *moves* to the resampled particles, see (Gilks and Berzuini, 2001), involving an MCMC kernel. The idea behind using moves is to adjust the position of the resampled particles, so that they are all not at the same location, and with the same

Algorithm 1 Bootstrap VRPF algorithm

- Initialize Particles

- 1: **for** i in 1 to N **do**
- 2: Set $\tau_0^{(i)} = 0$
- 3: Draw $\theta_0^{(i)} \sim \Gamma(\beta, \frac{1}{\beta})$
- 4: **end for**

- Start Algorithm

Require: T and N

1. Re-sampling

- 1: Compute $\hat{N}_{ess} = \frac{1}{\sum (w_k^i)^2}$
- 2: **if** $\hat{N}_{ess} < T$ **then**
- 3: **for** i in 1 to 10 **do**
- 4: Resample particle i with replacement, with probability $\sim w^{(i)}$
- 5: Set $w^{(i)} = \frac{1}{N}$
- 6: (optional) Move according to the prior dynamical density (4)
- 7: **end for**
- 8: **else**
- 9: Continue
- 10: **end if**

2. Propagation

- 1: **for** i in 1 to N **do**
 - 2: **Complete Neighborhood**
 - 3: $j = -1$
 - 4: **repeat**
 - 5: Increment j
 - 6: $\tau_{k_{t-1}^\square+j}^{(i)} \sim f(\tau | \tau_{k_{t-1}^\square+j}^{(i)}, \theta_{k_{t-1}^\square+j}^{(i)})$
 - 7: $\theta_{k_{t-1}^\square+j}^{(i)} \sim f(\theta | \theta_{k_{t-1}^\square+j}^{(i)})$
 - 8: Augment Sample Path
 - 9: **until** $\tau_{k_{t-1}^\square+j}^{(i)} > y_t$
 - 10: **Weight Computation**
 - 11: $\tilde{w}_t^{(i)} \propto w_{t-1}^{(i)} f(y_t | x_{\hat{s}_{t-1}})$
 - 12: **end for**
-

state-magnitude. This way, the diversity of the particles is ensured. Here we detail a simple particle movement regime.

The re-sampling regime involves moving the last sample in a particle according to a constrained prior dynamical density, at each re-sampling instant -

$$\begin{aligned} x_{k_t^\square}^{(i)} &\sim f(\tau, \theta | \tau_{k_t^\square-1}, \theta_{k_t^\square-1}, t) \\ &= f(\tau | \tau_{k_t^\square-1}, \theta_{k_t^\square-1}, \tau > t) f(\theta | \theta_{k_t^\square-1}) \end{aligned}$$

That is, the last sample in the particle sample path is moved to a new location according to the prior dynamical density, such that the jump instant is greater than the current observation instant. Next, we describe the experiments conducted and the results on these experiments.

4 Experiments and Results

We compare our algorithm to a standard exponentially weighted moving average (EWMA) filter, commonly used in time series filtering. As pointed out in the introduction, (Lambert et al., 2001) describe an algorithm for estimating the rate using a controlled version of the EWMA filter, such that a separate EWMA filter is maintained for each time interval of interest. We consider the simple version of the EWMA. We first briefly describe the EWMA and issues arising in its use, then describe the experiments run and finally present the results.

4.1 Exponentially Weighted Moving Average Filter

The EWMA is a very simple model that just averages the last n values of the observations to generate the current estimate of the mean value of the random variable of interest -

$$\bar{Y}_n = (1 - \alpha)\bar{Y}_{n-1} + \alpha x_n, \quad 0 < \alpha < 1$$

Where, Y_n and Y_{n-1} are the current and previous (resp.) estimated mean time between observed events, x_n is the current observation (time since the last transaction) and α is a fixed weighting value. The mean event rate we estimate is the inverse of the mean time between events. The question is, how does one chose the value of the weight α ? A general heuristic is to chose α such that $\alpha = \frac{2}{n+1}$. However, there is no set guideline and describing some point estimation method like maximum likelihood for this parameter is outside the purview of this paper.

4.2 Experiments

We ran the VRPF algorithm and the EWMA on data generated using the model described in section 2. Figure 2 shows the sample path that we consider in this test, along with the observation instants. This test can be thought of as a sanity check on the efficacy of the model, and also serves as a test-bed to compare a traditional solution (EWMA), with the proposed algorithm. We use the mean squared error (MSE) as the criterion to compare the algorithms -

$$MSE := \frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2$$

Where, n is the number of observations, x is the actual observation value and \hat{x} is the estimated value.

In order to generate the data, we set up the model with $\delta = 2.0$, $\alpha = 10.0$ and $\eta = 10.0$. Using this parametrization, we generated data such that there are exactly 10 jumps in the state process. We can see that figure 2 depicts different types of behavior - (from left to right on the time scale) jumping between moderate values, a jump to a very low rate, followed by a spike in the rate.

We use the same parametrization in the prior dynamical density in the VRPF. We present results for two versions of the algorithm, as noted above. First, we use simple re-sampling with no moves at each iteration of the algorithm, and a second version with a very simple move based on the prior dynamical density, (4). In the latter case we also present results with re-sampling when the effective sample size (ESS) falls below 0.4, and re-sampling at each iteration. In order to compare the VRPF algorithm with the EWMA, we ran the algorithms on the dataset above once, with the VRPF's initialized with 10,000 particles. For the EWMA, we ran the simulation multiple times, using different values for α in the set $\{0.1, 0.2 \dots 0.9\}$. We found that the best value of α in a mean-squared error sense is 0.2.

4.3 Results

Figure 3 shows the performance of the algorithms over the entire sample path. Figures 4(a), 4(b) and 4(c) show the performance at the rate spike, for clarity. We compare the EWMA to each of the VRPF algorithm flavors. It is clear that the particle filters are very close in performance to the EWMA, and indeed tend to converge to the actual value a lot quicker than the EWMA. However, we also see that the particle filter shows much more variance in the spikes, compared to the EWMA. This is most pronounced in the case of the VRPF algorithms with moves. Thus, incorporating moves into the algorithm certainly helps it converge faster, but it also tends to make it a bit more 'volatile'. The table below, table 1, shows the comparison of the algorithms using the MSE criterion. Clearly, the VRPF algorithms, VRPF_Move_ESS and VRPF_NoMove, perform better than the EWMA. Interestingly the VRPF with plain re-sampling at each iteration performed much better than the algorithms with moves.

Here, VRPF_Move_ESS denotes the VRPF algorithm with moves and re-sampling when ESS falls below 0.4, VRPF_Move_NoESS is the VRPF algorithm with moves and re-sampling at each iteration and VRPF_NoMove denotes the VRPF algorithm with only re-sampling at each iteration. In order to compare the VRPF algorithms themselves, we ran the algorithms 30 times on the same dataset, and estimated

Algorithm	MSE
EWMA	19.0839
VRPF_Move_ESS	15.4523
VRPF_Move_NoESS	25.8216
VRPF_NoMove	9.28208

Table 1: Comparison of EWMA to VRPF Algorithms

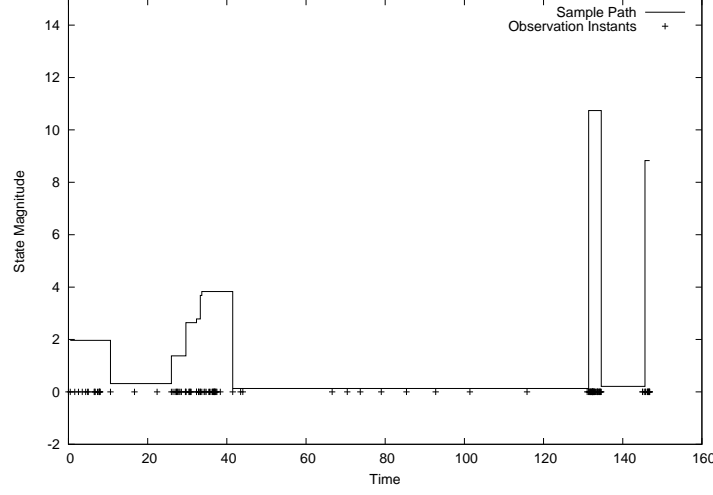


Figure 2: The figure shows the sample path of the state process that we use to compare the algorithms. The sample path is the solid line (-) and the observation instants are the '+’.

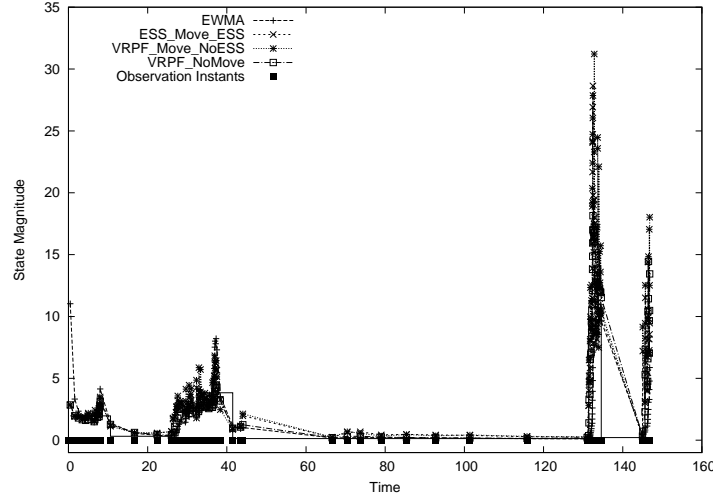


Figure 3: The relative performance of the algorithms on the entire sample path shown in figure 2. We show the estimated mean process.

the mean MSE over the runs. For this experiment, we changed the number of particles to 5,000. The mean MSE is estimated using-

$$\widehat{MSE} := \frac{1}{n} \sum_{j=1}^n \left\{ \frac{1}{m} \sum_{i=1}^m (x_{ji} - \hat{x}_{ji})^2 \right\}$$

Where, m is the number times the algorithms are run, n , as before, is the number of observations. The table below, table 2, summarizes the results of this test.

Figure 5 shows the performance of the VRPF algorithms at the rate spike. We can clearly see that the VRPF with no moves, figure 5(c), shows significant variation, as indicated by the large 95% confidence bands at each observation. Thus, even though we happened to obtain a good performance in the first experiment, the VRPF_NoMove is clearly a bad choice. The VRPF_Move_NoESS, figure 5(b), too, shows significant variance in the performance and the best algorithm would in fact be the VRPF with moves, and re-sampling when ESS falls below 0.4.

Algorithm	\widehat{MSE}
VRPF_MOVE_ESS	16.1922
VRPF_MOVE_NoESS	25.2604
VRPF_NoMove	24.7914

Table 2: Comparison of VRPF Algorithms

5 Discussion and Future Work

We have detailed a sequential Monte Carlo approach to estimating the rate at which customer initiated events are made. We assume that the events are governed by a doubly stochastic Poisson process. We assume that the rate process follows a Markov jump process, and we detailed an appropriate model of customer behavior. Based on the Poisson observation model and the Markov jump process model for the rate process, we showed how to use a particle filter,

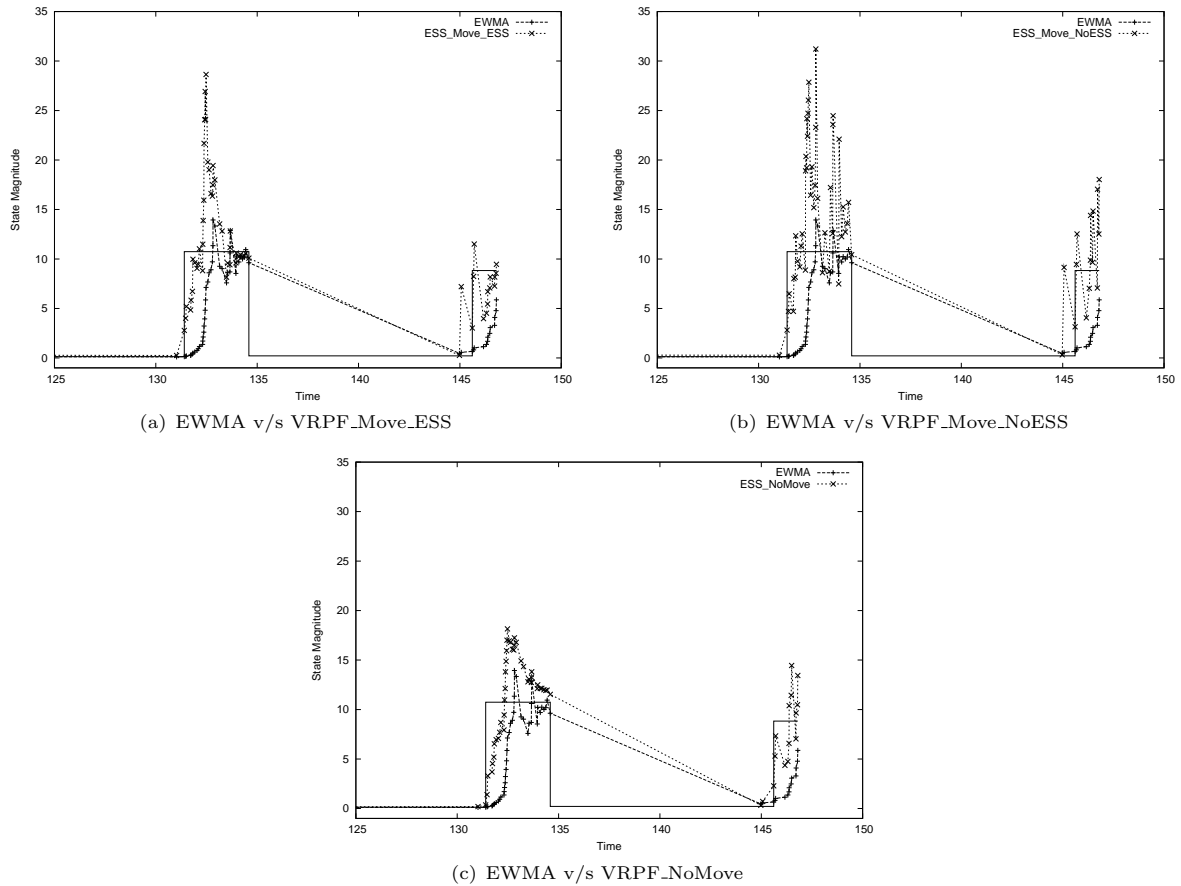


Figure 4: Relative Performance at the spike in the rate sample path. Here we compare the EWMA algorithm with each of the EWMA algorithm flavors. It is interesting to note that the EWMA_NoMove algorithm shows a much smoother behavior, with lesser variance in its estimates, compared to the other two algorithms. However, as seen in figure 5(c), below, it has significant variation from run to run.

the Variable Rate Particle Filter, to estimate the latent state. Our experiments show that the particle estimates of the mean rate is significantly better than the EWMA estimate of the state process sample, based on a mean squared error criterion.

5.1 Implications Of The Proposed Method

As noted in the introduction section, event rate estimation provides significant inputs towards solving many business problems, including fraud detection, click-stream analysis, targetted advertising etc. Consider, as an exemplar of these problems, fraud detection in credit cards. The events (card transactions) occur in a time-ordered fashion, with each consumer having a fairly stable transaction behavior. One of the measures of this stable behavior is the rate at which those transactions are made. Of course, in order to classify transactions as fraudulent or not, it is necessary to include a vast number of interesting features. However, knowledge of a customers historical transaction behavior is indispensable in detecting fraudulent behavior. The event rate is, arguably, the most important measure of past behavior. The change in transaction rate, in fact, can be an important marker for early detection of fraud events, as an indicator of abnormal consumer behavior. As another example, consider the case of click-stream analysis. By knowing how often a person browsing a website is likely to click on a link, it is possible to optimize the amount of caching to be done, for instance, to serve up webpages to millions of users simultaneously. Another example is in estimating

the amount of time a website patron spends on a particular page, thereby helping to optimize the amount of ad-spend on a particular page. Once again, knowledge of the event/transaction rate is very useful.

Thus, one can see that estimating the event rate is crucial to solving many important business problems. The estimate has to be as accurate as possible to be useful in any analytics used to solve these business problems. Our solution is to make some justifiable assumptions regarding the stochastic process that models the rate. Poisson models of events are widely used, and are justifiable by the great flexibility they afford. But, of course, the rate process is not observable, and has to be estimated from observations of the events alone. We showed that using a Bayesian method, Sequential Monte Carlo (SMC), the estimation of the rate from event observations is possible and with much better performance compared to the widely used EWMA approach. It is important to note that this paper lays out an extremely flexible framework for more accurate estimates of the rate process. Indeed, the experiments conducted, though on experimental data, clearly indicate the power and efficacy of the approach.

5.2 Future Directions

As future work, there are quite a few directions to consider -

- a) Recent results in (Whiteley et al., 2007) show that the SMC samplers framework, (Del Moral

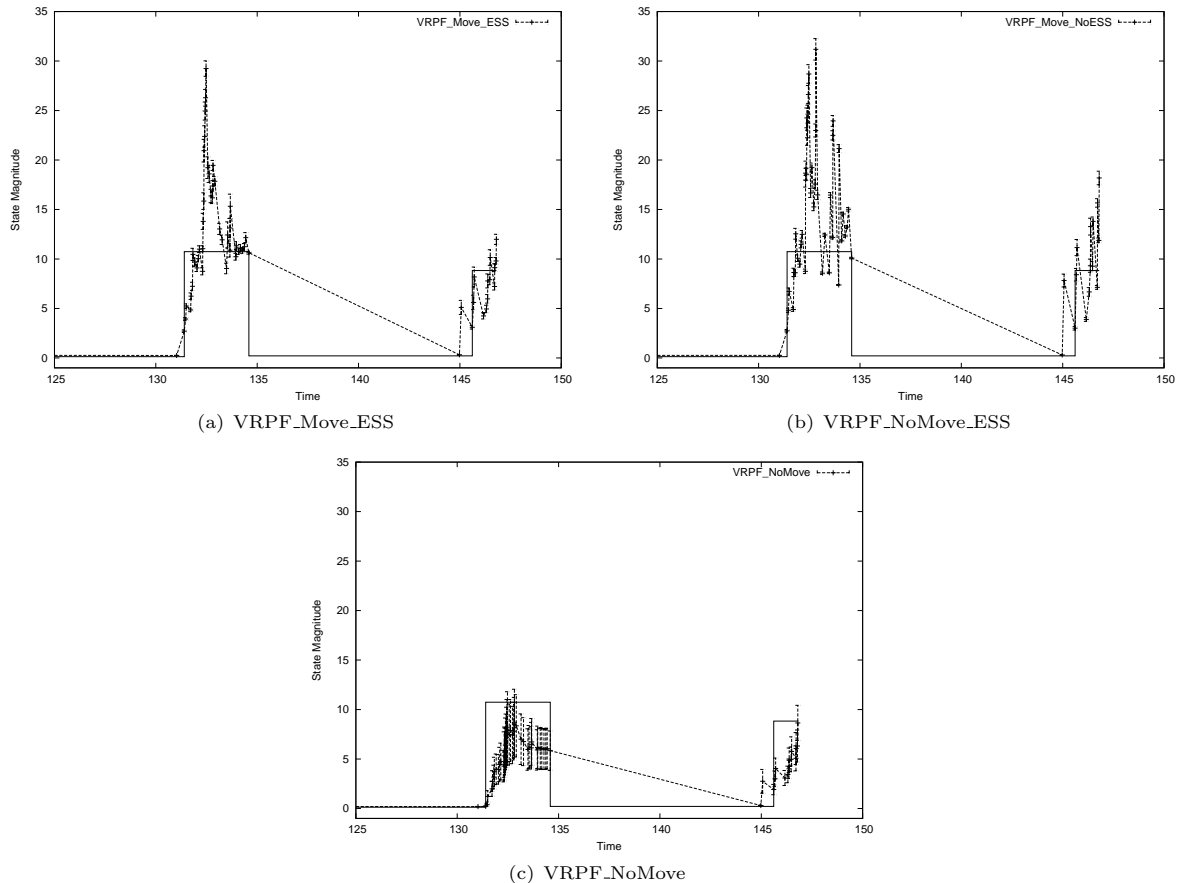


Figure 5: Relative Performance of the VRPF algorithms alone, at the spike in the rate sample path, over a 30 repeat run of the experiment. Also indicated are the 95% confidence bands for each estimate. Note that the VRPF_NoMove algorithm, (c), shows very significant confidence bands, compared to the other two algorithm flavors.

et al., 2006), is a better way of approaching particle filtering of MJP's. We intend to investigate this approach.

- b) The current work does not make any explicit reference to the time scale over which the state estimation is being done. In the introduction we mentioned a couple of approaches to solving the rate estimation problem in Poisson observed events - (Lambert et al., 2001) and (Weinberg et al., 2006). Both of these approaches incorporate controls into the models, so that the intra-day and inter-day rate variations can be captured. We need to introduce such controls into the model and the estimation algorithm.
- c) The results shown are on experimental data alone. It would be interesting to evaluate the approach on empirical data.
- d) Investigating the design of better importance densities for the MJP type setting. This point is related to a) above, since the SMC samplers framework allows for the design of better proposals, by considering mixtures of proposal kernels.
- e) The algorithm assumes knowledge of the parametrization of the jump time and state-magnitude distributions. This would require significant amount of data understanding in real world scenarios. Thus, there is also space to improve the algorithm by making it adaptive, and learn the distribution parameters from data.

References

- Arulampalam, M., Maskell, S., Gordon, N., Clapp, T., Sci, D., Organ, T. and Adelaide, S. (2002), 'A tutorial on particle filters for online nonlinear/non-GaussianBayesian tracking', *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]* **50**(2), 174–188.
- Bremaud, P. (1981), *Point Processes and Queues: Martingale Dynamics*, Springer-Verlag.
- Cappé, O., Godsill, S. and Moulines, E. (2007), 'An overview of existing methods and recent advances in sequential Monte Carlo', *Proceedings of the IEEE* **95**(5), 899–924.
- Daley, D. and Vere-Jones, D. (2003), *An Introduction to the Theory of Point Processes*, Springer.
- Dassios, A. and Jang, J. (2003), 'Pricing of catastrophe reinsurance and derivatives using the Cox process with shot noise intensity', *Finance and Stochastics* **7**(1), 73–95.
- Davis, M. (1984), 'Pieewise-deterministic Markov processes: a general class of nondiffusion stochastic models', *J. Roy. Statist. Soc. Ser. B* **46**(3), 353–388.
- Del Moral, P., Doucet, A. and Jasra, A. (2006), 'Sequential Monte Carlo samplers', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **68**(3), 411–436.

- Gilks, W. and Berzuini, C. (2001), ‘Following a moving target-Monte Carlo inference for dynamic Bayesian models’, *Journal of the Royal Statistical Society: Series B (Methodological)* **63**(1), 127–146.
- Godsill, S. and Vermaak, J. (2005), ‘Variable rate particle filters for tracking applications’, *Statistical Signal Processing, 2005 IEEE/SP 13th Workshop on* pp. 1280–1285.
- Godsill, S., Vermaak, J., Ng, W. and Li, J. (2007), ‘Models and Algorithms for Tracking of Maneuvering Objects Using Variable Rate Particle Filters’, *Proceedings of the IEEE* **95**(5), 925–952.
- Lambert, D., Pinheiro, J. and Sun, D. (2001), ‘Estimating Millions of Dynamic Timing Patterns in Real Time.’, *Journal of the American Statistical Association* **96**(453).
- Lando, D. (1997), ‘Modelling bonds and derivatives with default risk’, *Mathematics of Derivative Securities* pp. 369–393.
- Ross, S. (2007), *Introduction to Probability Models*, Academic Press.
- Scott, S. and Smyth, P. (2003), ‘The Markov Modulated Poisson Process and Markov Poisson Cascade with Applications to Web Traffic Modeling.’, *Bayesian Statistics*, 7.
- Weinberg, J., Brown, L. and Stroud, J. (2006), ‘Bayesian forecasting of an inhomogeneous Poisson process with applications to call center data’, *Journal of the American Statistical Association*.
- Whiteley, N., Johansen, A. and Godsill, S. (2007), ‘Monte Carlo Filtering of Piecewise Deterministic Processes’, *Technical Report CUED/F-INFENG/TR-592*, University of Cambridge, Department of Engineering.

Priority Driven K -Anonymisation for Privacy Protection

Xiaoxun Sun¹

Hua Wang¹

Jiuyong Li²

¹ Department of Mathematics & Computing
University of Southern Queensland
Toowoomba, Queensland 4350, Australia
Email: {sunx, wang}@usq.edu.au

² School of Computer and Information Science
University of South Australia, Adelaide, Australia
Email: jiuyong.li@unisa.edu.au

Abstract

Given the threat of re-identification in our growing digital society, guaranteeing privacy while providing worthwhile data for knowledge discovery has become a difficult problem. k -anonymity is a major technique used to ensure privacy by generalizing and suppressing attributes and has been the focus of intense research in the last few years. However, data modification techniques like generalization may produce anonymous data unusable for medical studies because some attributes become too coarse-grained. In this paper, we propose a priority driven k -anonymisation that allows to specify the degree of acceptable distortion for each attribute separately. We also define some appropriate metrics to measure the distance and information loss, which are suitable for both numerical and categorical attributes. Further, we formulate the priority driven k -anonymisation as the k -nearest neighbor (KNN) clustering problem by adding a constraint that each cluster contains at least k tuples. We develop an efficient algorithm for priority driven k -anonymisation. Experimental results show that the proposed technique causes significantly less distortions.

Keywords: K -Anonymity; Privacy Protection;

1 Introduction

Agencies and other organizations often need to publish microdata, e.g. medical data or census data, for research and other purpose. However, if individuals can be uniquely identified in the microdata then their private information (such as their medical condition) would be disclosed, and this is unacceptable. To avoid the identification of records in microdata, uniquely identifying information like names and social security numbers are removed from the table. Unfortunately, simply removing unique identifiers (e.g., names or phone numbers) from data is not enough, as individuals can still be identified when external data is linked to de-identified data, by using a combination of non-unique attributes such as age and postcode. Such non-unique attributes are often called quasi-identifiers (QIDs).

A recent study estimated that 87% of the population of the United States can be uniquely identified

“linking attack” using the seemingly innocuous attributes gender, date of birth, and 5-digit zip code (Sweeney 2000). To avoid linking attacks, Samarati and Sweeney (Samarati 2001, Sweeney 2002a) proposed a privacy principle called k -anonymity. It works by replacing a QID value with a more general one, such that the generalized QID values of each tuple are made identical to at least $k - 1$ other tuples in the anonymized table. This generalization process trades-off data utility for privacy protection. To illustrate this, consider Tables 1. Table 1(c) is a possible 2-anonymous view of Table 1(a). Here, queries such as “how many people live in an area with a postcode between 4350 and 4353 are male?” can no longer be answered accurately, and it is also more difficult to infer sensitive disease information about the individuals contained in the table.

Although the idea of k -anonymity is conceptually straightforward, the computational complexity of finding an optimal solution for the k -anonymity problem has been shown to be NP-hard, even when one considers only cell suppression (Aggarwal et al. 2005, Meyerson & Williams 2004, Sun et al. 2008b). The k -anonymity problem has recently drawn considerable interest from research community, and a number of algorithms have been proposed (Bayardo et al. 2005, Fung et al. 2005, Leferve et al. 2005, Sweeney 2002b, Sun et al. 2008a). Current solutions, however, suffer from high cost of information loss mainly due to relying on pre-defined generalization hierarchies (Fung et al. 2005, Leferve et al. 2005, Sweeney 2002b, Sun et al. 2008a) or total order imposed on each attribute domain (Bayardo et al. 2005). A more general view of k -anonymisation is clustering with a constraint of the minimum number of objects in every cluster (Aggarwal et al. 2006, Byun et al. 2006). A number of methods approach identity protection by clustering (Agrawal 2001, Aggarwal 2005). However, these methods are applicable to numerical attributes only. A recent work (Domingo-Ferrer et al. 2005) extends a clustering-based method (Domingo-Ferrer et al. 2002) to ordinal attributes, but it does not deal with attributes in hierarchical structures.

In this paper, we propose a priority driven k -anonymisation that allows to specify the degree of acceptable distortion for each attribute separately. We also define some appropriate metrics to measure the distance and information loss, which are suitable for both numerical and categorical attributes. Further, we formulate the priority driven k -anonymisation as the k -nearest neighbor (KNN) clustering problem by adding a constraint that each cluster contains at least k tuples. We develop an efficient algorithm for priority driven k -anonymisation. Experimental results show that the proposed technique causes significantly less distortions.

Copyright©2008, Australian Computer Society, Inc. This paper appeared at conference Seventh Australasian Data Mining Conference (AusDM 2008), Glenelg, Australia. Conferences in Research and Practice in Information Technology, Vol. 87. John F. Roddick, Jiuyong Li, Peter Christen and Paul Kennedy, Eds. Reproduction for academic, not-for profit purposes permitted provided this text is included.

Gender	Age	Pcode	Problem	Gender	Age	Pcode	Problem	Gender	Age	Pcode	Problem
male	middle	4350	stress	male	middle	4350	stress	*	middle	435*	stress
male	middle	4350	obesity	male	middle	4350	obesity	*	middle	435*	obesity
male	young	4351	stress	*	young	435*	stress	*	young	435*	stress
female	young	4352	obesity	*	young	435*	obesity	*	young	435*	obesity
female	old	4353	stress	female	old	4353	stress	*	old	435*	stress
female	old	4353	obesity	female	old	4353	obesity	*	old	435*	obesity

Table 1: (a) Left: a raw table. (b) Middle: a 2-anonymous table by local recoding. (c) Right: a 2-anonymous view by global recoding.

2 Preliminary Definitions

The objective of k -anonymisation is to make every tuple of privacy-related attributes in a published table identical to at least $k - 1$ other tuples. As a result, no privacy-related information can be easily inferred. For example, young people with stress and obesity are potentially identifiable by their unique combinations of gender, age and postcode attributes in Table 1(a). To preserve their privacy, we may generalize their gender and postcode attribute values such that each tuple in attribute set {Gender, Age, Postcode} has two occurrences. The view after the generalization is listed in Table 1(b).

Definition 1 A *quasi-identifier (QID) attribute set* is a set of attributes in a table that potentially reveal private information, possibly by joining with other tables. A *QI-group* of a table with respect to the QID attribute set is the set of all tuples in the table containing identical values for the QID attribute set.

For example, the attribute set {Gender, Age, Postcode} in Table 1(a) is a QID and Tuples 1 and 2 in Table 1(b) form a QI-group with respect to this QID since their corresponding values are identical. Table 1(a) potentially reveals private information of patients (e.g. young patients with stress and obesity). If the table is joined with other tables, it may reveal more information of patients' disease history. Normally, the QID set is understood by domain experts.

Definition 2 (k -anonymity) A table is called k -anonymous with respect to a QID if the size of every QI-group with respect to the QID set is at least k .

k -anonymity requires that every occurrence within an attribute set has the frequency at least k . For example, Table 1(a) does not satisfy 2-anonymity property since tuples male, young, 4351 and female, young, 4352 occur only once. Table 1(b) is a 2-anonymous view of Table 1(a) since the size of all QI-group with respect to the QID is 2.

Another objective for k -anonymisation is to minimize distortions. A table may have more than one k -anonymous views, but some are better than others. For example, we may have another 2-anonymous view of Table 1(a) as in Table 1(c). Table 1(c) loses much more information than Table 1(b).

In the literature of k -anonymity problem, there are two main models. One model is global recoding (Fung et al. 2005, Leferve et al. 2005, Sweeney 2002a, Samarati 2001), while the other is local recoding (Aggarwal et al. 2005, Sweeney 2002b). Here, we assume that each attribute has a corresponding conceptual generalization hierarchy or taxonomy tree. A lower level domain in the hierarchy provides more details than a higher level domain. For example, Postcode 4350 is a lower level domain and Postcode 435* is a higher level domain. We assume such hierarchies for numerical attributes too. In particular, we have a hierarchical structure defined with {value, interval, *}, where value is the raw numerical data, interval is the range of the raw data and * is a symbol representing any values. Generalization replaces lower level

domain values with higher level domain values. For example, Age 27, 28 in the lower level can be replaced by the interval (27-28) in the higher level. Examples of global and local recoding are shown in Table 1(b) and Table 1(c).

Definition 3 ((Li et al. 2006)) Let h be the height of a domain hierarchy, and let levels $1, 2, \dots, h - 1, h$ be the domain levels from the most general to most specific, respectively. Let the weight between domain level $j - 1$ and j be predefined, denoted by $w_{j,j-1}$, where $2 \leq j \leq h$. When a cell is generalized from level p to level q , where $p > q$. The weighted hierarchical distance of this generalization is defined as:

$$WHD(p, q) = \frac{\sum_{j=q+1}^p w_{j,j-1}}{\sum_{j=2}^h w_{j,j-1}}$$

In the following, we discuss two simple but typical schemes to define $w_{j,j-1}$.

(1). Uniform Weight: $w_{j,j-1} = 1$ ($2 \leq j \leq h$)

This is the simplest scheme where all weights are equal to 1. In this scheme, WHD is the number of steps a cell being generalized over all possible generalization steps. For example, let birth date hierarchy be {D/M/Y, M/Y, Y, 10Y, C/Y/M/O, *}, where 10Y stands for 10-year interval and C/Y/M/O for child, young, middle age and old age. WHD from D/M/Y to Y is $WHD(6,4) = (1+1)/5 = 0.4$. In gender hierarchy, {M/F, *}, WHD from M/F to * is $WHD(2,1) = 1/1 = 1$. This means that the distortion caused by the generalization of five cells from D/M/Y to Y is equivalent to the distortion caused by the generalization of two cells from M/F to *.

(2). Height Weight: $w_{j,j-1} = \frac{1}{(j-1)^\beta}$ ($2 \leq j \leq h$, $\beta \geq 1$).

For a fixed β , the intuition of this scheme is that the generalization near to the top should give greater distortion compared with the generalization far from the top. Thus, we formulate the height weight scheme, where the weight near to the top is larger and the weight far from the top is smaller. For example, consider a hierarchy: {D/M/Y, M/Y, Y, 10Y, C/Y/M/O, *} for birth date. Let $\beta = 1$. WHD from D/M/Y to M/Y is $WHD(6,5) = (1/5)/(1/5 + 1/4 + 1/3 + 1/2 + 1) = 0.087$. In gender hierarchy {M/F, *}, WHD from M/F to * is $WHD(2,1) = 1/1 = 1$. The distortion caused by the generalization of one cell from M/F to * in gender attribute is more than the distortion caused by the generalization of 11 cells from D/M/Y to M/Y in birth date attribute.

In some cases, attributes should be generalized only up to a certain degree or not transformed at all. Otherwise, their values become useless for an application domain. Priorities are used to specify the degree of desired anonymisation of attributes. In some applications, exact values for a specific attribute may be favored while the generalization degree of others is negligible. By specifying priorities the user is able to determine the degree of generalization and information loss s/he is willing to cope with. Attributes with lower priorities are generalized first while attributes

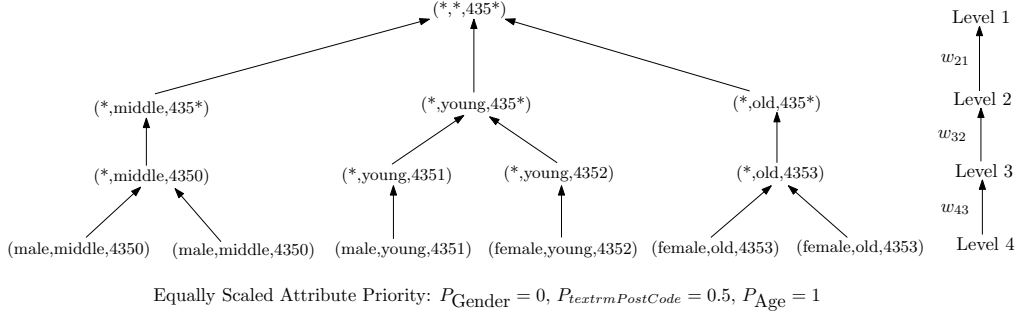


Figure 1: Generalization Hierarchy with Equally Scaled Attribute Priority

with higher priorities are only generalized when no other solution may be found. Priorities have values in the range $[0,1]$. The most important attribute has the highest priority value and the differences between any two consecutive priorities values are determined by a pre-defined weight $w'_{j,j-1}$.

Definition 4 (Attribute Priority) A priority $P_j \in [0,1]$ is assigned to each attribute α_j . Suppose attributes $\alpha_1, \alpha_2, \dots, \alpha_m$ sorted by their priorities, where α_1 is the highest and α_m is the lowest one and let the weight between attribute α_{j-1} and α_j be predefined, denoted by $w'_{j,j-1}$, where $2 \leq j \leq m$. Then, the priority P_i of attribute α_i is defined as:

$$P_j = \begin{cases} 1 & \text{if } j = 1 \\ 1 - w'_{j,j-1} \cdot \frac{j-1}{m-1} & \text{if } 2 \leq j \leq m \end{cases}$$

We can similarly define $w'_{j,j-1}$ as $w_{j,j-1}$. For the sake of simplicity, in this paper, we discuss the equally scaled attribute priority, i.e., when $w'_{j,j-1} = 1$ ($2 \leq j \leq m$). The following example illustrates the equally scaled priority values: $P_{\text{Gender}} = 0$, $P_{\text{textrmPostCode}} = 0.5$, $P_{\text{Age}} = 1$. According to this priority scheme, Gender is generalized first, Postcode next and Age the last. Because the anonymous solution is obtained after the generalization of Gender and Postcode, so no generalization needed for Age. (see Table 1(b)).

Priorities are used to weight the information loss (distortion) quantifiers of generalized attribute values. Hence, in the final generalization solution the information loss for attribute Age should be much smaller than for attribute Gender. In other words, attribute Gender might be transformed to a more general value than attribute Age, which is the case in our example.

In the following, we define distortions (information loss) caused by the generalization of tuples and tables.

Definition 5 (Weighted Tuple Distortions)

Let $t = \{v_1, v_2, \dots, v_m\}$ be a tuple and $t' = \{v'_1, v'_2, \dots, v'_m\}$ be a generalized tuple of t . Let $\text{level}(v_j)$ be the domain level of v_j in the attribute hierarchy of α_j and P_j is the attribute priority of α_j . Then, the distortion of this generalization is defined as:

$$\text{Distortion}(t, t') = \sum_{j=1}^m P_j \cdot \text{WHD}(\text{level}(v_j), \text{level}(v'_j))$$

Different from (Li et al. 2006), our distortion function is the weighted version which specifies the attribute priority. For example, let the weights of WHD be defined by the uniform weight, attribute Gender be in hierarchy of $\{M/F, *\}$ and attribute Postcode be in hierarchy of $\{dddd, ddd*, dd**, d***, *\}$.

$d***, *\}$. $P_{\text{Gender}} = 0$, $P_{\text{textrmPostCode}} = 0.5$, and $P_{\text{Age}} = 1$ are the equally scaled priority values. Let t_3 be tuple 3 in Table 1(a) and t'_3 be tuple 3 in Table 1(b). For attribute Gender, $\text{WHD}=1$. For attribute Postcode, $\text{WHD}=1/4=0.25$. For attribute Age, $\text{WHD}=0$. Therefore, $\text{Distortion}(t_3, t'_3) = 1*0 + 0.25*0.5 + 0*1 = 0.125$. Compare with (Li et al. 2006), our measurement causes less distortion.

Similar with (Li et al. 2006), we can define the total distortion for the table.

Definition 6 Let T' be generalized from table T , t_j be the j^{th} tuple in T and t'_j be the j^{th} tuple in T' . Then, the distortion of this generalization is defined as:

$$\text{Distortion}(T, T') = \sum_{j=1}^{|T|} \text{Distortion}(t_j, t'_j)$$

where $|T|$ is the number of tuples in T .

From Table 1(a) and (b), $\text{Distortion}(t_1, t'_1) = \text{Distortion}(t_2, t'_2) = \text{Distortion}(t_5, t'_5) = \text{Distortion}(t_6, t'_6) = 0$, and $\text{Distortion}(t_3, t'_3) = \text{Distortion}(t_4, t'_4) = 0.125$. So, the total distortion between the two tables is $\text{Distortion}(T, T') = 0.125 + 0.125 = 0.25$.

Definition 7 All allowable values of an attribute form a hierarchical value tree. Each value is represented as a node in the tree, and a node has a number of child nodes corresponding to its more specific values. Let t_1 and t_2 be two tuples. t_c is the closest common generalization of t_1 and t_2 for all attributes α_j ($1 \leq j \leq m$). Then, t_c is defined as:

$$v_c^j = \begin{cases} v_1^j & \text{if } v_1^j = v_2^j \\ \text{the closest common ancestor} & \text{Otherwise} \end{cases}$$

For example, Figure 1 shows a hierarchical structure with four domain levels. Let $t_1 = \{\text{male, young, 4351}\}$ and $t_2 = \{\text{female, young, 4352}\}$, then $t_c = \{*, \text{young, 435*}\}$. Now, we define the distance between two tuples.

Definition 8 Let t_1, t_2 be two tuples and t_c be their closest common generalization. Then, the distance between t_1 and t_2 is defined as:

$$\text{Dist}(t_1, t_2) = \text{Distortion}(t_1, t_c) + \text{Distortion}(t_2, t_c)$$

For example, let the weights of WHD be defined by the uniform weight, attribute Gender and Postcode be in hierarchy shown in Figure 1. $t_1 = \{\text{male, young, 4351}\}$ and $t_2 = \{\text{female, young, 4352}\}$. Then, $t_c = \{*, \text{young, 435*}\}$ and $\text{Dist}(t_1, t_2) = \text{Distortion}(t_1, t_c) + \text{Distortion}(t_2, t_c) = 0.125 + 0.125 = 0.25$. We discuss some properties of the distance metric in the following.

Theorem 1 *The distance between two tuples t_1 and t_2 $Dist(t_1, t_2)$ satisfies the following properties:*

- (1) $Dist(t_1, t_1)=0$;
- (2) $Dist(t_1, t_2)=Dist(t_2, t_1)$;
- (3) $Dist(t_1, t_3) \leq Dist(t_1, t_2) + Dist(t_2, t_3)$

3 KNN-Clustering Problem

Typical clustering problems require that a specific number of clusters be found in solutions. However, the k -anonymity problem does not have a constraint on the number of clusters; instead, it requires that each cluster contains at least k tuples. Thus, we pose the k -anonymity problem as a clustering problem, referred to as k -Nearest Neighbor(KNN) Clustering Problem.

Definition 9 (KNN Clustering Problem) *The k -Nearest Neighbor(KNN) Clustering Problem is to find a set of clusters from a given set of n tuples such that each cluster contains k ($k \leq n$) data points and that the average intra-cluster distances is minimized.*

The distance functions that measure the similarities among data points and the cost function which the clustering problem tries to minimize are the heart of every clustering problem. The distance functions are usually determined by the type of data being clustered, while the cost function is defined by the specific objective of the clustering problem. In this section, we describe our distance and cost functions which have been specifically tailored for the priority driven k -anonymisation problem.

Distance Function: A distance function in a clustering problem measures how dissimilar two data points are. As the data we consider in the k -anonymity problem are person-specific records that typically consist of both numeric and categorical attributes, we need a distance function that can handle both types of data at the same time. We are aware that the distance metric $Dist()$ defined in Section ?? can deal with both categorical and numeric attributes, so we introduce a density metric called k -Nearest Neighbor(KNN) distance which is defined as follow:

Definition 10 (KNN Distance) *Let T be a set of tuples and t be a tuple in T , and $DistK(i)$ ($i = 1, 2, \dots, k$) be the minimal k values in all $Dist(t, t_j)$ ($1 \leq j \leq |T|$). Then, the KNN distance of t is defined as:*

$$DistKNN(t) = \frac{\sum_{i=1}^k DistK(i)}{k}$$

where $|T|$ is the number of tuples in T .

Definition 11 (Density) *Let $DistKNN(t)$ be the KNN distance of tuple $t \in T$. Then, the density of t is defined as:*

$$Density(t) = \frac{1}{DistKNN(t)}$$

The smaller the distances between t and other records around it are, the larger the density of t is. The tuple (record) with larger density will be made as a cluster center with high probability because the cluster has a smaller distortion. Next, we discuss the cost function which the KNN Clustering Problem.

Cost Function: As the ultimate goal of our clustering problem is the k -anonymisation of data, we formulate the cost function as in Definition 6 to represent the amount of distortion (i.e., information loss) caused by the generalization process. Note that in the rest of the paper, for a table T , to make the notions simple, we use $Distortion(T)$ rather than $Distortion(T, T')$ to represent the distortion between T and its generalized form T' .

As in most clustering problems, an exhaustive search for an optimal solution of the KNN-clustering problem is potentially exponential. Since the k -anonymity problem is shown NP-hard (Aggarwal et al. 2005, Meyerson & Williams 2004, Sun et al. 2008b), and it is also a special case of priority driven k -anonymity problem when each attribute has the same priority, so the priority driven k -anonymity problem is NP-hard as well. Because of the hardness of the problem, we propose a simple and efficient density-based clustering algorithm. The idea is as follows. Given a set T of $|T|$ records, the choice of cluster center points can be based on the distribution density of data points. We pick a record $t \in T$ whose density is the maximal and make it as the center of a cluster C . Then we add $k - 1$ records which have minimal distance with t to C . Choose the next cluster center and repeat the clustering process until there are less than k records left. We then iterate over these leftover records and insert each record into a cluster with respect to which the increment of the distortion is minimal.

How to choose the next cluster center is another important issue when one iteration has finished, because we consider that the next cluster center is a record which has the maximal density in remainder records. The next cluster center is not in the k -nearest-neighbor records of this center, thus a principle of choosing the next cluster center is needed.

Definition 12 *Let T be a set of records, t_C be a center of cluster C and t'_C be the next cluster center. The choice of $t'_C \in T \setminus C$ must satisfy the follow two requirements:*

$$Density(t'_C) = \max\{Density(t_i), t_i \in \{T \setminus C\}\}$$

$$Dist(t_C, t'_C) > DistKNN(t_C) + DisKNN(t'_C)$$

As the algorithm finds a cluster with exactly k records as long as the number of remaining records is equal to or greater than k , every cluster contains at least k records. If there remain less than k records, these leftover records are distributed to the clusters that are already found. That is, in the worst case, $k - 1$ remaining records are added to a single cluster which already contains k records. Therefore, the maximum size of a cluster is $2k - 1$. The total time complexity is in $O(n^2)$.

The focus of most k -anonymity work is heavily placed on the QID, and therefore other attributes are often ignored. However, these attributes deserve more careful consideration. In fact, we want to minimize the distortion of QID not only because the QID itself is meaningful information, but also because a more accurate QID will lead to good predictive models on the transformed table. In fact, the correlation between the QID and other attributes can be significantly weakened or perturbed due to the ambiguity introduced by the generalization of the QID. Thus, it is critical that the generalization process does preserve the discrimination of classes using QID. Iyengar (Iyengar 2002) proposed the classification metric (CM) as:

$$CM = \frac{\sum_{\text{all rows}} \text{Penalty}(\text{row } r)}{|T|}$$

Attribute	Distinct Values	Generalizations	Height
Age	74	5-,10-20-year range	5
Work class	8	Taxonomy Tree	3
Education	16	Taxonomy Tree	4
Country	41	Taxonomy Tree	3
Marital Status	7	Taxonomy Tree	3
Race	5	Taxonomy Tree	3
Occupation	14	Taxonomy Tree	2
Gender	2	Suppression	21
Salary class	2	Suppression	1

Table 2: Features of Adult Dataset

where $|T|$ is the total number of records, and $\text{Penalty}(\text{row } r)=1$ if r is suppressed or the class label of r is different from the class label of the majority in the QI-group.

Inspired by this, the algorithm is now forced to choose clusters with the same class label for a record, and the enforcement is controlled by the row penalty. We show the results in Section 4 that our modified algorithm can effectively reduce the cost of classification metric without increasing much distortion.

4 Empirical Study

The main goal of the experiments was to investigate the performance of our approach in terms of data quality, efficiency. To evaluate our approach, we also compared our implementation with another algorithm, namely the *median partitioning algorithm*(MPA) proposed in (Leferve et al. 2006). We conduct the experiments with two type of distortion measurement discussed in Section ??-weighted hierarchical distance and attribute priority.

In our experiment, we adopted the publicly available data set, Adult Database, at the UC Irvine Machine Learning Repository¹, which has become the benchmark for evaluating the performance of k -anonymity algorithms adopted by (Leferve et al. 2005, 2006, Fung et al. 2005, Sun et al. 2008c). We eliminated the records with unknown values. The resulting data set contains 45222 tuples. For k -anonymisation, we considered {age, work class, education, marital status, occupation, race, gender, and native country} as the QID. In addition to that, we also retained the salary class attribute to evaluate the classification metric (CM). The feature of QIDs is shown in Table 2.

We report experimental results on the Density-Based Clustering Algorithm(DBCA) and its modification to reduce classification error(DBCA:CM) for data quality and execution efficiency.

Figure 2 reports the Total distortion of the three algorithms (MPA, DBCA, and DBCA:CM). For increasing values of k . As the figure illustrates, the DBCA:CM algorithm results in the least distortion for all k values. Note also that the distortion of DBCA is very close to the modified DBCA:CM. The superiority of our algorithms over the MPA results from the fact that the MPA considers the proximity among the data points only with respect to a single dimension at each partitioning.

Figure 3 shows the experimental result with respect to the CM metric. As expected, the DBCA:CM modified to minimize classification errors outperforms all the other algorithms. Observe that even without the modification, the DBCA still produces less classification errors than the MPA for every k value. We also measured the execution time of the algorithms for different k values. The results are shown in Figure 4. Even though the execution time for the DBCA

is higher than the MPA, we believe that it is still acceptable in practice as k -anonymisation is often considered an off-line procedure.

5 Conclusion and Future Work

In this paper, we propose a priority-driven anonymisation technique that allows to specify the degree of acceptable distortion for each attribute separately. We define generalization distances between tuples to characterize distortions by generalizations, which works for both numerical and categorical attributes. Further, we propose a density-based clustering technique to minimize information loss and thus ensure good data quality. We experimentally show that the proposed method is more scalable and causes significantly less distortions than an optimal k -anonymity method.

In the future work, we focus on two important extensions. First, we would try to extend this priority driven anonymisation framework to other privacy requirements, like (p^+, α) -sensitive k -anonymity (Sun et al. 2008c), l -diversity (Machanavajjhala et al. 2006), (α, k) -anonymity (Li et al. 2006) and t -closeness (Li et al. 2007), etc, to make it a systematic approach. Second, we could like to do more experimental studies to compare the performance with other clustering methods (Byun et al. 2006).

Acknowledgement

We would like to thank anonymous reviewers for their useful comments on this paper. This research was supported by Australian Research Council (ARC) grant DP0774450 and DP0663414.

References

- Aggarwal, G & Feder, T & Kenthapadi, K & Motwani, R & Panigrahy, R & Thomas, D & Zhu, A (2005), Anonymizing tables, *in* In Proc. of the 10th International Conference on Database Theory, pp. 246-258, Edinburgh, Scotland.
- Aggarwal, G & Feder, T & Kenthapadi, K & Zhu, A & Panigrahy, R & Tomas, D (2006), Achieving anonymity via clustering in a metric space, *in* in Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, 2006.
- Aggarwal, C. C (2005), On k -anonymity and the curse of dimensionality, Proceedings of the 31st international conference on Very large data bases, pages 901-909. VLDB Endowment, 2005.
- Agarwal, D & Aggarwal, C. C (2001), On the design and quantification of privacy preserving data mining algorithms, in Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium

¹available at www.ics.uci.edu/~mllearn/MLRepository.html

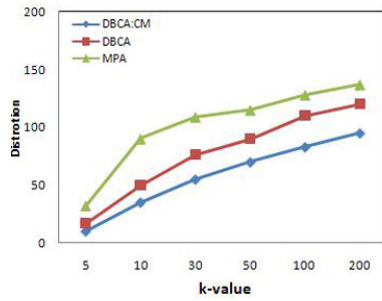


Figure 2: Distortion Metric

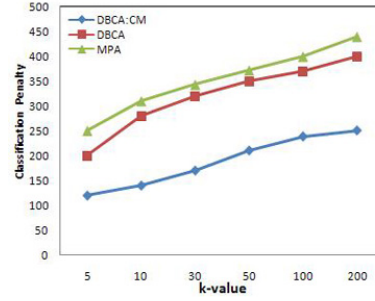


Figure 3: Classification Metric

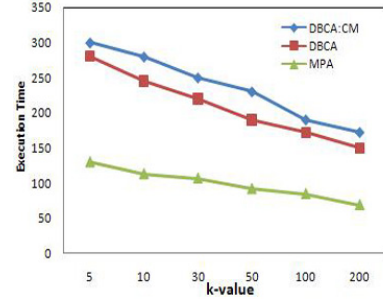


Figure 4: Running Time

on Principles of database systems, pages 247-255, New York, NY, USA, 2001. ACM Press.

Byun, J. W., & Kamra, A & Bertino, E & Li, N (2006), Efficient k -Anonymity using Clustering Technique, CERIAS Tech Report 2006-10, 2006.

Fung, B & Wang, K & Yu, P (2005), Top-down specialization for information and privacy preservation, In Proc. of the 21st International Conference on Data Engineering, Tokyo, Japan.

Bayardo, R & Agrawal, R (2005), Data privacy through optimal k -anonymity, In Proceedings of the 21st International Conference on Data Engineering (ICDE) 2005.

Domingo-Ferrer, J & Torra, V (2005), Ordinal, continuous and heterogeneous k -anonymity through microaggregation, Data Mining and Knowledge Discovery, 11(2):195-212, 2005.

Domingo-Ferrer, J & Mateo-Sanz, J. M (2005), Practical data-oriented microaggregation for statistical disclosure control, IEEE Transactions on Knowledge and Data Engineering, 14(1):189-201, 2002.

Iyengar (2002), Transforming data to satisfy privacy constraints. In ACM Conference on Knowledge Discovery and Data mining, 2002.

Leferve, K & Dewitt, D & Ramakrishnan, R. (2005), Incognito: Efficient Full-Domain k -Anonymity, ACM SIGMOD International Conference on Management of Data, June 2005.

Leferve, K & Dewitt, D & Ramakrishnan, R. (2006), Mondrian multidimensional k -anonymity. In International Conference on Data Engineering, 2006.

Machanavajjhala, A & Gehrke, J & Kifer, D & Venkatasubramanian, M (2006), l -Diversity: Privacy beyond k -anonymity, In ICDE, 2006.

Meyerson, A & Williams, R (2004), On the complexity of optimal k -anonymity, in Proc. of the 23rd ACM-SIGMOD-SIGACT-SIGART Symposium on the Principles of Database Systems, pp. 223-228, Paris, France, 2004.

Li, N & Li, T & Venkatasubramanian, S (2007), t -Closeness: Privacy Beyond k -Anonymity and l -Diversity, In 23rd IEEE International Conference on Data Engineering (ICDE), April 2007

Samarati, P (2001), Protecting respondents' identities in microdata release, IEEE Transactions on Knowledge and Data Engineering, 13(6):1010-1027, 2001

Sun, X & Li, M & Wang, H & Plank, A (2008), An efficient hash-based algorithm for minimal k -anonymity. In: 31st Australasian Computer Science Conference (ACSC 2008), 22-25 Jan 2008, Wollongong, NSW, Australia.

Sun, X & Wang, H & Li, J (2008), On the complexity of restricted k -anonymity problem, The 10th Asia Pacific Web Conference (APWEB2008), LNCS 4976, pp: 287-296, Shenyang, China. 2008.

Sun, X & Wang, H & Li, J & Traian, T. M & Ping, L (2008), (p^+, α) -sensitive k -anonymity: a new enhanced privacy protection model. In 8th IEEE International Conference on Computer and Information Technology (IEEE-CIT 2008), 8-11 July 2008, Sydney, Australia. pp:59-64.

Sweeney, L (2002), Achieving k -anonymity Privacy Protection Using Generalization and Suppression, International Journal of Uncertainty, Fuzziness and Knowledge-Based System, 10(5) pp. 571-588, 2002.

Sweeney, L (2002), k -anonymity: A Model for Protecting Privacy, International Journal on Uncertainty Fuzziness Knowledge-based Systems, 10(5), pp 557-570, 2002.

Sweeney, L (2000), Uniqueness of simple demographics in the u.s. population, Technical report, Carnegie Mellon University, 2000.

Wong, R & Li, J & Fu, A & Wang, K (2006), (α, k) -anonymity: an enhanced k -anonymity model for privacy preserving data publishing, *KDD 2006*: 754-759.

Li, J & Wong, R & Fu, A & Pei, J (2006), Achieving k -Anonymity by clustering in attribute hierarchical structures. In: 8th International Conference on Data Warehousing and Knowledge Discovery, 4-8 Sept 2006, Krakow, Poland.

ShrFP-Tree: An Efficient Tree Structure for Mining Share-Frequent Patterns

Chowdhury Farhan Ahmed, Syed Khairuzzaman Tanbeer,
Byeong-Soo Jeong, and Young-Koo Lee

Database Lab, Department of Computer Engineering, Kyung Hee University
1 Seochun-dong, Kihung-gu, Youngin-si, Kyunggi-do, 446-701, Republic of Korea
Email: {farhan, tanbeer, jeong, yklee}@khu.ac.kr

Abstract

Share-frequent pattern mining discovers more useful and realistic knowledge from database compared to the traditional frequent pattern mining by considering the non-binary frequency values of items in transactions. Therefore, recently share-frequent pattern mining problem becomes a very important research issue in data mining and knowledge discovery. Existing algorithms of share-frequent pattern mining are based on the level-wise candidate set generation-and-test methodology. As a result, they need several database scans and generate-and-test a huge number of candidate patterns. Moreover, their numbers of database scans are dependent on the maximum length of the candidate patterns. In this paper, we propose a novel tree structure ShrFP-Tree (Share-frequent pattern tree) for share-frequent pattern mining. It exploits a pattern growth mining approach to avoid the level-wise candidate set generation-and-test problem and huge number of candidate generation. Its number of database scans is totally independent of the maximum length of the candidate patterns. It needs maximum three database scans to calculate the complete set of share-frequent patterns. Extensive performance analyses show that our approach is very efficient for share-frequent pattern mining and it outperforms the existing most efficient algorithms.

Keywords: Data mining, Knowledge discovery, Frequent pattern mining, Share-frequent pattern mining, Pattern growth mining.

1 Introduction

Frequent pattern mining (Agrawal et al., 1993; Agrawal and Srikant, 1994; Han et al., 2004, 2007) plays an important role in data mining and knowledge discovery techniques such as association rule mining, classification, clustering, time-series mining, graph mining, web mining etc. The initial solution of frequent pattern mining is the Apriori algorithm (Agrawal et al., 1993; Agrawal and Srikant, 1994) which is based on the level-wise candidate set generation-and-test methodology and needs several database scans. For the first database scan, it finds all the single-element frequent patterns and based on that result it generates the candidates for two-element frequent patterns. In the second database scan, it finds all the two-element frequent patterns and based on that result it generates the candidates for three-

element frequent patterns and so on. FP-growth (Han et al., 2004) solved this problem by introducing a prefix-tree (FP-tree) based algorithm without candidate set generation-and-test. This algorithm is called the pattern growth or FP-growth algorithm and needs two database scans.

In practice, considering the binary frequency (either absent or present) or support of a pattern may not be a sufficient indicator for finding meaningful patterns from a transaction database because it only reflects the number of transactions in the database which contain that pattern. In our real world scenarios, one user can buy multiple copies of items. We can consider an example in market basket data. For example, customer X has bought 3 pens, 4 pencils and 1 eraser, customer Y has bought 10 apples, customer Z has bought 3 breads and 5 milks and customer R has bought 2 shirts and 1 shoe. Therefore, traditional frequency/support measure cannot analyse the exact number of items (itemsets) purchased. For that reason, itemset share approach (Carter et al., 1997; Barber and Hamilton, 2000, 2001, 2003; Li et al., 2005a,b) has been proposed to discover more important knowledge in association rule mining (Agrawal et al., 1993; Agrawal and Srikant, 1994; Verma and Vyas, 2005; Wei et al., 2006). Share measure can provide useful knowledge about the numerical values that are typically associated with the transactions items. In addition to our real world retail market, it is also well applicable to find more useful web path traversal patterns because time spent in each website by the user is different. Other application areas, such as biological gene database, stock tickers, network traffic measurements, web-server logs, data feeds from sensor networks and telecom call records can have similar solutions.

Motivated from these real world scenarios, we propose one efficient approach to discover share-frequent patterns. The existing solutions (Carter et al., 1997; Barber and Hamilton, 2000, 2001, 2003; Li et al., 2005a,b) suffer in the level-wise candidate set generation-and-test problem and they need several database scans depending on the length of the candidate patterns. In this paper, we propose a novel tree structure ShrFP-Tree (Share-frequent pattern tree) for share-frequent pattern mining. By holding useful property and exploiting a pattern growth technique, ShrFP-Tree finds all the actual share-frequent patterns very efficiently. Moreover, it needs maximum three database scans for finding out complete set of the share-frequent patterns, i.e. its number of database scans is not dependent on the maximum length of candidate. Extensive performance analyses show that our approach outperforms the existing most efficient algorithms ShFSM (Li et al., 2005a) and DCG (Li et al., 2005b) in both dense and sparse datasets.

In summary, the main contributions of this paper

Copyright ©2008, Australian Computer Society, Inc. This paper appeared at the Seventh Australasian Data Mining Conference (AusDM 2008), Glenelg, Australia. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 87, John F. Roddick, Jiuyong Li, Peter Christen and Paul Kennedy, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

are (1) Devising a novel tree structure that can efficiently maintain the maximum share information of each item in a tree for finding all the candidate share-frequent patterns, (2) Applying a pattern growth mining approach on the above tree structure in order to eliminate the level-wise candidate generation-and-test methodology in share-frequent pattern mining, (3) Demonstration of how to achieve the complete share-frequent patterns by using the proposed approach and by scanning the database maximum three times, and (4) Extensive performance study to compare the performance of our algorithm with the existing most efficient algorithms ShFSM (Li et al., 2005a) and DCG (Li et al., 2005b).

The remainder of this paper is organized as follows. In Section 2, we describe related work and the main problems of the existing most efficient ShFSM and DCG algorithms for mining share-frequent patterns. In Section 3, we describe the share-frequent pattern mining problem. In Section 4, we describe the construction and mining process of our proposed ShrFP-Tree structure for share-frequent pattern mining. In Section 5, experimental results are presented and analysed. Finally, conclusions are presented in Section 6.

2 Related Work

2.1 Frequent Pattern Mining

Let $I = \{i_1, i_2, \dots, i_m\}$ be a set of items and D be a transaction database $\{T_1, T_2, \dots, T_n\}$ where each transaction $T_i \in D$ is a subset of I . The support/frequency of a pattern $X\{x_1, x_2, \dots, x_p\}$ is the number of transactions containing the pattern in the transaction database. The problem of frequent pattern mining is to find the complete set of patterns satisfying a minimum support in the transaction database. The *downward closure* property (Agrawal et al., 1993; Agrawal and Srikant, 1994) is used to prune the infrequent patterns. This property tells that if a pattern is infrequent then all of its super patterns must be infrequent. Apriori (Agrawal et al., 1993; Agrawal and Srikant, 1994) algorithm is the initial solution of frequent pattern mining problem and very useful in association rule mining. But it suffers from the candidate generation-and-test problem and needs several database scans.

FP-growth (Han et al., 2004) solved the problem of candidate generation-and-test by using a tree-based (FP-tree) solution without any candidate generation. It needs only two database scans to find all the frequent patterns. FP-array (Grahne and Zhu, 2005) technique was proposed to reduce the FP-tree traversals and it efficiently works especially in sparse datasets. One interesting measure *h-confidence* (Xiong et al., 2006) was proposed to identify the strong support affinity frequent patterns. Some other research (Wang et al., 2005; Han et al., 2007; Dong and Han, 2007; Liu et al., 2007) has been done for frequent pattern mining. Tree structures have been proposed to calculate all the frequent patterns using a single pass of database such as CanTree (Leung et al., 2007), CP-tree (Tanbeer et al., 2008), etc. This traditional frequent pattern mining problem considers only the binary occurrence (0/1), i.e. either absent or present of the items in one transaction.

2.2 Share-Frequent Pattern Mining

Carter et al. 1997 first introduced the share-confidence model to discover useful knowledge about numerical attributes associated with items in a transaction. ZP and ZSP (Barber and Hamilton, 2000,

2003) algorithms use heuristic methods to generate all the candidate patterns. Moreover, they cannot rely on the *downward closure* property and therefore their searching method is very time-consuming and does not work efficiently in large databases. Some other algorithms such as SIP, CAC and IAB (Barber and Hamilton, 2000, 2001, 2003) have been proposed to mine share-frequent patterns but they may not discover all the share-frequent patterns. Fast share measure (ShFSM), (Li et al., 2005a) improves the previous algorithms by using the *level closure* property. This property cannot maintain the *downward closure* property. ShFSM wastes the computation time on the join and the prune steps of candidate generation in each pass, and generates too many useless candidates.

Direct Candidates Generation (DCG) (Li et al., 2005b) algorithm outperforms ShFSM by generating candidates directly without the prune and the join steps in each pass. Moreover, the number of candidates generated by DCG is less than ShFSM. DCG can maintain the *downward closure* property by using the potential maximum *local measure value* (Definition 5) of an itemset which is actually the *transaction measure value* (Definition 6) of an itemset. Still, DCG has a big problem of level-wise candidate generation-and-test methodology. As a result, its number of database scans is dependent on maximum candidate length and it tests huge unnecessary candidate patterns. In the k -th pass, DCG scans the whole database to count the *local measure value* of each candidate k -itemset X and counts the potential maximum share value of each monotone $(k+1)$ -superset of X (Li et al., 2005b). Therefore, DCG generates and tests too many candidate patterns in the mining process. For example, if the number of distinct items is 10000, then it tests $\binom{10000}{2}$ two-element candidate patterns in pass-1 to get the actual candidate patterns of pass-2. Moreover, its number of database scans is dependent on the maximum length of the actual candidate patterns. For example, if the maximum length of a candidate pattern is 4 ("abcd"), DCG has to scan database 4 times to find all the share-frequent itemsets. If the maximum actual candidate length is 20, a total of 20 database scans are required. So, for maximum actual candidate length N , a total of N database scans are required. As a result, DCG is very inefficient for (1) dataset where the number of distinct items is large and (2) dense datasets where the maximum candidate pattern length is big.

In this paper, we propose a novel tree structure to remove these problems of the existing most efficient known algorithm DCG. Our approach generates a very few candidates using a pattern growth technique and its maximum number of database scans is three which is totally independent of the maximum length of the candidate patterns.

3 Problem Definition

We have adopted similar definitions presented in the previous works (Carter et al., 1997; Barber and Hamilton, 2000, 2001, 2003; Li et al., 2005a,b). Let $I = \{i_1, i_2, \dots, i_m\}$ be a set of items and D be a transaction database $\{T_1, T_2, \dots, T_n\}$ where each transaction $T_i \in D$ is a subset of I .

Definition 1: The *measure value* $mv(i_p, T_q)$, represents the quantity of item i_p in transaction T_q . For example, in Table 1, $mv(a, T_3) = 5$.

Definition 2: The *transaction measure value* of a transaction T_q denoted as $tmv(T_q)$ means the *total*

Table 1: Example of a transaction database with counting

TID	Transaction	Count	Total count
T_1	$\{a, b, f, g\}$	$\{2, 1, 2, 1\}$	6
T_2	$\{b, c, h\}$	$\{3, 2, 2\}$	7
T_3	$\{a, c, e\}$	$\{5, 3, 3\}$	11
T_4	$\{c\}$	$\{5\}$	5
T_5	$\{b, c, d\}$	$\{4, 3, 2\}$	9
T_6	$\{a, c, e, f, g\}$	$\{1, 3, 1, 2, 1\}$	8
T_7	$\{a, d\}$	$\{1, 3\}$	4
T_8	$\{a, c, e, f\}$	$\{4, 2, 1, 5\}$	12

measure value of a transaction T_q and it is defined by,

$$tmv(T_q) = \sum_{i_p \in T_q} mv(i_p, T_q) \quad (1)$$

For example, $tmv(T_7) = mv(a, T_7) + mv(d, T_7) = 1 + 3 = 4$ in Table1.

Definition 3: The total measure value $Tmv(DB)$ represents the *total measure value* in DB . It is defined by,

$$Tmv(DB) = \sum_{T_q \in DB} \sum_{i_p \in T_q} mv(i_p, T_q) \quad (2)$$

For example, $Tmv(DB) = 62$ in Table1.

Definition 4: The *itemset measure value* of an itemset X in transaction T_q , $imv(X, T_q)$ is defined by,

$$imv(X, T_q) = \sum_{i_p \in X} mv(i_p, T_q) \quad (3)$$

where $X = \{i_1, i_2, \dots, i_k\}$ is a k -itemset, $X \subseteq T_q$ and $1 \leq k \leq m$. For example, $imv(bc, T_2) = 3 + 2 = 5$ in Table1.

Definition 5: The *local measure value* of an itemset X is defined by,

$$lmv(X) = \sum_{T_q \in DB_X} \sum_{i_p \in X} imv(i_p, T_q) \quad (4)$$

where DB_X is the set of transactions contain itemset X . For example, $lmv(ac) = imv(ac, T_3) + imv(ac, T_6) + imv(ac, T_8) = 8 + 4 + 6 = 18$ in Table1.

Definition 6: The *transaction measure value* of an itemset X , denoted by $tmv(X)$, is the sum of the tmv values of all the transactions containing X .

$$tmv(X) = Tmv(DB_X) = \sum_{X \subseteq T_q \in DB_X} tmv(T_q) \quad (5)$$

For example, $tmv(g) = tmv(T_1) + tmv(T_6) = 6 + 8 = 14$ in Table 1.

Definition 7: The *share* value of an itemset X , denoted as $SH(X)$, is the ratio of the *local measure value* of X to the *total measure value* in DB . So, $SH(X)$ is defined by,

$$SH(X) = \frac{lmv(X)}{Tmv(DB)} \quad (6)$$

For example, $SH(ac) = 18/62 = 0.29$, in Table 1.

Definition 8: Given a minimum share threshold, $minShare$, an itemset is *share-frequent* if $SH(X) \geq minShare$. If $minShare$ is 0.25 (we can also express it as 25%), in the example database, “ac” is a share-frequent itemset, as $SH(ac) = 0.29$.

Definition 9: The *minimum local measure value*, min_lmv , is defined as

$$min_lmv = ceiling(minShare \times Tmv(DB)) \quad (7)$$

In Table 1, if $minShare = 0.25$, then $min_lmv = ceiling(0.25 \times 62) = ceiling(15.5) = 16$. So, for any itemset X , if $lmv(X) \geq min_lmv$, then we can say that X is a share-frequent pattern.

Main challenging problem of share-frequent pattern mining area is, itemset share does not have the *downward closure* property. For example, $SH(a) = 0.2096$ in Table 1, so “a” is a share-infrequent item in Table 1 for $minShare = 0.25$, but $SH(ac) = 0.29$, so “ac” is a share-frequent itemset. As a result, the *downward closure* property does not satisfy. Therefore, maintaining the *downward closure* property is very challenging here.

The *downward closure* property can be maintained by using the *transaction measure value* (Definition 6). For a pattern X , if $tmv(X) < min_lmv$, then we can prune that pattern without further consideration. For example, In Table 1 if we consider $minShare = 0.25$, then $tmv(g) = 14 < min_lmv(16)$. As a result, according to the *downward closure* property none of the super patterns of “g” can be a share-frequent pattern and therefore we can easily prune “g” at the early stage.

4 ShrFP-Tree: Our Proposed Tree Structure

4.1 Construction Process of ShrFP-Tree

In this section, we describe the construction process of the ShrFP-Tree (Share-frequent pattern tree) for share-frequent pattern mining. We maintain Header table and keep item name and tmv values of items in it. To facilitate the tree traversals adjacent links are also maintained (not shown in the figures for simplicity) in the ShrFP-Tree.

In the first database scan, ShrFP-Tree captures the tmv value of all the items. Consider the database shown in Table 1 and $minShare = 0.25$. According to equation 7, $min_lmv = 16$. After the first database scan, the tmv values of the individual items are $a : 41, b : 22, c : 52, d : 13, e : 31, f : 26, g : 14$ and $h : 7$. To be a candidate share-frequent item, the tmv of an item must be at least 16. Therefore, the items “d”, “g” and “h” are not candidate items. According to the *downward closure* property, we can prune these items without further consideration. Next, we sort

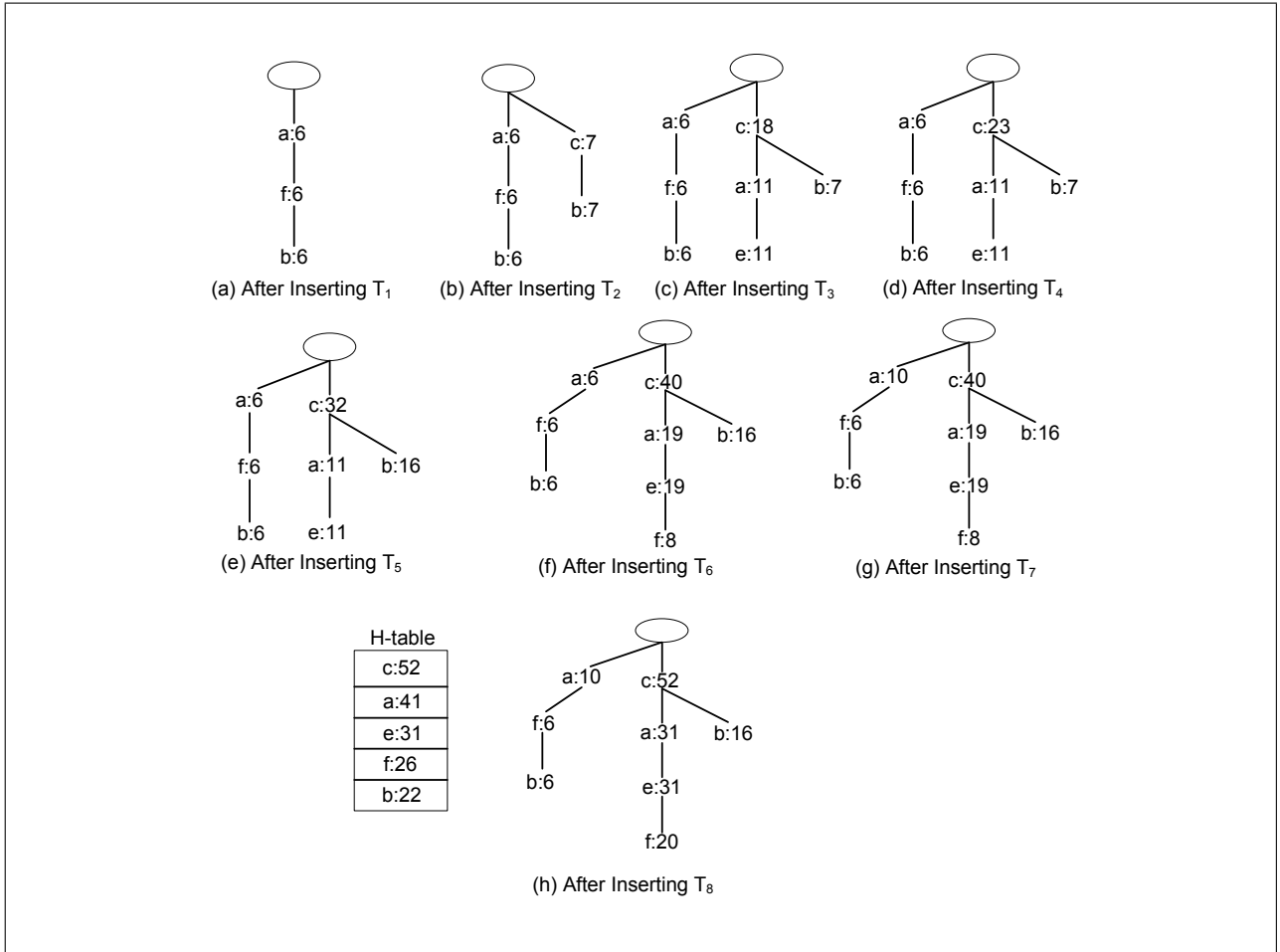


Figure 1: Construction process of ShrFP-Tree

the header table in descending order according to tmv values of the items. The header table order of items is $c : 52, a : 41, e : 31, f : 26$ and $b : 22 >$.

In the second database scan we consider only candidate items from each transaction and sort them according to the header table sort order, and then we insert them into the tree. For the first transaction T_1 , which contains item “a”, “b”, “f” and “g”, we discard the non-candidate item “g” at first and then arrange the items according to the header table sort order. Items “a”, “f” and “b” get the tmv value of T_1 , which is 6. Figure 1(a) shows the ShrFP-Tree after inserting T_1 . In T_2 , non-candidate item “h” is removed and the remaining items of T_2 are arranged (at first “c” then “b”) in the descending tmv order presented in the header table. After that, items “c” and “b” are inserted in the ShrFP-Tree as a new path with tmv value 7, shown in Figure 1(b). In T_3 , all items are candidate items. They are arranged in the same way and the order is “c”, “a”, “e”. Item “c” gets the prefix sharing with the existing node containing item “c”. The tmv value of “c” becomes $7+11=18$, item “a” becomes its child with tmv value 11 and item “e” becomes the child of “a” with same tmv value 11 (shown in Figure 1(c)). Figure 1(d) to Figure 1(h) show the insertion process of transactions T_4 to T_8 . Figure 1(h) shows the final tree with the header table for the full database presented in Table 1. In the next section we will perform the mining operation in this tree presented at Figure 1(h).

Property 1: The total count of tmv value of any node in ShrFP-Tree is greater than or equal to the sum of total counts of tmv values of its children.

4.2 Mining Process of ShrFP-Tree

ShrFP-Tree exploits a pattern growth mining approach to mine all the candidate share-frequent patterns. As our tree-structure has the important property of FP-tree stated in property 1, pattern growth mining algorithm can be directly applicable to it by using the tmv value.

Consider the database of Table 1 and $minShare = 0.25$ in that database. The final ShrFP-Tree is created for that database is shown in Figure 1(h). At first the conditional tree of the bottom most item “b” (shown in Figure 2(a)) is created by taking all the branches prefixing the item “b” and deleting the nodes containing an item which cannot be a candidate pattern with the item “b”. Obviously, items “f” and “a” cannot be candidate patterns with item “b” as they have low tmv values with it. Both the items “f” and “a” has tmv value 6 with the item “b” and minimum tmv value must be 16 to be a candidate pattern. So, the conditional tree of item “b” does not contain the items “f” and “a”. Therefore, candidate patterns (1) $\{b, c\}$ and (2) $\{b\}$ are generated here.

In the similar fashion, conditional tree for item “f” is created in Figure 2(b) and candidate patterns (3) $\{e, f\}$, (4) $\{a, f\}$, (5) $\{c, f\}$ and (6) $\{f\}$ are generated. The conditional tree of itemset “fe” is shown in Figure 2(c) and candidate patterns (7) $\{a, e, f\}$, (8) $\{c, e, f\}$ and (9) $\{a, c, e, f\}$ are generated. The conditional tree of itemset “fa” is shown in Figure 2(d) and candidate pattern (10) $\{a, c, f\}$ is generated. The conditional tree of item “e” is shown in Figure 2(e) and candidate patterns (11) $\{a, e\}$, (12) $\{c, e\}$, (13) $\{a, c, e\}$ and (14) $\{e\}$ are generated. The conditional tree of item “a” is shown in Figure 2(f) and candidate patterns (15)

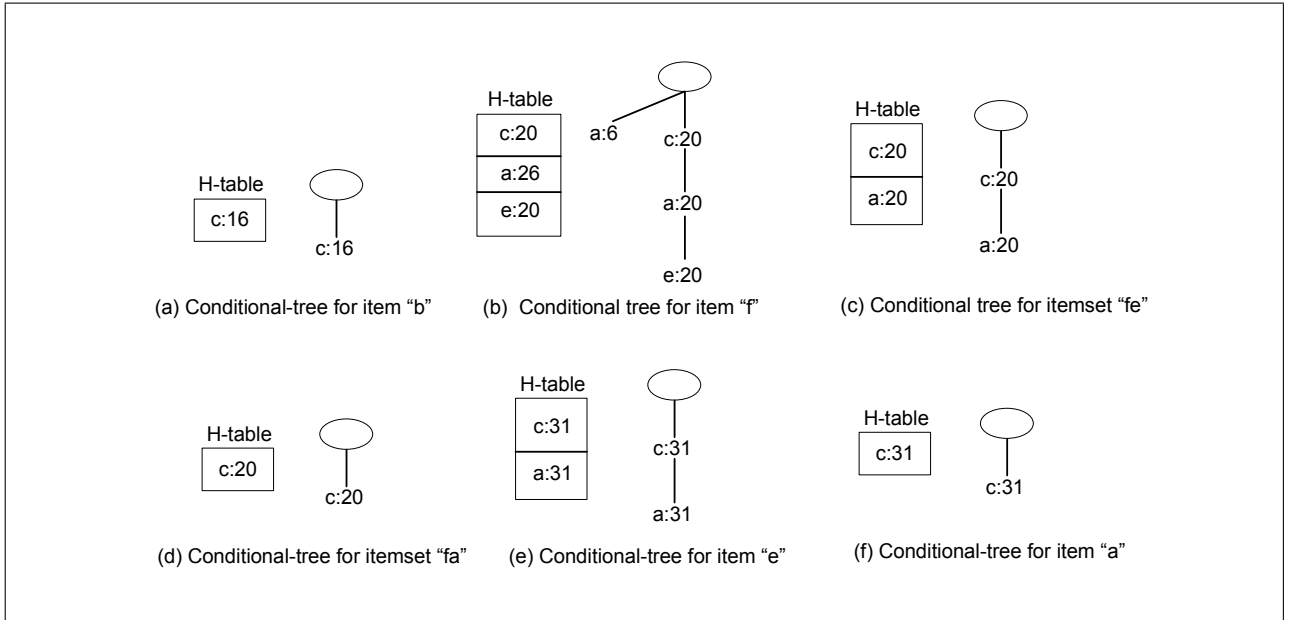


Figure 2: Mining process of ShrFP-Tree

Table 2: Calculation process of share-frequent patterns

No.	Candidate patterns	tmv	lmv	SH	Share-frequent patterns
1.	<i>bc</i>	16	12	0.1935	No
2.	<i>b</i>	22	8	0.129	No
3.	<i>ef</i>	20	9	0.145	No
4.	<i>af</i>	26	16	0.258	Yes
5.	<i>cf</i>	20	12	0.1935	No
6.	<i>f</i>	26	9	0.145	No
7.	<i>aef</i>	20	14	0.2258	No
8.	<i>cef</i>	20	14	0.2258	No
9.	<i>acef</i>	20	19	0.306	Yes
10.	<i>acf</i>	20	17	0.274	Yes
11.	<i>ae</i>	31	15	0.2419	No
12.	<i>ce</i>	31	13	0.2096	No
13.	<i>ace</i>	31	23	0.37	Yes
14.	<i>e</i>	31	5	0.08	No
15.	<i>ac</i>	31	18	0.29	Yes
16.	<i>a</i>	41	13	0.2096	No
17.	<i>c</i>	52	18	0.29	Yes

$\{a, c\}$ and (16) $\{a\}$ are generated. The last candidate pattern (17) $\{c\}$ is generated for the top-most item "c". Our approach performs the third database scan to find share-frequent patterns from these 17 candidate patterns. Table 2 shows the calculation process of the actual share-frequent patterns from the candidate patterns. The resultant share-frequent patterns are, $\{a, f\}$, $\{a, c, e, f\}$, $\{a, c, f\}$, $\{a, c, e\}$, $\{a, c\}$ and $\{c\}$.

5 Experimental Results

In this section, we present our experimental results on the performance of our proposed approach in comparison with the most efficient share-frequent pattern mining algorithms, DCG (Li et al., 2005b) and ShFSM (Li et al., 2005a). The main purpose of this experiment is to show how efficiently and effectively the share-frequent patterns can be discovered in both dense and sparse datasets by our approach compared to the existing algorithms.

To evaluate the performance of our proposed tree structure, we have performed several experiments on IBM synthetic dataset *T10I4D100K* and real life datasets *mushroom* and *kosarak* from frequent itemset mining dataset repository (<http://fimi.cs.helsinki.fi/data/>) and UCI Machine Learning Repository (<http://kdd.ics.uci.edu/>). These datasets provides binary quantity of each item for each transaction. As like the performance evaluation of the previous share-frequent pattern mining (Li et al., 2005a,b) we have generated random numbers for the quantity of each item in each transaction, ranging from 1 to 10. Our programs were written in Microsoft Visual C++ 6.0 and run with the Windows XP operating system on a Pentium dual core 2.13 GHz CPU with 2GB main memory.

Dense datasets (Sucahyo et al., 2003) have too many long frequent as well as share-frequent patterns. The probability of an item's occurrence is very high in every transaction. As a result, for comparatively higher threshold, dense datasets have too many can-

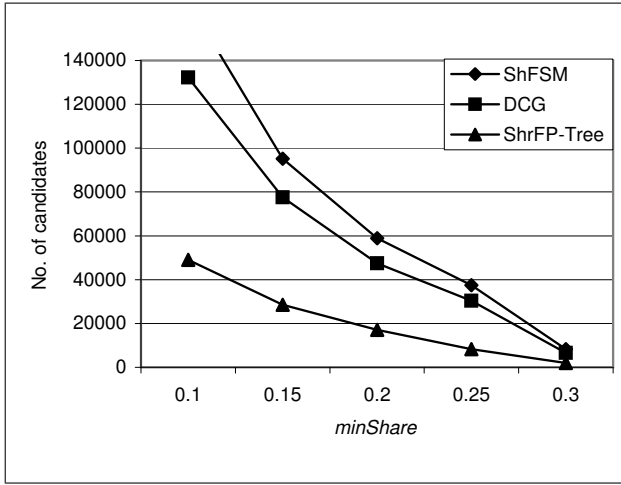


Figure 3: No. of candidates comparison on the *mushroom* dataset

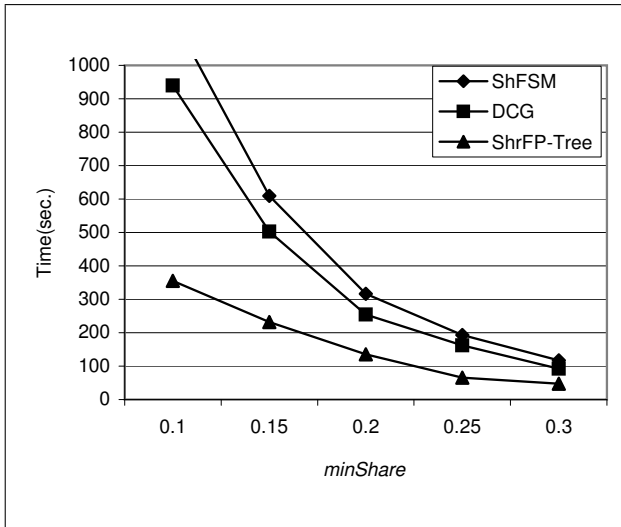


Figure 4: Execution time comparison on the *mushroom* dataset

didate patterns. Actually, long patterns need several database scans. That's why, when the dataset becomes denser, or minimum threshold becomes low, the number of candidates and total running time sharply increases in the Apriori based existing algorithms.

The *mushroom* dataset contains 8,124 transactions and 119 distinct items. Its mean transaction size is 23, around 20% $((23/119) \times 100)$ of its distinct items are present in every transaction and therefore it is a dense dataset. At first we compare the number of candidate patterns that have been tested by each algorithm. Figure 3 shows the number of candidate patterns comparison. The numbers of candidates of the existing algorithms rapidly increase below *minShare* = 0.2 (i.e. 20%). For *minShare* 0.1 and 0.15, the amounts of their candidate patterns are remarkable larger from our candidate patterns.

Figure 4 shows the running time comparison in the *mushroom* dataset. In the case of existing algorithms, for lower thresholds they have too many long candidate patterns and several database scans are needed for the huge number of long candidate patterns. As a result, time difference between existing algorithms and our algorithm becomes larger when the *minShare* decreases. So, these results demonstrate that existing algorithms are very inefficient for dense dataset when the *minShare* is low.

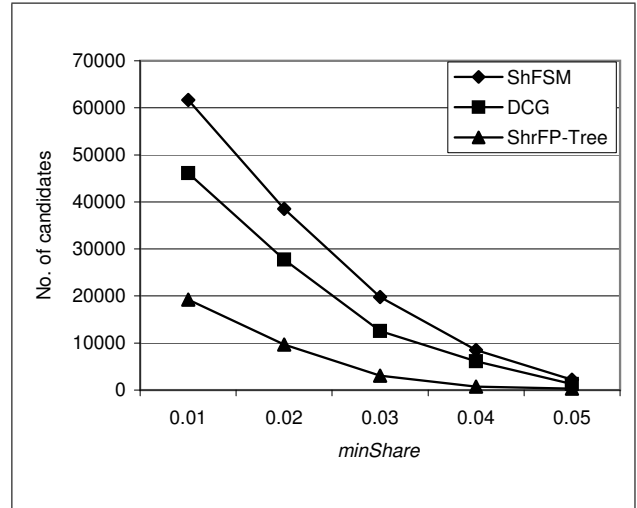


Figure 5: No. of candidates comparison on the *T10I4D100K* dataset

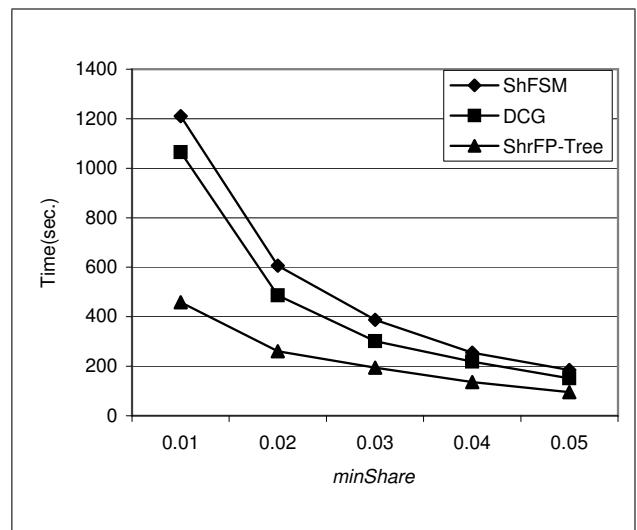


Figure 6: Execution time comparison on the *T10I4D100K* dataset

Sparse datasets (Ye et al., 2005; Grahne and Zhu, 2005) normally have too many distinct items. Although in the average case their transactions length is small, but they normally have many transactions. As we described in Section 2.2, handling too many distinct items is a severe problem in Apriori-like existing algorithms. We show here that handling large number of distinct datasets and several database scans over long sparse datasets also make the existing algorithms inefficient in sparse datasets.

The *T10I4D100K* dataset contains 100,000 transactions and 870 distinct items. Its mean transaction size is 10.1, around 1.16% $((10.1/870) \times 100)$ of its distinct items are present in every transaction and therefore it is a sparse dataset. The performance of our algorithm is better than the existing algorithms in both number of candidates and execution time comparisons, shown in Figure 5 and Figure 6 respectively. Obviously, the difference of candidate patterns and running time of existing algorithms and our approach becomes larger when the *minShare* becomes low.

The dataset *kosarak* contains click-stream data of a Hungarian on-line news portal. It contains 990,002 transactions and 41,270 distinct items. Its mean transaction size is 8.1, and it is a large sparse dataset. Around 0.0196% $((8.1/41270) \times 100)$ of its distinct items are present in every transaction. As we

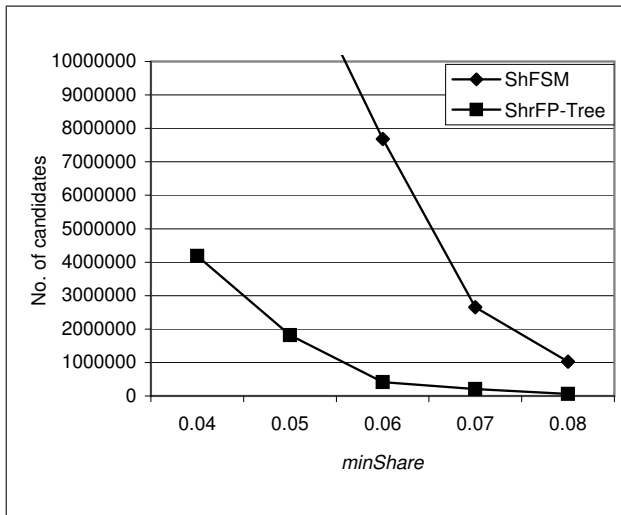


Figure 7: No. of candidates comparison on the *kosarak* dataset

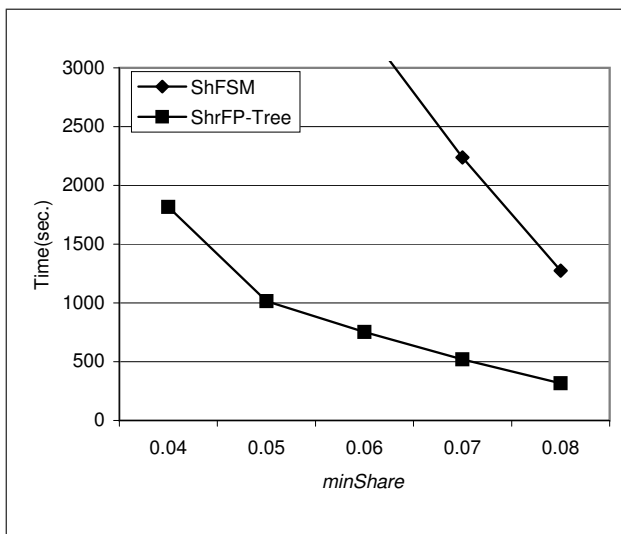


Figure 8: Execution time comparison on the *kosarak* dataset

described in Section 2.2, existing algorithms generate-and-test too many candidate patterns for this large number of distinct items shown in Figure 7. Obviously, too much time is needed for handling these candidates and scanning the long *kosarak* dataset with them. Running time comparison is shown in Figure 8. We compared the performance of our approach with the existing ShFSM algorithm. Since DCG maintains an extra array for each candidate (Li et al., 2005b), we could not keep all its candidates in each pass in the main memory. Figure 8 shows that our approach outperforms the existing algorithm on the *kosarak* dataset. Therefore, the existing algorithms are very inefficient for sparse datasets having too many distinct items and number of transactions.

Fig. 8 also shows that ShrFP-Tree has efficiently handled the 41,270 distinct items and around 1 million transactions in the *kosarak* dataset. Therefore, these experimental results demonstrate the scalability of our tree structure to handle a large number of distinct items and transactions.

6 Conclusions

The main contribution of this paper is to provide a very efficient research work for share-frequent pat-

tern mining in the area of data mining and knowledge discovery. To solve the level-wise candidate set generation-and-test problem of the existing algorithms, we propose a novel tree structure ShrFP-Tree. Our technique prunes huge number of unnecessary candidates during tree creation time by eliminating non-candidate single-element patterns and also during mining time by using a pattern growth approach. Its maximum number of database scans is totally independent of the maximum length of candidate patterns. It needs maximum three database scans in contrast to several database scans needed for the existing algorithms. Moreover, ShrFP-Tree is very simple, easy to construct and handle. Extensive performance analyses show that our approach is very efficient and it outperforms the existing most efficient algorithms in both dense and sparse datasets.

Acknowledgement

This study was supported by a grant of the Korea Health 21 R&D Project, Ministry for Health Welfare and Family Affairs, Republic of Korea (A020602).

References

- Agrawal, R., Imielinski, T., and Swami, A. (1993), Mining association rules between sets of items in large databases, in 'Proceedings of the 12th ACM SIGMOD International Conference on Management of Data', pp. 207-216.
- Agrawal, R. and Srikant, R. (1994), Fast Algorithms for Mining Association Rules in Large Databases, in 'Proceedings of the 20th International Conference on Very Large Data Bases', pp. 487-499.
- Barber, B. and Hamilton, H.J. (2000), Algorithms for mining share frequent itemsets containing infrequent subsets, in D.A. Zighed, H.J. Komorowski, J.M. Zytkow (Eds.), '4th European Conf. on Principles of Data Mining and Knowledge Discovery (PKDD 2000)', Lecture Notes in Computer Science, Vol. 1910, Springer-Verlag, Berlin, pp. 316-324.
- Barber, B. and Hamilton, H.J. (2001), 'Parametric algorithm for mining share frequent itemsets', *Journal of Intelligent Information Systems*, Vol. 16, pp. 277-293.
- Barber, B. and Hamilton, H.J. (2003), 'Extracting share frequent itemsets with infrequent subsets', *Data Mining and Knowledge Discovery*, Vol. 7, pp. 153-185.
- Carter, C.L., Hamilton, H.J., and Cercone, N. (1997), Share based measures for itemsets, in H.J. Komorowski, J.M. Zytkow (Eds.), '1st European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD 1997)', Lecture Notes in Computer Science, Vol. 1263, Springer-Verlag, Berlin, pp. 14-24.
- Dong, J. and Han, M. (2007), 'BitTableFI: An efficient mining frequent itemsets algorithm', *Knowledge-Based Systems*, Vol. 20, pp. 329-335.
- Grahne, G. and Zhu, J. (2005), 'Fast Algorithms for frequent itemset mining using FP-Trees', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, no. 10, pp. 1347-1362.
- Han, J., Cheng, H., Xin, D. and Yan, X. (2007), 'Frequent pattern mining: current status and future directions', *Data Mining and Knowledge Discovery*, Vol. 15, pp. 55-86.

- Han, J., Pei, J., Yin, Y., and Mao, R., (2004) 'Mining frequent patterns without candidate generation: a frequent-pattern tree approach', *Data Mining and Knowledge Discovery*, Vol. 8, pp. 53-87.
- Leung, C. K.-S., Khan, Q.I., Li, Z. and Hoque, T. (2007), 'CanTree: a canonical-order tree for incremental frequent-pattern mining', *Knowledge and Information Systems*, Vol. 11, no. 3, pp. 287-311.
- Li, Y.-C., Yeh, J.-S. and Chang, C.-C. (2005a), A fast algorithm for mining share-frequent itemsets, in 'Proceedings of the 7th Asia-Pacific Web Conference on Web Technologies Research and Development (APWeb)', Lecture Notes in Computer Science, Vol. 3399, Springer-Verlag, Berlin, pp. 417-428.
- Li, Y.-C., Yeh, J.-S. and Chang, C.-C. (2005b), Direct candidates generation: a novel algorithm for discovering complete share-frequent itemsets, in 'Proceedings of the 2nd Intl. Conf. on Fuzzy Systems and Knowledge Discovery (FSKD)', Lecture Notes in Artificial Intelligence, Vol. 3614, Springer-Verlag, Berlin, pp. 551-560.
- Liu, G., Tsai, Lu, H. and Yu, J. X. (2007), 'CFP-tree: A compact disk-based structure for storing and querying frequent itemsets', *Information Systems*, Vol. 32, pp. 295-319.
- Suchahyo, Y. G., Gopalan, R. P. and Rudra, A. (2003), 'Efficient mining frequent patterns from Dense Datasets Using a Cluster of Computers', AI 2003: Advances in Artificial Intelligence, LNAI 2903, pp. 233-244.
- Tanbeer, S. K., Ahmed, C. F., Jeong, B. -S. and Lee, Y. -K. (2008), CP-tree: A tree structure for single pass frequent pattern mining, in 'Proceedings of the 12th Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD)', pp. 1022-1027.
- Verma, K. and Vyas, O.P. (2005), 'Efficient calendar based temporal association rule', *SIGMOD Record*, Vol. 34, no. 3, pp. 63-70.
- Wang, J., Han, J., Lu, Y. and Tzvetkov, P. (2005), 'TFP: an efficient algorithm for mining top-k frequent closed itemsets', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, pp. 652-664.
- Wei, J. -M., Yi, W. -G. and Wang, M. -Y. (2006), 'Novel measurement for mining effective association rules', *Knowledge-Based Systems*, Vol. 19, pp. 739-743.
- Xiong, H., Tan, P.-N. and Kumar, V. (2006), 'Hyperclique Pattern Discovery', *Data Mining and Knowledge Discovery*, Vol. 13, pp. 219-242.
- Ye, F. -Y., Wang, J. -D. and Shao, B. -L. (2005), New algorithm for mining frequent itemsets in sparse database, in 'Proceedings of the 4th International Conference on Machine learning and Cybernetics', pp. 1554-1558.

Rare Association Rule Mining via Transaction Clustering

Yun Sing Koh¹

Russel Pears²

School of Computing Science and Mathematics
Auckland University of Technology, New Zealand,
Email: ykoh@aut.ac.nz¹, rpears@aut.ac.nz²

Abstract

Rare association rule mining has received a great deal of attention in the recent past. In this research, we use transaction clustering as a pre-processing mechanism to generate rare association rules. The basic concept underlying transaction clustering stems from the concept of large items as defined by traditional association rule mining algorithms. We make use of an approach proposed by Koh & Pears (2008) to cluster transactions prior to mining for association rules. We show that pre-processing the dataset by clustering will enable each cluster to express their own associations without interference or contamination from other sub groupings that have different patterns of relationships. Our results show that the rare rules produced by each cluster are more informative than rules found from direct association rule mining on the unpartitioned dataset.

Keywords: Rare Association Rule Mining, Transaction Clustering, Apriori-Inverse

1 Introduction

The main goal of association rule mining is to discover relationships among sets of items in a transactional database. Association rule mining was introduced by Agrawal et al. (1993). It aims to extract interesting correlations, frequent patterns, associations or casual structures among sets of items in transaction databases or other data repositories. The relationships are not based on inherent properties of the data themselves but rather based on the co-occurrence of the items within the database. The associations between items are also known as association rules. In the classical association rule mining process, all frequent itemsets are found, where an itemset is said to be frequent if it appears with minimum frequency s , called minimum support. Association rules are then derived from frequent items and are represented in the form $A \rightarrow B$ where AB is a frequent itemset. Strong association rules are those that meet the minimum confidence c threshold (the percentage of transactions containing A that also contain B).

A much less explored area in association mining is infrequent itemset mining or rare association rule mining. Items that rarely occur are in very few transactions and are normally pruned out. One limitation of common association rule mining approaches, i.e. Apriori, are that they rely on there being a

meaningful minimum support level that is reasonable (sufficiently strong) to reduce the number of frequent itemsets generated to a manageable level. However, in some data mining applications relatively infrequent associations are likely to be of great interest as they relate to rare but crucial cases. Examples of mining rare itemsets include identifying relatively rare diseases, predicting telecommunication equipment failure, and finding associations between infrequently purchased supermarket items. Indeed, infrequent itemsets warrant special attention because they are more difficult to find using traditional data mining techniques.

In this paper we first pre-process the dataset by clustering transactions before performing association rule mining. The rationale behind clustering transactions prior to mining association rules is that the latter is performed on partitions that are essentially distinct from each other. Each cluster would be expected to contain associations without interference or contamination from other sub groupings that have different patterns of relationships. We thus adopt a two phased approach. The first phase comprises the transaction clustering phase and adopts the clustering method proposed by Koh & Pears (2008). In the second phase we generate rare rules based on the clusters generated in the initial phase.

The basic concept underlying transaction clustering stems from the concept of large items as defined by association rule mining algorithms. Currently, none of the techniques proposed offer a good solution to scenarios where large items overlap across clusters. A further limitation with some of the existing algorithms is that they rely on some form of domain specific knowledge, thus limiting their range of applicability. Koh & Pears (2008) overcome the aforementioned limitations by using cluster seeds that represent initial centroids. Seeds are generated from sets of transaction items that occur together above a certain threshold and such seeds may overlap in their itemsets across clusters.

In the second phase we run Apriori-Inverse (Koh & Rountree 2005) on the clusters generated. In this approach we consider itemsets that are above a minimum absolute support requirement (Koh et al. 2008) and below a maximum support threshold. We show that we find more informative rules compared with Apriori-Inverse on the unclustered dataset.

The remainder of this paper is organised as follows. Section 2 provides a review of related research. In Section 3 we introduce the notion of transaction clustering by seeding. Section 4 describes how rare association rule mining is applied on the clusters produced with transaction clustering. Experimental results of applying the method on several real-world datasets is presented in Section 5. The paper concludes in Section 6 with a summary of the contributions made in this research.

Copyright ©2008, Australian Computer Society, Inc. This paper appeared at the Seventh Australasian Data Mining Conference (AusDM 2008), Glenelg, Australia. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 87, John F. Roddick, Jiuyong Li, Peter Christen and Paul Kennedy, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

2 Related Work

In this section we look at three related areas, clustering transaction, association rule mining, and the combination of the two.

2.1 Clustering Transactions

In the recent past there has been an increasing level of interest in transaction clustering. All such approaches have employed quite different methods when compared to traditional clustering methods. Wang et al. (1999) utilised the concept of large items (Agrawal et al. 1993) to cluster transactions. Their approach measures the similarity of a cluster based on the large items in the transaction dataset. Each transaction is either allocated to an existing cluster or assigned to a new cluster based on a cost function. The cost function measures the degree of similarity between a transaction and a cluster based on the number of large and small items shared between that transaction and the given cluster.

To speed-up the method proposed above, Yun et al. (2001) introduced a method called SLR (Small-Large Ratio). Their method essentially uses the measurement of the ratio between small to large items to cluster transactions. Both the large item (Wang et al. 1999) and SLR (Yun et al. 2001) method suffers a common drawback. In some cases, they may fail to give a good representation of the clusters. Suppose that A and B are large items in a transaction dataset, with A and B occurring individually 60% of the time and AB occurring together 40% of the time. If the support threshold is set at 40%, then the cost function that they use results in two clusters, one having transactions that contain the item A and the other that contains item B. However, in this case, it is clear that the optimal cluster configuration requires an additional cluster containing the itemset AB. Their approach thus tends to discourage bonding between itemsets already occurring in other clusters. This in turn forces transactions to choose between sub-optimal clusters when deciding what cluster that they should belong to.

Xu et al. (2003) proposed a method using the concept of a caucus. The basic idea of introducing a caucus to cluster transactions is motivated by the fact that cluster quality is sensitive to the initial choice of cluster centroids (Xu et al. 2003). Fundamentally different from most other clustering algorithms, their approach attempts to group customers with similar behaviour. In their approach they first determine a set of background attributes from the dataset that are significant. A set of caucuses, consisting of different subsets of items is then constructed to identify the initial cluster centroids.

The main drawback of this method is that it requires the user to define the initial centroids which is difficult as it requires some form of prior knowledge about the dataset. The cluster seeding method Koh & Pears (2008) overcomes the two main issues with the current approaches in that it copes well with overlapping centroids and does not require background domain specific knowledge.

2.2 Rare Association Rule Mining

Detecting sporadic association rules, rules with low support but high confidence efficiently is a difficult data mining problem. To find these rules in traditional approaches, such as the Apriori algorithm, minimum support (minsup) has to be set very low, which results in a large amount of redundant rules. As a specific example of the problem, consider the

association mining problem where we want to determine if there is an association between buying a food processor and buying a cooking pan (Liu et al. 1999). The problem is that both items are rarely purchased in a supermarket. Thus, even if the two items are almost always purchased together when either one is purchased, this association may not be found. Modifying the minsup threshold to take into account the importance of the items is one way to ensure that rare items remain in consideration. To find this association minsup must be set low. However setting this threshold low would cause a combinatorial explosion in the number of itemsets generated. Frequently occurring items will be associated with one another in an enormous number of ways simply because the items are so common that they cannot help but appear together. This is known as the rare item problem (Liu et al. 1999). It means that using the Apriori algorithm, we are unlikely to generate rules that may indicate rare events of potentially dramatic consequence.

Liu et al. (1999) note that some individual items can have such low support that they cannot contribute to rules generated by Apriori, even though they may participate in rules that have very high confidence. They overcome this problem with a technique called MSAPriori whereby each item in the database can have a minimum item support (MIS) given by the user. By providing a different MIS for different items, a higher minimum support is tolerated for rules that involve frequent items and a lower minimum support for rules that involve less frequent items. Yun et al. (2003) proposed the RSAA algorithm to generate rules in which significant rare itemsets take part, without any "magic numbers" specified by the user. This technique uses relative support: RSup is used in place of support. Thus, this algorithm decreases the support threshold for items that have low frequency and increases the support threshold for items that have high frequency.

Koh et al. (2008) proposed an approach to find rare rules with candidate itemsets that fall below a maxsup (maximum support) level but above a minimum absolute support value. They introduced an algorithm called Apriori-Inverse to find sporadic rules efficiently: for example, a rare association of two common symptoms indicating a rare disease. They later proposed another approach called MIISR. In their approach, the consequent of these rules is an item below maxsup threshold and the antecedent has support below maxsup but may consist of individual items above maxsup. In both approaches they use minimum absolute support (minabssup) threshold value derived from an inverted Fisher's exact test to prune out noise. At the low levels of co-occurrences of candidate itemsets that need to be evaluated to generate rare rules, there is a possibility that such co-occurrences happen purely by chance and are not statistically significant. The Fisher test provided a statistically rigorous method of evaluating significance of co-occurrences and was thus an integral part of their approach.

Like Apriori and MSAPriori, RSAA is exhaustive in its generation of rules, so it spends time looking for rules which are not sporadic (i.e. rules with high support and high confidence). If the minimum-allowable relative support value is set close to zero, RSAA takes a similar amount of time to that taken by Apriori to generate low-support rules in amongst the high-support rules.

2.3 Combining Clustering and Association Rule Mining

Recently, Plasse et al. (2007) proposed a method of analysing links between binary attributes in a large sparse data set. Initially the variables are clustered to obtain homogeneous clusters of attributes. Association rules are then mined in each cluster. Plasse et al. (2007) used several clustering methods and compared the resulting partitions. They generated their clusters based on hierarchical methods which are divided into two groups: ascendant methods based on an agglomerative algorithm and descendant methods performed by a divisive algorithm. The similarity coefficients used in their clustering technique includes Russel and Rao, Jaccard, Ochiai, and Dice. Once the clusters have been generated by the different techniques, association rules were produced on the different clusters. While their method did succeed in finding association rules that could not be discovered without clustering, the inherent weakness was in the clustering algorithms that they employed. None of the methods proposed offered a good solution to scenarios where large items overlap across clusters.

A further limitation with some of the existing transaction clustering algorithms is that they rely on some form of domain specific knowledge, thus limiting their range of applicability. Executing numerous different clustering methods and then generating rules based on each of the clusters produced becomes prohibitively expensive in certain cases. This is especially true when clustering is employed over a variety of datasets from different domains.

In the next section we examine in detail the transaction clustering approach that we adopt. We show that the process of transaction clustering is fundamentally different from that of traditional clustering and discuss the specific concepts and methods that are required to generate high quality clusters containing a high degree of homogeneity of transactions.

The clustering algorithm that we describe achieves a much higher degree of homogeneity of items within a cluster, with either all or a large percentage of its items falling into the frequent category. Furthermore, the degree of heterogeneity across clusters was also significantly greater than with the Large Item approach, with very few frequent items being duplicated across clusters (Koh & Pears 2008). Results on a range of real world datasets showed that it significantly outperformed its Large Item counterpart in both respects. We believe that cluster quality is crucial in discovering association rules that would otherwise be undetectable via mining on an unclustered dataset, and this motivated us to choose the approach proposed by (Koh & Pears 2008).

3 Transaction Clustering By Seeding

Clustering is the process of finding naturally occurring groups in data. Clustering is one of the most widely studied techniques in the context of data mining and has many applications, including disease classification, image processing, pattern recognition, and document retrieval. Traditional clustering techniques deal with horizontal segmentation of data, whereby clusters are formed from sets of non-overlapping instances. Many efficient algorithms exist for the traditional clustering problem (Jain et al. 1999, Ganti et al. 1999, Guha et al. 2000). In contrast, transaction clustering has fundamentally different requirements, and has been gaining increasing attention in recent years. Unlike traditional clustering, transaction clustering requires that transactions be partitioned across clusters in such a manner that instances within a cluster

share a common set of large items, where the concept of large follows the same meaning attributed to frequent items in association rule mining (Agrawal et al. 1993). Thus it is clear that transaction clustering requires a fundamentally different approach from the traditional clustering techniques. Compounding the level of difficulty is the fact that transaction data is known to have high dimensionality, sparsity, and a potentially large number of outliers (Xu et al. 2003).

Current research in both data mining and information retrieval suggests that transaction clustering functionality needs to extend well beyond a near neighbourhood search for similar instances (Wang et al. 1999, Cutting et al. 1992). This form of clustering provides a natural solution to many applications such as targeted marketing/advertising, discovering causes of diseases, and others.

In this paper we adopt a recent approach used by (Koh & Pears 2008) for transaction clustering that is based on an initial seeding of cluster centroids. Their approach consists of two phases: a seed generation phase followed by a transaction allocation phase. In the seed generation phase the seeds are identified in a progressive manner by a candidate generation process based on Apriori (Agrawal et al. 1993). Large items are extended in precisely the same manner as Apriori. The chi square significance test is used to ensure that only strongly associated items are joined together into an itemset. The improvement constraint restricts the growth of a seed to ensure that it only consists of items that increase the value of an improvement function. Once seeds are generated, the next phase assigns transactions to clusters. Each transaction is allocated to a cluster centroid with the highest similarity. Once all transactions have been allocated, the centroid is recalculated for each cluster. The new centroid consists of large items that reside in the cluster. Transactions are then reallocated to clusters on the basis of proximity to the new centroids that were defined. In order to determine the optimal value for the number of clusters, the allocation phase is repeated until the value of a fitness function reaches a plateau.

Let $D = \{t_1, \dots, t_n\}$ be a set of transactions. Each transaction is a set of items $\{i_1, \dots, i_m\}$. C is a partition of the transaction, $\{C_1, \dots, C_k\}$ of $\{t_1, \dots, t_n\}$. Each C_i is called a cluster. Overall the clustering is divided into two main phases: seed generation and allocation phases.

3.1 Seed Generation Phase

We start by describing the method used for finding the optimal number of clusters. The initial choice of seeds are the large items in the dataset. A minimum support threshold, θ is used to identify large items, where $0 < \theta < 1$. Any item in the dataset that has support above $|D| * \theta$ is considered a large item. Let L_i denote the set of large items or large itemsets. The items L_i are extended to itemsets L_{i+1} in the same way as Apriori generates candidate frequent itemsets. For a large itemset to be considered a cluster seed the frequency of co-occurrence of all pairs of subsets within the seed must occur together with a frequency above a threshold value at a given significance level. This effectively ensures that cluster seeds of size ≥ 2 have items that co-occur together at a frequency that is statistically significant. In addition, all cluster seeds satisfy an improvement constraint when they are extended. This constraint is based on the concept of relative support.

Definition 1 (Relative Support). The relative support of an itemset X_k of size k is defined to be the ratio of the support of X_k to the support of Y_{k-1}

which is that $(k-1)$ -sized subset of X_k with the maximum support. Thus,

$$RS(X_k) = \frac{\text{supp}(X_k)}{\text{supp}(Y_{k-1})}$$

Definition 2 (Extension of a Seed). Given two existing seeds, X_{k-1} and Y_{k-1} , X_{k-1} is extended to a new seed $X_{k-1} \cup Y_{k-1}$ if and only if:

$$\phi(X_{k-1}, Y_{k-1}) > \chi_c^2,$$

$$RS(X_{k-1} \cup Y_{k-1}) - RS(X_{k-1}) > \sigma, \text{ and}$$

$$RS(X_{k-1} \cup Y_{k-1}) - RS(Y_{k-1}) > \sigma$$

where ϕ denotes the chi square correlation coefficient, χ_c^2 , the chi square cut-off threshold at the $c\%$ confidence level and σ is a user-defined threshold.

The rationale behind extension lies in the fact that the new itemset to be added to the seed has a statistically strong correlation with the existing seed and that the inclusion of the new itemset will improve the relative support of the seed above a user defined minimum threshold. The algorithm for the seed clustering phase is shown below in Figure 1.

Algorithm for Seed Generation Phase

Input: Transaction database D , θ value, σ value, universe of items I
Output: Cluster Seeds, $S = \{s_1 \dots s_k\}$
 $k \leftarrow 1$
 $s_k \leftarrow \{\{i\} | i \in I, \text{count}(\{i\}) \geq |D| * \theta\}$
while $l_k \neq \emptyset$ do
 $k \leftarrow k + 1$
 $l_k \leftarrow \{x \cup y | x, y \in s_{k-1}, |x \cap y| = k - 2\}$
 $s_k \leftarrow \{x \cup y | x \cup y \in l_k, \phi(x, y) \geq \chi_c^2, RS(x \cup y) - RS(y) > \sigma, RS(x \cup y) - RS(x) > \sigma\}$
end while
return $\bigcup_{t=1}^{k-1} s_t$

Figure 1: Algorithm for Seed Generation Phase

3.2 Allocation Phase

The seeds produced in the initial phase are considered as the initial centroids for the clusters. In this phase, transactions are assigned to clusters on the basis of similarity to cluster centroids. In order to measure similarity we modified the Jaccard similarity coefficient. For each transaction, t , we calculate the similarity between t and the existing centroid, c_k . The similarity, sim , is between t and the c_k is calculated as:

$$\text{sim}(t, c_k) = \frac{|t \cap c_k|}{|t \cup c_k| - |t \cap c_k| + 1}$$

Given $t_1 = \{\{a\}, \{b\}, \{c\}, \{d\}, \{e\}\}$ and $c_1 = \{\{b\}, \{c\}\}$, here $t_1 \cap c_1 = \{\{b\}, \{c\}\}$ and $t_1 \cup c_1 = \{\{a\}, \{b\}, \{c\}, \{d\}, \{e\}\}$. Using our measure, the similarity between t_1 and c_1 is calculated as $2/(5-2+1) = 0.5$. The greater the overlap between t and C_k , the greater the value of sim coefficient.

Once all transactions are allocated to clusters, further refinement is accomplished by recomputing the centroids which may need to be updated with large items belonging to transactions allocated to a given cluster but not currently part of its centroid. The updating of centroids will result in the need for reorganisation of the clusters, thus the process of centroid update and cluster reorganisation will need to be repeated in tandem until a suitable point of stabilisation is reached. In order to determine the point at which stabilisation is reached, a fitness function adapted from particle swarm optimisation approach was used to find the optimal clusters. For all cluster $\{C_1, \dots, C_k\}$, the fitness function is calculated as:

$$J = \frac{1}{k} \sum_{j=1}^k \frac{\sum_{t \in C_j} d(t, c_j)}{|C_j|}$$

Algorithm for Allocation Phase

Input: Transaction database, $D = \{t_1, \dots, t_n\}$, Cluster Seed, $S = \{s_1, \dots, s_k\}$
Output: Cluster, $C = \{C_1, \dots, C_k\}$
 $J_{prev} \leftarrow 0$
 $C \leftarrow \{C_k \leftarrow \emptyset | k \in S\}$
/* Assign transactions to clusters with the highest similarity */
 $C \leftarrow \{C_k \cup t | \arg \max \{k \mapsto \text{sim}(t, s_k) | s_k \in S\}, t \in D\}$
/* Removes the empty clusters */
 $C \leftarrow \{C_k | C_k \neq \emptyset, C_k \in C\}$
 $J_{curr} \leftarrow \frac{1}{|C|} \sum_{j=1}^{|C|} \frac{\sum_{t \in C_j} \text{sim}(t, C_j)}{|C_j|}$
/* Refine clusters */
while $J_{prev} < J_{curr}$ do
 $J_{prev} \leftarrow J_{curr}$
 $c \leftarrow \{c_k | \{i\} \in C_k, \text{count}(\{i\}, D) \geq |D| * \theta, C_k \in C\}$
 $C \leftarrow \{C_k \cup t | \arg \max \{k \mapsto \text{sim}(t, c_k) | c_k \in c\}, t \in D\}$
 $C \leftarrow \{C_k | C_k \neq \emptyset, C_k \in C\}$
 $J_{curr} \leftarrow \frac{1}{|C|} \sum_{j=1}^{|C|} \frac{\sum_{t \in C_j} \text{sim}(t, c_j)}{|C_j|}$
end while
return C

Figure 2: Algorithm for Allocation Phase

The fitness measure calculates the average similarity between every transaction in a cluster to its centroid and thus the intention is to maximise the fitness value generated. The algorithm for the allocation phase is shown in Figure 2 above.

4 Rare Association Rule Mining via Transaction Clustering (AICluster)

The following is a formal statement of association rule mining for transaction databases. Let $I = \{i_1, i_2, \dots, i_m\}$ be the universe of items and D be a set of transactions, where each transaction T is a set of items. An association rule is an implication of the form $X \rightarrow Y$, where $X \subset I$, $X \subset I$, and $X \cap Y = \emptyset$. X is referred to as the antecedent of the rule, and Y as the consequent. The rule $X \rightarrow Y$ holds in the transaction set D with confidence $c\%$ if $c\%$ of transactions in D that contain X also contain Y . The rule has support $s\%$ in the transaction set D , if $s\%$ of transactions in D contain XY . Throughout this article we shall use XY to denote an itemset that contains both X and Y .

Apriori Inverse

Input: Transaction Clusters (cluster), maxsup value
Output: Rare Itemsets, R
 $N \leftarrow |\text{cluster}|$
 $Idx \leftarrow \text{invert}(\text{cluster}, I)$
 $k \leftarrow 1$
 $R_k \leftarrow \{\{i\} | i \in \text{dom } Idx, \text{count}(\{i\}, Idx) \geq 1\}$
while $(L_k \neq \emptyset)$ do
 $k \leftarrow k + 1$
 $C_k \leftarrow \{x \cup y | x, y \in R_{k-1}, |x \cap y| = k - 2\}$
 $R_k \leftarrow \{c | c \in C_k, \text{count}(\{i\}, Idx) > \text{minabssup}, \text{count}(\{i\}, Idx) < \text{maxsup}\}$
end while
return $\bigcup_{t=2}^{k-1} R_t$

Figure 3: Algorithm for Apriori Inverse

Initially we cluster the transactions into different partitions and then mine each of the partitions for rare association rules. Our rule mining approach is based on the Apriori Inverse algorithm introduced by Koh & Rountree (2005). Apriori Inverse inverted the downward-closure principle of the Apriori algorithm; rather than all subsets of rules satisfying the minsup

lower bound support threshold, all subsets are under maximum support threshold (maxsup). Since making a candidate itemset larger cannot increase its support, all extensions are viable except those that fall under the minimum absolute support requirement. The minimum absolute requirement is necessary to detect noise. Those exceptions are pruned out, and are not used to extend itemsets in the next round. Figure 3 gives the pseudo code for Apriori Inverse.

Our approach is able to produce interesting rules which are not detected in Apriori Inverse. For example, consider a case where we are looking at diagnosis which leads to mortality in a medical scenario. When we partition the datasets into clusters, a trend we may notice is treatment which leads to a much higher mortality rate in the cluster corresponding to the intensive care section when compared to the rest of the dataset. This is however not very interesting. On the other hand, if we detect rules in clusters from the outpatient unit which refer to mortality, this instead would be considered more interesting as it represents relatively rare and unexpected events which deserve closer examination as to the circumstances that led to the fatalities. These types of rules may never have manifested with Apriori Inverse on the unclustered dataset.

We now offer a formal proof of AICluster's rule coverage vis-a-vis the Apriori Inverse algorithm.

Lemma 1. *If a rule R exists on the unclustered data set then that rule must have confidence $C_i > C_{min}$ on at least one cluster cl_i .*

Proof. Let N be the total number of clusters. Suppose that the Lemma was false and hence for all clusters, $cl_i = 1, \dots, N$ we have $C_i \leq C_{min}$ for rule R . (1)

Let the number of instances of the antecedent of the rule be L_i for the i th cluster, cl_i . Let the number of instances of the antecedent and the consequent occurring together in the rule be LR_i for the i th cluster. We now have $L_i P_i = LR_i$ where P_i is essentially the confidence for i th cluster.

We now have $\sum_{i=1}^N L_i P_i = \sum_{i=1}^N LR_i \leq C_{min} \times \sum_{i=1}^N L_i$ from (1) above. Let us denote $\sum L_i P_i$ by LR . We thus have $LR \leq C_{min} \times \sum_{i=1}^N L_i$. (2)

Now consider the rule on the unclustered data set. We have $LC = GLR$ where LC is the support of the antecedent of the rule and GLR is the number of instances of the antecedent and consequent occurring together in the rule. We also have $C > C_{min}$ since the rule R exists on the unclustered data set and so we have $GLR > C_{min} \times \sum_{i=1}^N L_i$ from (2) above.

We also have $GLR = LR$ as the total number of occurrences taken across all clusters must be the same as the total number of instances across the unclustered data set, as the instances in the unclustered data set is the union of all instances in the clusters.

Substituting for GLR in the expression above we have $LR > C_{min} \times \sum_{i=1}^N L_i$ (3), which leads to a contradiction with (2) above. Thus our initial assumption that the Lemma is false is untrue and this proves the Lemma. \square

Theorem 1. *AICluster together with traditional frequent mining has a coverage that is greater than a combination of Apriori Inverse with traditional association rule mining.*

Proof. Once again consider a rule R that exists on the unclustered data set. Now suppose that

the rule R does not exist on any of the clusters cl_i where $i = 1, \dots, N$. Let us consider the case where rule R is not picked across any of the clusters.

$C_i \leq C_{min}$ or $S_i \geq S_{max}$ (4) for all clusters $i = 1, \dots, N$; where S_i denotes the support of rule R on cluster cl_i and S_{max} is the upper bound support threshold for finding rare rules.

According to the Lemma above there must exist at least one cluster where the confidence of the rule exceeds C_{min} . Let us pick one of these clusters at random, say cl_j . We now have $S_j > S_{max}$ for this cluster from (4) above since $C_j > C_{min}$.

This means that the rule meets the confidence value on cluster cl_j and the only thing preventing it from appearing is the upper bound support threshold which was set to be S_{max} . This means that the rule will be discovered R will be discovered across at least one cluster under traditional (frequent) association rule mining.

In a real world setting we envisage that rare association rule mining will be done in conjunction with frequent rule mining and thus any rule discovered on the unclustered data set will be picked up by AICluster under association rule mining as a whole.

We now consider the reverse situation. Will all rules discovered by AICluster be also discovered by Apriori Inverse? This will not be the case as we have certain rules R' that apply on clusters that will not apply on the unclustered data set as rules such as R' will fail to meet the lower bound confidence threshold C_{min} . There are two cases to consider:

Case 1: The antecedent occurs in other clusters without the occurrence of the consequent thus lowering the confidence of the rule R' on the unclustered data set. This is the effect of the contamination issue that we referred to earlier in the paper.

Case 2: The antecedent does not occur in any other clusters. In this case it is possible that the Fisher test fails as the number of co-occurrences of the antecedent and consequent is a very small proportion of the total number of instances across the unclustered data set (which is much larger than the number of instances in the cluster where R' is true). Thus the Fisher test would flag these co-occurrences as chance collisions.

The cases 1 and 2 represent cases where rules on clusters do not manifest on the unclustered data set. We thus make the claim that AICluster outperforms Apriori Inverse with respect to rule coverage. \square

In the next section, we present the results from our technique and compare our approach with the Apriori Inverse algorithm.

5 Experimental Results

In this section, we compare the performances of the standard Apriori Inverse algorithm with our proposed Apriori Inverse with Clustering (AICluster) algorithm. Testing of the algorithms was carried out on seven different datasets from the UCI Machine Learning Repository (Newman et al. 1998).

Table 1 represents the rules found using the AICluster and Apriori Inverse algorithms. For AICluster we set the maximum support threshold

Table 1: Results Based On AICluster and Apriori Inverse algorithm

Dataset	AICluster		Apriori Inverse					
	maxsup (0.30)		maxsup (0.10)		maxsup (0.20)		maxsup (0.30)	
	Rules	Avg Rule Support	Rules	Avg Rule Support	Rules	Avg Rule Support	Rules	Avg Rule Support
Zoo	2	4	1	8	10	11	72	13
Hepatitis	1	3	0	0	11	6	11	6
Flag	20	4	3	4	27	6	135	9
Heart	2	6	0	0	0	0	0	0
Soybean-Large	1862	6	166	7	6446	7	6975	8
Congressional Votes	3	3	0	0	0	0	0	0
Dataset	maxsup (0.01)		maxsup (0.05)		maxsup (0.10)		maxsup (0.15)	
	Rules	Avg Rule Support	Rules	Avg Rule Support	Rules	Avg Rule Support	Rules	Avg Rule Support
Mushroom	76179	4	10772	17	27505	17	39543	17

(maxsup) to 0.30 for all datasets except for the mushroom dataset. For six of the datasets, we ran Apriori Inverse at three different maxsup values of 0.10, 0.20, and 0.30. This was done in order to obtain a benchmark for comparison with AICluster on the all important rule support measure. In all of the experiments, we set the minimum threshold values for confidence and lift to 0.90 and 1.0 respectively.

We now compare the number of rules generated using AICluster with maxsup at 0.30 and Apriori Inverse with maxsup at 0.10. From Table 1 above we can clearly see that comparable levels of actual rule support occur at a maxsup of 0.3 for AICluster and 0.1 for Apriori Inverse. As we expected, the maxsup threshold for AICluster had to be set higher due to the fact that each cluster is smaller in size than the unclustered dataset. At these support thresholds we can see from Table 1 that the rule coverage for AICluster is consistently greater than that of Apriori Inverse for the first six datasets in Table 1 (with the exception of mushroom).

In the case of the mushroom dataset we lowered the maxsup thresholds for both algorithms in view of its relatively large size. We set maxsup to 0.01 for AICluster and then tested Apriori Inverse at lower values to compensate for the larger dataset size that it operates on, in keeping with our experimentation other six datasets that we experimented with. However, these settings caused Apriori Inverse to perform very poorly with respect to rule coverage and we thus decided to test it with maxsup values of 0.05, 0.1 and 0.15. Despite these favourable settings for Apriori Inverse, Table 1 shows that AICluster still significantly outperforms Apriori Inverse with respect to rule coverage.

Table 2 gives a more in-depth view of the rule bases covered by the two algorithms. The most striking feature is the very low degree of overlap between the two algorithms. The degree of overlap ranged from 0.73% (for mushroom) to 10% (for the much smaller Flag dataset) across the range of datasets tested. Coupled with the fact that AICluster on its own covers a very large percentage (93% and 88% for the two larger datasets, soybean and mushroom respectively) of the total rule base in the rare mining mode, this shows once again its superiority over Apriori Inverse in terms of rule coverage. As shown in section 4 above this percentage will rise to 100% when frequent association rule mining is done in conjunction with rare mining via AICluster.

In the next section we analyse the information content of the rules produced by these techniques.

5.1 Rule Analysis on Congressional Votes

The AICluster algorithm identified 4 rare rules from the set of clustered transactions in Congressional Votes dataset using a maximum support threshold of 0.3. Clustering on this dataset produced a total of 4 clusters. Two of the rules were from cluster 0 and the other 2 rules came from cluster 3.

Cluster 0:

physician-fee-freeze:y → Class:republican, (Conf:1.00, Lift:31.0)

Class:republican → physician-fee-freeze:y, (Conf:1.00, Lift:31.0)

Cluster 3:

anti-satellite-test-ban:y → export-administration-act-south-africa: y, (Conf:0.94, Lift:1.54)

physician-fee-freeze:n → Class:democrat, (Conf:1.00, Lift:16.83)

However the Apriori Inverse algorithm failed to produce any rare rules at the maximum support threshold that we set. This is due to the groupings contaminating each other and preventing candidate rules from meeting the minimum confidence threshold.

5.2 Rule Analysis on Zoo

Using the AICluster algorithm we were able to find 2 rules from the set of clustered transactions in the Zoo Dataset.

Cluster 0:

fins:1 → aquatic:1 (Conf =1.00, Lift =6.33)

legs:0 → fins:1 (Conf =1.00, Lift =9.50)

These two rules are particularly interesting, as the class of animals in the cluster was Type 1 which was mammal. In this instance, we only had three transactions (seal, dolphins, porpoise) which have fins, are aquatic and are mammals. Finding these rules within this cluster is indeed interesting. In the case of Apriori Inverse we were able to detect only 1 rule with maxsup at 0.10.

Apriori Inverse(maxsup =0.10)

type:6 → legs:6 (Conf:1.00, Lift:10.10)

Setting the maximum support at 0.30, we were able to find 2 rules using AICluster which were not belonging to “Type 6”. This was because the dataset was clustered in such a manner which allowed all homogeneous transactions to be clustered together. In each of the homogeneous clusters we were not able to find any rare rules. “Type 6” appeared 8 times within

Table 2: Summary of Rules from AICluster and Apriori Inverse

Dataset	No. of Rules Apriori Inverse	No. of Rules AICluster	No. of Rules Complete Set	No. of Rules Overlap
Zoo	1	2	3	0
Hepatitis	0	1	1	0
Flag	3	20	21	2
Heart	0	2	2	0
Soybean-large	166	1862	1993	35
Congressional Votes	0	3	3	0
Mushroom	10772	76179	86314	637

the original dataset which had 101 transactions. After transaction clustering we found that type 6 was clustered together in a particular cluster. The cluster consists of “Type 6” and 8 transactions. The support of “Type 6” in the cluster was 1.00 and in effect was no longer considered rare.

5.3 Rule Analysis on Heart Dataset

Using AICluster algorithm we were able to find 2 rules from the set of clustered transactions in the Heart Cleveland Dataset. The attribute *num* represents the diagnosis of heart disease and *ca* represents the number of major vessels (0-3) coloured by fluoroscopy. The attribute *thal* has three values, 3 for normal, 6 for fixed defect, and 7 for reversible defect.

Cluster 0:

num:3 \rightarrow ca:2 (Conf =1.00, Lift =11.22)

Cluster 1:

num:0 \rightarrow thal:normal (Conf =0.92, Lift =2.08)

We were able to find one rule with AICluster which is the rule found in Cluster 0 above. However we were not able to detect any rules with Apriori Inverse on the unpartitioned dataset due to contamination from other sub groupings.

6 Conclusion

Our approach first clustered transactions into homogeneous clusters and then generated rare rules from each of the clusters formed. These rules expressed their own associations without interference or contamination from other sub groupings that have different patterns of relationships. We were able to demonstrate that the clustering process added value to the rare association rule mining process, generating interesting rules that could not be found otherwise.

One possible direction for future work would be to weigh itemsets according to their support, with lower frequency itemsets given a higher weight over their higher frequency counterparts. This would lead to the generation of rare rules that have lower frequency in each of their individual terms than would be possible with the current version of AICluster.

References

- Agrawal, R., Imielinski, T. & Swami, A. (1993), Mining association rules between sets of items in large databases, in P. Buneman & S. Jajodia, eds, ‘Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data’, pp. 207 – 216.
- Cutting, D. R., Pedersen, J. O., Karger, D. & Tukey, J. W. (1992), Scatter/gather: A cluster-based approach to browsing large document collections, in ‘Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval’, pp. 318–329.
- Ganti, V., Gehrke, J. & Ramakrishnan, R. (1999), CACTUS: Clustering categorical data using summaries, in ‘KDD ’99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining’, ACM Press, New York, NY, USA, pp. 73–83.
- Guha, S., Rastogi, R. & Shim, K. (2000), ‘ROCK: A robust clustering algorithm for categorical attributes’, *Information Systems* **25**(5), 345–366.
- Jain, A. K., Murty, M. N. & Flynn, P. J. (1999), ‘Data clustering: a review’, *ACM Comput. Surv.* **31**(3), 264–323.
- Koh, Y. S. & Pears, R. (2008), Transaction clustering using a seeds based approach, in T. Washio, E. Suzuki, K. M. Ting & A. Inokuchi, eds, ‘PAKDD’, Vol. 5012 of *Lecture Notes in Computer Science*, Springer, pp. 916–922.
- Koh, Y. S. & Rountree, N. (2005), Finding sporadic rules using Apriori Inverse, in T. B. Ho, D. W.-L. Cheung & H. Liu, eds, ‘PAKDD’, Vol. 3518 of *Lecture Notes in Computer Science*, Springer, pp. 97–106.
- Koh, Y. S., Rountree, N. & O’Keefe, R. A. (2008), ‘Mining interesting imperfectly sporadic rules’, *Knowl. Inf. Syst.* **14**(2), 179–196.
- Liu, B., Hsu, W. & Ma, Y. (1999), Mining association rules with multiple minimum supports, in ‘Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining’, pp. 337 – 341.
- Newman, D., Hettich, S., Blake, C. & Merz, C. (1998), ‘UCI repository of machine learning databases’, <http://www.ics.uci.edu/~mllearn/MLRepository.html>.
- Plasse, M., Niang, N., Saporta, G., Villeminot, A. & Leblond, L. (2007), ‘Combined use of association rules mining and clustering methods to find relevant links between binary rare attributes in a large data set’, *Computational Statistics & Data Analysis* **52**(1), 596–613.
- Wang, K., Xu, C. & Liu, B. (1999), Clustering transactions using large items, in ‘CIKM ’99: Proceedings of the Eighth International Conference on Information and Knowledge Management’, ACM Press, New York, NY, USA, pp. 483–490.
- Xu, J., Xiong, H., Sung, S. Y. & Kumar, V. (2003), A new clustering algorithm for transaction data via caucus, in ‘Advances in Knowledge Discovery and Data Mining: 7th Pacific-Asia Conference, PAKDD 2003, Seoul, Korea, April 30 - May 2, 2003. Proceedings’, pp. 551–562.

- Yun, C.-H., Chuang, K.-T. & Chen, M.-S. (2001), An efficient clustering algorithm for market basket data based on small large ratios, *in* 'COMPSAC '01: Proceedings of the 25th International Computer Software and Applications Conference on Invigorating Software Development', IEEE Computer Society, Washington, DC, USA, pp. 505–510.
- Yun, H., Ha, D., Hwang, B. & Ryu, K. H. (2003), 'Mining association rules on significant rare data using relative support', *The Journal of Systems and Software* **67**(3), 181 – 191.

S²MP: Similarity Measure for Sequential Patterns

Hassan Saneifar

Sandra Bringay

Anne Laurent

Maguelonne Teisseire

Laboratory of Informatics, Robotics, and Microelectronics of Montpellier (LIRMM)
University of Montpellier 2,
161 rue Ada, 34392 Montpellier Cedex 5, France
Email: {saneifar, bringay, laurent, teisseire}@lirmm.fr

Abstract

In data mining, computing the similarity of objects is an essential task, for example to identify regularities or to build homogeneous clusters of objects. In the case of sequential data seen in various fields of application (e.g. series of customers purchases, Internet navigation) this problem (*i.e. comparing the similarity of sequences*) is very important. There are already some similarity measures as Edit distance and LCS suited to simple sequences, but these measures are not relevant in the case of complex sequences composed of sets of items, as is the case of sequential patterns. In this paper, we propose a new similarity measure taking the characteristics of sequential patterns into account. S^2MP is an adjustable measure depending on the importance given to each characteristic of sequential patterns according to context, which is not the case of existing measures. We have experimented the accuracy and quality of S^2MP against Edit distance by using them in a clustering of sequential patterns. The results show that the clusters obtained by S^2MP are more homogeneous. Moreover these clusters are more precise and more complete according to the clusters obtained using Edit distance. The experiments show also that S^2MP is efficient in term of calculation time and size of used memory.

Keywords: Data Mining, Sequential Patterns, Similarity Measure, Clustering, Clustering of Sequential Patterns, S^2MP .

1 Introduction

In some areas, like biology, logs analysis, anomaly detection, natural language processing and telecommunications, data can be seen in the form of sequences. Sequential patterns introduced by Agrawal & Srikant (1995) represent a frequent diagrams often extracted from sequential databases. Sequential patterns can be considered as an extension of association rules on the dimension of time. Indeed, they highlight inter-transaction associations. For example, the frequent sequential patterns extracted from a market basket data identify common and frequent customers behaviour in terms of purchased products. An example of sequential patterns is $\langle (Chocolate, Soda)(cakes, chips)(leanness product) \rangle$, which means: "customers buy chocolate and soda in the same time, then in the next purchase, they buy cakes and chips and then they come back later on to buy a slimming product."

The extraction of frequent sequential patterns in these areas provides important knowledge about frequent correlations. However, these patterns do not always convey enough information for the end-users. In order to get a clear view of the data, the clustering of sequential patterns is for instance a solution to group similar behaviours uncovered by frequent sequential patterns. This facilitates the interpretation of sequential patterns, allows to model behaviours and also to seek for outliers in the data. For clustering sequential patterns, the similarity of sequential patterns needs to be computed. Note that comparing sequential patterns has many other applications than clustering. For example, the extraction of sequential patterns under similarity constraints is of great interest, as well as sequential pattern visualization.

In this context, the definition of similarity may vary depending on the type of resemblance that we look for. The different similarity measures may reflect the different faces of data and of their context. Two objects can be seen very similar by one measure and very different by another measure (Moen 2000). Moreover, we argue that similarity has not to be always symmetrical, as pointed out by Tversky (1977), with the famous example he provided in his seminal paper: We say "Turks fight like tigers" and not "Tigers fight like Turks". In some applications, for instance, the extraction of sequential patterns under similarity constraints or querying a set of sequential patterns, there is a reference pattern that we compare others with. Indeed, we do a directional comparison. In these applications, a non-symmetric measure is well applicable. Several approaches have been developed to compare the similarity between two sequences in particular in bio-informatics. However, the existing measures are not adapted to the specific characteristics of sequential patterns.

To compute the similarity of sequential patterns, we define here a similarity measure (S^2MP : Similarity Measure for Sequential Patterns), which takes the characteristics and the semantics of sequential patterns into account. This measure compares two sequential patterns both at the level of itemsets and their positions in the sequences and also at the level of items in itemsets, thus resulting from the combination of two scores:

1. the score given by the weight of itemsets mapping (resemblance between the items of mapped itemsets),
2. the score given by the resemblance of corresponding itemsets in terms of their positions in the two sequences.

S^2MP is a measure which is very well suited to the contexts and the characteristics of sequential patterns. Each score may vary according to the context. It makes hence S^2MP a modular measure. For example, according to the definition of resemblance of the itemsets in a particular field, we can adapt the score given by the weights

of the mapping of the itemsets. We can also decide that the order is more important than the resemblance of the mapped itemsets and change the coefficients of the two scores in the calculation of the final similarity. This makes our measure flexible, which is not the case of the other existing measures.

The paper is organized as below. Section 2 states the problem. In Section 3 we describe the existing works on the similarity measures for the sequential patterns. Our similarity measure (S^2MP) is presented in Section 4. The results obtained by some experiments on S^2MP are detailed in Section 5.

2 Problem Statement

The volumes of data stored in databases are dramatically increasing. In many applications like telecommunication, bio-informatics, market basket data etc. the data are stored in sequential form. In general, we can consider two major types of **sequences**:

- sequence of items,
- sequence of itemsets.

The sequence of items, are the most simple kind of sequences. In such a sequence, the elements of sequence are atomic. But, in many real application (e.g. market basket data), the sequences have more complex elements. The sequences of itemsets are an example of complex sequence. In Data Mining problems, we can note two kinds of **itemset sequences**:

- Data sequences,
- Frequent Sequential Patterns.

We consider here a transactional database of market basket data containing a set of transactions, where every transaction is a *set of items* (attributes) usually referred to as an *itemset*. A data sequence is defined as follow:

Definition 2.1. A **data sequence** consists of all transactions of a customer when they are ordered chronologically.

Frequent sequential patterns introduced by Agrawal & Srikant (1995) are a kind of **schema** extracted from data sequences. Indeed, frequent sequential patterns are the frequent subsequences of data sequences of a transactional database. We define the sequential patterns as below:

Definition 2.2. A **sequential pattern** is a non-empty ordered list of itemsets where an itemset is a set (non-ordered) of items.

Although the data sequences and sequential patterns are semantically different (sequential patterns are the schemas extracted from data sequences), they share some common characteristics. The main characteristics of itemset sequences (e.g. sequential patterns and data sequences) are:

1. itemsets as a set of items (non-ordered),
2. order of itemsets in sequence.

As described, in itemset sequence (e.g. data sequences, sequential patterns), the elements of sequence (i.e. itemsets) are composed of various items. Thus, the ways that we treat the sequences of items are not necessarily adapted to the sequences of itemsets like sequential patterns. In this paper, we are specially interested to compare the similarity between the sequential patterns.

Sequential patterns are very interesting kind of diagram extracted from sequential data. They describe the

inter-transaction correlations. In order to find regularities from such data (**itemset sequences**), it is necessary to describe how far from each other two data objects are. This is the reason why **similarity** between objects is one of the central concepts in data mining and knowledge discovery (Moen 2000). According to the volumes of data, we should consider also the scalability aspect of the similarity measures.

A similarity measure for sequential patterns can be used for clustering of sequential patterns. The principle of such a clustering is to regroup the extracted sequential patterns into several clusters. Each cluster represents a homogeneous kind of correlations. The clustered sequential patterns can, for instance, be used to create the behaviour profiles. For anomaly detection using data mining techniques, for example, we can use the sequential patterns extracted from normal connection logs to identify the general behaviour of network users. Several kinds of extraction are conceivable. But in any case, the patterns should be regrouped to achieve more abstract behaviour representation. The clustering of sequential patterns is a relevant and scalable solution for behaviour modeling. (Sequeira & Zaki 2002).

Besides, by clustering, outliers in data can be identified. Sequential patterns which are not assigned to any cluster, may be considered as anomalous. In market basket data, for example, the clustering of sequential patterns may help in customer segmentation or prediction in terms of their purchasing behaviour.

The notion of similarity between sequential patterns could also be used when extracting sequential patterns. The extracted sequential patterns by apriori-like algorithms (Agrawal & Srikant 1995) are usually very voluminous. There are thus many works trying to integrate some constraints like similarity constraint (Capelle et al. 2002) to reduce the size of the output of the algorithms and to better meet the end-user needs. Given a reference pattern, the idea is to extract only the patterns that are similar to the reference pattern.

The querying a set of sequences is another application of similarity measure for sequential patterns. Given a sequential pattern as a query, for example, we look for the similar patterns. Querying sequences sets has real application in bio-informatics and more generally in sequential patterns visualisation.

As described, a similarity measure has many applications in sequence analysis especially when considering sequential patterns. A great deal of works has been done in the field of item sequences. On the contrary, there are not so many works in itemset sequences area. The existing measures, used for item sequences, are not necessarily adapted to the itemsets sequences. Hence, we define a similarity measure which takes the characteristics of itemset sequences into account. As domain knowledge about the notion of similarity can vary according to different contexts, we define a similarity measure that is adaptable to the context. That is the reason why we define a modular measure by combining two scores. The final similarity degree is the weighted average of values of these scores.

3 Related Works

We report here the main approaches dealing with comparing sequential data especially sequential patterns. The two main similarity measures used for itemset sequences are **Edit distance** and **LCS**. We explain the disadvantages of these two measures for sequential patterns. Next, we cite an approach based on the comparison of corresponding itemsets.

The **Edit distance** (Levenshtein 1966) was used by Capelle et al. (2002) for extracting sequential patterns under similarity constraints. The authors define a sequen-

tial pattern as an ordered list of symbols belonging to Σ where Σ is a finite set of alphabet. We show why this measure is irrelevant for sequential patterns by taking an example according to the given definition and representation of sequential patterns by Capelle et al. (2002):

Example 3.1. Given two sequential patterns $M_1 = \{(ab)(c)\}$ and $M_2 = \{(a)(c)\}$ represented as:
 $M_1\{(ab)(c)\} \Rightarrow X \rightarrow Y \mid X = (ab), Y = (c)$
 $M_2\{(a)(c)\} \Rightarrow Z \rightarrow Y \mid Z = (a), Y = (c)$

where X, Y, Z are symbols from Σ

Since Edit distance's operators are applied on the elements of sequence (*i.e. itemsets*), the distance is the cost of replacing X and Z ($repl(X, Z, 1)$). In fact, in this work, an itemset in a sequential pattern is reduced to an event type. Hence, a sequential pattern is treated as an event type sequence¹. In such a sequences, an event type is characterized by the values of some attributes. A list of event types ordered according to the occurrence time of events is an event type sequences (Mannila & Ronkainen 1997, Moen 2000). As described, in this work, an itemset is seen as an event type when their items are considered as the values of the event's attributes. Hence, the itemsets (ab) and (a) are treated as two symbols (event) completely different. However, we argue that (ab) and (a) are not completely different; but on the contrary, these are two similar behaviours.

Although a sequential pattern is sometimes seen as an event sequence (like in this work), this interpretation of itemset as an event is not always relevant. We explain this issue in more details with Examples 3.2 and 3.3.

Example 3.2. Let $R = \{Sen_1, Sen_2, \dots, Sen_m\}$ be a set of sensors in an automatic alarms system. An alarm (*event*) occurs according to the values of sensors within a specific time. $Alarm_A$, for instance, is characterized by the values of sensors:

$A = (Sen_1 = 0, Sen_2 = 1, Sen_3 = 0)$. If, for example, the value of Sen_3 becomes 1, the system is in a new situation and it sets off hence another alarm $Alarm_B = (Sen_1 = 0, Sen_2 = 1, Sen_3 = 1)$. We see that despite a small difference in the attributes (value of sensor Sen_3), but according to the data and the context, $Alarm_A$ and $Alarm_B$ are two events (*situation of the system*) completely different.

Example 3.3. Let us now consider a sequential pattern:

$M = \{(chips, soda, breads)(pizza)(chips, soda, chocolate)(flour)\}$ extracted from market basket data. The two itemsets $(chips, soda, breads)$ and $(chips, soda, chocolate)$ differ by a single item, but in this context, we know that these two items correspond to two very close behaviours of a customer. Thus, we can not consider them as two behaviours (*events*) being completely different.

The Edit distance measure is also used in ApproxMAP approach to cluster the sequences of itemsets. ApproxMAP developed by Kum et al. (2003), identifies the consensus sequences in large database in two phases : (1) clustering of itemsets sequences (2) extraction of consensus patterns directly from each cluster. In Phase 1, the authors used Edit distance as a measure of similarity, but with a modification on the cost of "replacement operator" to adjust the measure to itemsets sequences. The *normalized set difference* is adopted as the cost of replacement operator. Although this modification overcomes the disadvantage of Edit distance argued previously, the authors noted that the *normalized set difference* emphasis the common elements. This behaviour is appropriated if the commonalities are more important than the differences.

In addition, as noted by (Moen 2000), the type of edit operations and their costs have a remarkable effect on what kind of sequences are considered to be similar or not. She indicates that it is more natural to give more weight to the insertion (remove) of rare itemsets that to

insertion (remove) of frequent itemsets in the sequences. With different choices, we obtain different results.

Edit distance is not adaptable (*configurable*) to the various definitions of similarity. In sequential patterns extracted from bio-informatics data (*data from the analysis of DNA chips*), for example, to seek similar patterns, the content of itemsets (*i.e. items*) is more important than the order of itemsets. It means that if we look for similar patterns, we should consider the similarity of itemsets according to their contents more important than the similarity of itemset's order. However, Edit distance does not take this characteristic of data into account. Hence, Edit distance is not sensible to the different definitions of similarity for sequences.

According to these examples, for a relevant comparison, it is necessary to compare two sequential patterns by considering their itemsets and their positions in sequences and importantly by considering the content of itemsets (*i.e. common and non-common items*). We should pay attention that the elements of a sequential patterns (*i.e. itemsets*) are not the atomic elements. We should not hence treat them without considering their content (*items*). Also, it is necessary that a similarity measure be adaptable to the different contexts and to the characteristics of data. This allows us to capture different kind of similarity.

The **LCS** measure (Longest Common Subsequence) is used for the comparison of sequences (Sequeira & Zaki 2002). The *LCS* gives the length of the longest common subsequence of two sequences. It is possible to use *LCS* to compare the similarity of sequential patterns (*itemsets sequences*) without being optimal. We note three reasons why the *LCS* is not a optimal measure for sequential patterns (*itemset sequences*).

Firstly, *LCS* does not take the position of itemsets (*in order of sequence*) into account in the two sequences.

Example 3.4. Let us consider:

$M_1 = \{(bc)(df)(e)\}$
 $M_2 = \{(abc)(mn)(de)(egh)(fg)\}$
 $M_3 = \{(e)(bc)(df)\}$.

$LCS(M_1, M_2) = 2$ and $LCS(M_1, M_3) = 2$ with the longest common subsequence = $\{(bc)(d)\}$. This subsequence ($\{(bc)(d)\}$) corresponds to the consecutive itemsets in $M_1 = \{(bc)(df)(e)\}$ and $M_3 = \{(e)(bc)(df)\}$. But, it is not appeared consecutively in $M_2 = \{(abc)(mn)(de)(egh)(fg)\}$. Obviously, *LCS* does not take into account this fact. However, semantically the emergence of the subsequence $\{(bc)(d)\}$ in M_1 and M_3 is not similar to its appearance in M_2 .

Secondly, *LCS* does not consider the length of the part which is not common.

Example 3.5. The non-common part in M_2 (*i.e. $\{(abc)(mn)(de)(egh)(fg)\}$*) is longer than M_3 's (*i.e. $\{(e)(bc)(df)\}$*). This is because the value of *LCS* is not normalized by the number of items in the sequence.

Thirdly, the number of different items in itemsets (in which the subsequence appears) does not affect the value of *LCS*.

Example 3.6. In M_2 , the itemset (bc) of subsequence is included in the itemset (abc) while in M_1 and M_3 , it is included in the itemset (bc) . *LCS* does not consider that in the itemset (abc) of M_3 , there is another item different from " bc " (*i.e. "a"*). This problem is not resolved with the normalization because it depends on the number of items in sequence and not by the number of items in itemsets in which the subsequence appears.

A comparison of the similarity between two multidimensional sequential patterns is done by Plantevit et al.

¹Edit distance is also used for event sequences (Moen 2000)

(2007) for outliers detection in a data cube. The distance between two multidimensional sequences is defined as: Let $S_1 = \{b_1, b_2, \dots, b_k\}$ and $S_2 = \{b'_1, b'_2, \dots, b'_k\}$ be two multidimensional sequences, $dist$ is a distance measure and Op an aggregation operator. The distance between S_1 and S_2 is defined as follows: $d(s_1, s_2) = Op(dist(b_j, b'_j))$ for $j = 1 \dots k$. Comparison is done between corresponding blocks². It means that we compare the block i in S_1 with the block i in S_2 . This kind of comparison is not useful when, for example, there is a shift in one of the sequence.

There are some approach developed to compare the XML document (Lee et al. 2004). A XML structure can be seen as a sequence of complex objects. But in this kind of sequences the object are not ordered. It means that there is not any order relation between objects. But in this paper, we introduce a similarity measure for sequential patterns. The order relation between the objects of a sequential pattern (*i.e.* itemsets) is a fundamental concept. Thus, we must compare the sequential pattern according to their order.

The existing measures in the literature have some disadvantages in the case of sequential patterns or they are just applicable for sequences of items. A similarity measure for sequential patterns must take into account the fact that sequential patterns are sequences of ordered itemsets and not items. The positions of itemsets in sequences (*distance in order*) must also be taken into account when calculating the similarity. Moreover, the number of common items and non-common items must be considered both at the sequence level as well as at the level of corresponding itemsets.

In this paper, we define a similarity measure (S^2MP : Similarity Measure for Sequential Patterns) which takes these characteristics into account.

4 S^2MP : Description

Our similarity measure (S^2MP : Similarity Measure for Sequential Patterns) results from the aggregation of two scores:

1. The *mapping score* which measures the resemblance of two sequences based on the links that can be established between itemsets,
2. The *order score* which measures the resemblance of the two sequences based on the order and positions of itemsets in sequences.

As we consider the itemset proximity as very important, we first build the best mapping of itemsets based on the similarity of their contents (common and non-common items) in step 1. The mapping score is calculated at the end of this step by considering all the mappings and their degree of mapping (*weight of mapping*).

In step 2, the goal is to give a score according to the resemblance of two sequences in terms of their order and position of mapped itemsets in two sequences. We take the result of step 1 without any rearrangement of the mappings. Then, we are firstly looking for the mappings, which comply with the order of itemsets in two sequences (*cf.* Figure 2). It means that we discard the mappings, which associate the current itemset of a sequence to an itemset before the last mapped itemset of other sequence. This kinds of mapping are called cross-mappings (*cf.* Figure 2). Moreover, we also measure the resemblance of the mapped itemsets according to their positions in two sequences.

Finally, the order score is calculated based on the proportion of the mappings complying with order to the all

mappings in addition to the score measuring the similarity of itemsets according to their positions.

Step 1: Mapping Score Calculation. Let us consider Seq_1 and Seq_2 as two sequential patterns to be compared to. For each itemset i in 1^{st} sequence $Seq_1(i)$, we are looking for the most similar itemset j in the 2^{nd} sequence $Seq_2(j)$. Then we match these two itemsets and we give each pair of mapping a weight which is the degree of similarity between the two itemsets. We define below how the weight of mapping is computed:

Definition 4.1. *Weight*(i, j) between the i^{th} itemset of Seq_1 and the j^{th} itemset of Seq_2

$$Weight(i, j) = \frac{|Seq_1(i) \cap Seq_2(j)|}{(|Seq_1(i)| + |Seq_2(j)|) / 2}$$

For equal weights, we choose the itemset with the lowest *timeStamp* (ts)³. It means the itemset located before others regarding the order of sequence to comply with the order of itemsets. Thus, the mapping is formalized as:

$$Mapping(Seq_1(i), Seq_2(j)) \mid Weight(i, j) = \max_{x \in [0, |Seq_2|]} (Weight(i, x)) \\ \&\& Weight(i, j) \neq 0$$

$$If \ Weight(i, j) = Weight(i, k) \Rightarrow ts(j) < ts(k)$$

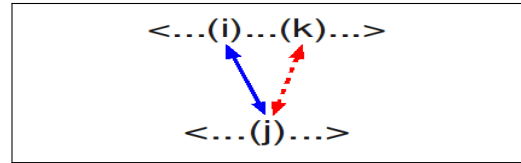


Figure 1: Conflict of mapping.

Conflict of mapping. When computing the mappings, an itemset selected from the 2^{nd} sequence to correspond to an itemset of the 1^{st} sequence may have already been mapped with another itemset. This situation, called “conflict of mapping”, is illustrated in Figure 1. The itemset $Seq_2(j)$ is proposed to map with $Seq_1(k)$. But $Seq_2(j)$ is already mapped with $Seq_1(i)$. To solve this problem, we must find another mapping candidate for one of the itemsets in conflict. For this, we use a function (*SolveConflict*) to propose a new itemset as a new candidate for mapping.

In the *SolveConflict*, for each itemset (*in conflict*) of sequence 1 (*i.e.* $Seq_1(i)$ and $Seq_1(k)$), we are seeking two other mapping candidates in the Seq_2 (other than $Seq_2(j)$, the current candidate itemset):

- The 1^{st} candidate is the itemset located before $Seq_2(j)$ which also owns the maximum weight among itemsets placed before $Seq_2(j)$. We name them as: *nextMaxBefore_i* as a candidate for $Seq_1(i)$ and *nextMaxBefore_k* for $Seq_1(k)$.
- The 2^{nd} candidate is selected the same way but it is sought after $Seq_2(j)$: *nextMaxAfter_i* for $Seq_1(i)$ and *nextMaxAfter_k* for $Seq_1(k)$.

Next, we create all possible mapping pairs. We get at the most four possible cases of mapping:

- $\langle Seq_1(i), Seq_2(j) \rangle, \langle Seq_1(k), nextMaxBefore_k \rangle,$
- $\langle Seq_1(i), Seq_2(j) \rangle, \langle Seq_1(k), nextMaxAfter_k \rangle,$
- $\langle Seq_1(k), Seq_2(j) \rangle, \langle Seq_1(i), nextMaxBefore_i \rangle,$
- $\langle Seq_1(k), Seq_2(j) \rangle, \langle Seq_1(i), nextMaxAfter_i \rangle.$

²Here, a data block can be seen as an itemset

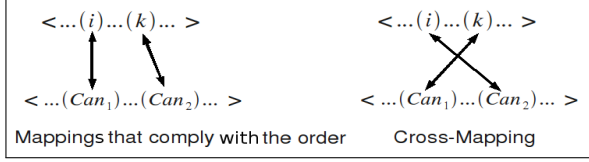


Figure 2: Cross mapping and the order of itemsets.

We consider here two opposite kinds of mapping pairs, namely mapping pairs that comply with the order and mapping pairs, which violate the order (see Figure 2) referred to as *cross-mappings*.

More precisely, let us consider a pair of mappings: in the first mapping the itemset at position i on the first sequence is mapped with the itemset at position i' in the second sequence; in the second mapping the itemset at position j (where $i < j$) on the first sequence is mapped with the itemset at position j' in the second sequence. Then this mapping pair is said to be *order compliant* if $i' < j'$. It is said to be *cross-mapping* is $i' > j'$.

We calculate the relevance of the four possible cases of mapping with *localSim*. The calculation of *localSim* depends on the type of mappings:

– when mappings comply with order, we consider:

$$\text{localSim}(i, \text{Can}_1)(k, \text{Can}_2) = \frac{\text{Weight}(i, \text{Can}_1) + \text{Weight}(k, \text{Can}_2)}{2}$$

– when we have cross mapping, as the order is half respected in cross-mapping, we divide the *localSim* by two. We thus consider:

$$\text{localSim}(k, \text{Can}_1)(i, \text{Can}_2) = \frac{1}{2} \times \frac{\text{Weight}(k, \text{Can}_1) + \text{Weight}(i, \text{Can}_2)}{2}$$

The mapping pair having the highest *localSim* is then selected as the output of the *SolveConflict* function. Note that at the end, the initial candidate (*i.e.* $\text{Seq}_2(j)$) is proposed as a candidate either for $\text{Seq}_1(i)$, or for $\text{Seq}_1(k)$ depending on the value of *localSim*.

Conflict Loop. Mapping function handles the cases when there is a conflict loop. The conflict loop is a situation where the candidate itemset, which is proposed to resolve a conflict of mapping (output of the *SolveConflict*) is itself mapped to another itemset. In the case of a conflict loop, we continue to call the function *SolveConflict* until its output (new proposed candidate) is not already mapped. In each loop, we exclude the proposed candidate which is already mapped. In the situation where there is not any candidate for one of the itemsets in conflict (*i.e.* $\text{Seq}_1(i)$ and $\text{Seq}_1(k)$), we associate the initial candidate ($\text{Seq}_2(j)$) with the itemset owning the highest weight of mapping. The other itemset remains without any corresponding itemset in the 2^{nd} sequence (without mapping).

Output of Step 1. At the end of step 1, we have the final mappings. We come up with the mappings associating each itemset from sequence 1 with the more relevant (in terms of common and non-common items) itemset from sequence 2. These mappings are stored in a list named *mapOrder*. The first element (resp. $2^{\text{nd}}, \dots, n^{\text{th}}$) in the list is the timeStamp of the itemset from sequence 2 corresponding to the first itemset (resp. $2^{\text{nd}}, \dots, n^{\text{th}}$) from

³The *timeStamp* in our context, corresponds to the position of itemsets in the order of sequence. For example in the sequence (a)(ab)(c), $ts(ab) = 2$.

sequence 1. We thus have:

$$\text{mapOrder} = \{t_1, t_2, \dots, t_i, \dots, t_n\}$$

t_i = the timeStamp of the itemset of the Seq_2 mapped with i^{th} itemset in Seq_1 .

At last, we calculate the *mapping score* by average of weight of mappings (*AveWeightScore*).

We explain here why the content of itemsets, which have cross-mappings, should be considered into the mapping score. In fact, if we consider only the weight of itemsets, which have ordered mappings, we lose some information about the similarity of sequences. For instance, when comparing (a)(b) with (b)(a) and (a)(b) with (b)(d), if we take only the weight of itemsets with ordered mapping (*i.e.* the weight of (b)→(b)), then (a)(b) and (b)(a) will be treated as same as (a)(b) and (b)(d). But, by considering the weight of all found mappings, we consider also the weight of (a)→(a) (*equal to 1*) and the weight of (a)→(d) (*equal to 0*).

We now go to Step 2, which aims at evaluating to what extent the mappings found in Step 1 respect the order of the sequences.

Step 2: Order Score Calculation This step is two-folded:

- *totalOrder*,
- *positionOrder*.

Firstly, we aim at discarding cross-mappings (*cf.* Figure 2) using *totalOrder*. *mapOrder* is a list of integers representing the timeStamps of itemsets. In this list, as while as the integers (timeStamps) are increasing, the corresponding mappings are totally ordered (there are not cross-mappings). We look hence for the maximum increasing subsequences of *mapOrder* to find all the possible series of mappings which comply with order. In this way, we avoid the cross-mappings.

More precisely, *totalOrder* measures the percentage of non-cross mappings (*i.e.* mappings respecting the order of itemsets in the two sequences). The calculation of *totalOrder* is formalized as follow:

$$\text{totalOrder} = \frac{\text{nbOrderedItemSets}}{\text{aveNbItemSets}}$$

nbOrderedItemSets = The number of itemsets in the increasing subsequence
aveNbItemSets = The average of the number of all itemsets in the two sequences

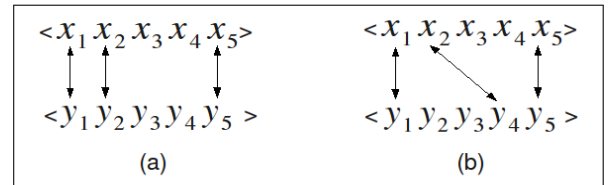


Figure 3: Distance between mappings according to positions of itemsets.

Secondly, we aim at considering the width between the mapped itemsets using *positionOrder*. For instance, Figure 3 shows that the itemsets can be mapped differently. Here, in part “a” of Figure 3, the itemsets are mapped in a closer way compared to part “b”.

The timeStamps also allows us to check similarities of mapped itemsets according to their positions in the sequences.

positionOrder shows if the distance between two successive mappings based on the positions of itemsets in

the sequence 1 is equal to that according to the positions of itemsets in the sequence 2. Figure 3-a shows an example where the distance between the successive mappings according to the positions of itemsets in the sequence 1 is equal to that according to the positions of itemsets in the sequence 2. But in Figure 3-b, the distance between two first mappings depending on the positions of x_1 and x_2 is not equal to the distance depending on the positions of y_1 and y_2 . The *positionOrder* formula is:

positionOrder =

$$\sum_{i=1}^{|sub|} \frac{|sub(i) - sub(i-1)| - |mapOrder^{-1}(sub(i)) - mapOrder^{-1}(sub(i-1))|}{aveNbItemSets}$$

$sub(i)$ = The value of i^{th} position in the subsequence

$mapOrder^{-1}(x)$ = The position of itemset "x" in *mapOrder*

At last, for each increasing subsequence of *mapOrder*, we calculate the multiplication of *totalOrder* and *positionOrder* and we keep the highest score as (*orderScore*):

$$orderScore = \max\{totalOrder(sub) \times (1 - positionOrder(sub))\} \\ sub \in \{maximum\ increasing\ subsequences\ of\ mapOrder\}$$

At the end of Step 2, the *orderScore* is calculated. The final similarity degree can now be computed by an aggregation between order score (*orderScore*) and the average of weight of mappings (*AveWeightScore*) calculated at the end of Step 1. This final degree is calculated in Step 3 as described below.

Step 3: Final Similarity Degree Calculation. We calculate the similarity degree *SimDegree* as an aggregation between the *orderScore* and the average weight of mapping of itemsets *AveWeightScore*. The *orderScore* compares the similarity of two sequences based on the positions of itemsets (likeness of order of itemsets) in sequences. By *AveWeightScore*, we compare the similarity in terms of common items and non-common items in itemsets. The aggregation can be a weighted average while we can define the coefficient for each score. By defining the coefficient for each score, we choose to consider the order as more (or less or as) important than the content according to the application context. Our measure is thus a very flexible measure.

SimDegree =

$$\frac{(orderScore \times Co_1) + (AveWeightScore \times Co_2)}{Co_1 + Co_2}$$

The calculation of the similarity is explained in the following example. We tried to treat most cases and also the conflicts of mapping in the illustration.

Example 4.1. Let us take $M_1 = \{(bc)(df)(e)\}$ and $M_2 = \{(abc)(mn)(de)(egh)(fg)\}$ that we used to show disadvantages of *LCS* as the two sequential patterns.

Step 1: Mapping Score Calculation. For each itemset $M_1(i)$ in the first sequence, we are looking for the most similar itemset $M_2(j)$ in the 2nd sequence. This is done by the weight of mapping calculations.

$$\begin{aligned} Weight(M_1(1), M_2(1)) &\Rightarrow Weight((bc), (abc)) = \frac{2}{3+2} = 0.8 \\ Weight(M_1(1), M_2(2)) &\Rightarrow Weight((bc), (mn)) = \frac{0}{2+2} = 0 \\ Weight(M_1(1), M_2(3)) &\Rightarrow Weight((bc), ((de))) = \frac{0}{2+2} = 0 \\ Weight(M_1(1), M_2(4)) &\Rightarrow Weight((bc), ((egh))) = \frac{0}{2+3} = 0 \\ Weight(M_1(1), M_2(5)) &\Rightarrow Weight((bc), ((fg))) = \frac{0}{2+2} = 0 \end{aligned}$$

We choose the case with the highest weight: *MappedItemSets.put*($M_1(1)$, $M_2(1)$).

We continue by the following itemset $M_1(2)$:

$$\begin{aligned} Weight(M_1(2), M_2(1)) &\Rightarrow Weight((df), (abc)) = \frac{0}{3+2} = 0 \\ Weight(M_1(2), M_2(2)) &\Rightarrow Weight((df), (mn)) = \frac{0}{2+2} = 0 \\ Weight(M_1(2), M_2(3)) &\Rightarrow Weight((df), ((de))) = \frac{1}{2+2} = 0.5 \\ Weight(M_1(2), M_2(4)) &\Rightarrow Weight((df), ((egh))) = \frac{0}{2+3} = 0 \\ Weight(M_1(2), M_2(5)) &\Rightarrow Weight((df), ((fg))) = \frac{1}{2+2} = 0.5 \end{aligned}$$

The *Weight*((df), (de)) and the *Weight*((df), (fg)) are equal, so we select the itemset with lower timeStamp (i.e. (de)) to map with the itemset (df). We have thus: *MappedItemSets.put*($M_1(2)$, $M_2(3)$).

We do the same for the 3rd itemset $M_1(3)$:

$$\begin{aligned} Weight(M_1(3), M_2(1)) &\Rightarrow Weight((e), (abc)) = \frac{0}{3+2} = 0 \\ Weight(M_1(3), M_2(2)) &\Rightarrow Weight((e), (mn)) = \frac{0}{2+2} = 0 \\ Weight(M_1(3), M_2(3)) &\Rightarrow Weight((e), ((de))) = \frac{1}{1+2} = 0.6 \\ Weight(M_1(3), M_2(4)) &\Rightarrow Weight((e), ((egh))) = \frac{1}{1+3} = 0.5 \\ Weight(M_1(3), M_2(5)) &\Rightarrow Weight((e), ((fg))) = \frac{1}{2+2} = 0 \end{aligned}$$

According to the calculation, we select the itemset $M_2(3)$ (i.e. (de)). But this itemset (de) has already been associated with the itemset (df) of M_1 . Therefore, we use the function of conflict resolving.

We look for new candidates in M_2 for itemsets in conflict ((df) and (e)) before and after the current candidate itemset (de). We get the following candidates:

- **for the itemset (df):**
 $nextMaxBefore_{(df)} = \emptyset$
 $nextMaxAfter_{(df)} = M_2(5) = (fg)$
- **for the itemset (e):**
 $nextMaxBefore_{(e)} = \emptyset$
 $nextMaxAfter_{(e)} = M_2(4) = (egh)$
- **Possible pairs of mappings:**
 $\langle((df), (de)), ((e), (egh))\rangle$
 $\langle((e), (de)), ((df), (fg))\rangle$

Using the weight of mapping and considering case of cross-mapping ($\langle((e), (de)), ((df), (fg))\rangle$), we get:

$$\begin{aligned} localSim(\langle((df), (de)), ((e), (egh))\rangle) &= \frac{0.5+0.5}{2} = 0.5 \\ localSim(\langle((e), (de)), ((df), (fg))\rangle) &= \frac{1}{2} \times \frac{0.6+0.5}{2} = 0.27 \end{aligned}$$

We select the pair having the highest *localSim*: $\langle((df), (de)), ((e), (egh))\rangle$. The itemset (df) is thus mapped with (de) and the itemset (e) with (egh):

MappedItemSets.put($M_1(2)$, $M_2(3)$)
MappedItemSets.put($M_1(3)$, $M_2(4)$).

Final mappings are:

$$\begin{aligned} \langle M_1(1) = (abc), M_2(1) = (ab) \rangle \\ \langle M_1(2) = (df), M_2(3) = (de) \rangle \\ \langle M_1(3) = (e), M_2(4) = (egh) \rangle \end{aligned}$$

We create now the *mapOrder* list. We put at the i^{th} place in *mapOrder* the timeStamp of itemset mapped to

i^{th} itemset $Seq_1(i)$ of the first sequence. Hence, the 1st place in $mapOrder$ is taken with “1” according to the timeStamp of the itemset $M_2(1)$ mapped with the 1st itemset $M_1(1)$ of the first sequence. For the 2nd place, the timeStamp of the itemset $Seq_2(3)$ mapped with the 2nd itemset $M_1(2)$ of the first sequence (*i.e.* “3”) and in the same manner we put “4” in the 3th place in the $mapOrder$. Therefore the $mapOrder$ is:

$$mapOrder = \{1, 3, 4\}$$

At last, we calculate the mapping score by averaging the weight of mappings (*AveWeightScore*).

$$AveWeightScore = \frac{Weight((bc),(abc)) + Weight((df),(de)) + Weight((e),(egh))}{3} = \frac{0.8 + 0.5 + 0.5}{3} = 0.6$$

$$AveWeightScore = 0.6$$

Step 2: Order Score Calculation. In this step the aim is to compare the order of itemsets in the two sequences. We seek all maximum increasing subsequences of $mapOrder$ (output of step 1). In this Example, there is only one maximum increasing subsequence.

The only maximum increasing subsequence of $mapOrder$:

- $subseq = \{1, 3, 4\}$

According to the formula of $totalOrder$, of $positionOrder$ and of $orderScore$:

- $totalOrder((1,3,4)) = \frac{3}{(3+5)/2} = 0.75$
- $positionOrder((1,3,4)) = \frac{|(3-1)-(2-1)|}{(3+5)/2} + \frac{|(4-3)-(3-2)|}{(3+5)/2} = 0.25$
- $orderScore = 0.75 \times (1 - 0.25) = 0.56$

Step 3: Similarity Degree Calculation. With the multiplication of $orderScore$ and the $AveWeightScore$, we get the degree of similarity between the two sequential patterns:

$$SimDegree = 0.56 \times 0.6 = 33\%$$

5 Experiments

In this paper, we introduce a measure of similarity for sequential patterns. In this section, we report the experiments led to show the accuracy, the relevance and the scalability of our approach. A measure of similarity must capture the similarity of compared items. Such a measure is usually used within another algorithm like as clustering or extraction of sequential patterns under similarity constraint. It must be efficient and scalable. We consider two main directions :

- the accuracy of the similarity degree obtained by S^2MP ,
- the efficiency of the S^2MP algorithm at execution time and size of used memory.

Accuracy of S^2MP . We experiment S^2MP to assess its quality (*accuracy*) and compare the results obtained by S^2MP and Edit distance. We apply two clusterings of sequential patterns: one with S^2MP and one with Edit Distance. In both cases we use the same dataset. To compare clusters obtained by each measure, we calculate

the entropy of each cluster.

The dataset consists of 100 sequential patterns. We manually create 10 categories of sequential patterns. In each category, we put the similar patterns. These categories will be used as references. The sequential patterns have different sizes. Among these 10 reference categories, 4 categories contains different patterns and 6 categories contain patterns similar to the patterns of at least one other category. This allows us to assess the accuracy of each measure when it comes to distinguish clusters with a small inter-cluster distance.

We adopted the K-means clustering for sequential patterns. We cluster the patterns at first by using S^2MP and then by using Edit distance. For each clustering, we calculate the entropy of obtained clusters. We compare also the clusters with reference categories for calculating the precision and recall of each clustering. These experiments show that the cluster obtained with S^2MP are more homogeneous (*according to entropy of clusters*) than those obtained with Edit distance. Moreover, the clusters obtained by S^2MP are more accurate (*according to precision of clusters*) and more complete (*according to recall of clusters*).

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
S^2MP	0.98	0	0.99	0.86	0.95	0	0.97	0.95	0.65	0
Edit dist	0.97	0	0.99	1.20	0.89	0	0.98	0.98	0.70	0.99

Table 1: Entropy of clusters obtained by S^2MP and Edit distance.

The table 1 shows the entropy of clusters obtained with each measure. More entropy of a cluster is small, more cluster is homogeneous and it contains more informations. The average entropy for clustering with S^2MP is 0.63 and using Edit distance is 0.77. The precision and recall of clusters obtained by each measure is illustrated in table 2. The precision and recall are calculated based on the reference categories.

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Precision (S^2MP)	0.57	1	0.53	0.71	0.65	1	0.5	0.6	0.68	1
Recall (S^2MP)	0.4	1	0.7	1	1	0.5	0.5	0.4	1	0.4
Precision (Edit)	0.6	1	0.53	0.58	0.68	1	0.4	0.62	0.83	0.57
Recall (Edit)	0.3	1	0.6	1	0.9	0.8	0.2	0.5	1	0.4

Table 2: Precision and recall of clusters using S^2MP and using Edit distance.

We also experiment S^2MP and Edit distance on their ability to identify similar sequential patterns in different contexts. We consider the bioinformatics domain and more precisely the characteristics of sequential patterns extracted from DNA chips. In this area, according to experts, the contents of itemsets (*i.e.* *items*) are more important than the order of itemsets. For example, the two sequential patterns $M_1 = \langle (G_1, G_2)(G_3)(G_4) \rangle$, $M_2 = \langle (G_3)(G_1, G_2)(G_4) \rangle$ are so similar because the content of itemsets are similar however their order are not. To experiment S^2MP in this situation, we manually create 10 categories each one contains 10 similar sequential patterns according to the content of their itemsets, which are ordered so differently in different sequences (*i.e.* *each category contains 10 sequential patterns, which are similar based in the content of itemsets but order of itemsets differs*). We also consider the categories, which contain the patterns rather similar.

We do a clustering on the data set with S^2MP by giving to the score of mapping a weight two times more than the weight of order score (*i.e.* *we configure S^2MP in the way that the content of itemsets is more important than the order of itemsets*). Then, we do other clustering on this

dataset with Edit distance. The results with S^2MP show that we can capture similar patterns according to the particular definition of similarity in this context. This shows that S^2MP is well parametrizable and is adaptable to different definition of similarity for sequential patterns. Results obtained with Edit distance in this context, are not satisfactory.

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
S^2MP	0.99	0.52	0.99	0.99	0.99	0	0	0.99	0	0.98
Edit dist	1.2	1.2	1.3	1.3	1.3	1.2	1.4	0.95	0.4	1.7

Table 3: Entropy of clusters obtained by S^2MP and Edit distance when the contents of itemsets are more important than the order of itemsets – (e.g. patterns extracted from the DNA chips data)

Table 3 shows the relevance of S^2MP in this context and its adaptability to different definition of similarity for sequential patterns. On this dataset, the average entropy of clustering using S^2MP is 0.64 and using Edit distance is 1.19. We demonstrate the precision and recall of clusters for each clustering in the table 4. The precision and recall of clusters are calculated according to the reference categories.

Efficiency of S^2MP . Despite the complex appearance of our measure's algorithm, we show that our method is very efficient in terms of runtime and size of memory used, by studying how it performs depending on three factors, as detailed below. We test execution time and size of memory used by our similarity measure in three directions: (1) depending on the number of itemsets in sequential patterns(2) depending on the number of items in sequential patterns and finally (3) depending on the number of sequential patterns that we want to calculate their similarities.

We create a matrix of similarity with $(n \times n)$ dimensions where n represents the number of sequential patterns. Our measure of similarity is not symmetrical, we calculate thus all the matrix instead of a diagonal calculation. The time for calculating the similarity matrix is the time necessary to make $n \times n$ comparisons of similarity. Our results show that our measure of similarity is calculated very quickly even when there are many conflict loops at the mapping phase.

The experiments are performed on a machine with a 2GHz Intel CPU with 2GB of RAM under the ubuntu Linux operating system. Our algorithm is implemented using Java 5.

Data set. We carry out experiments on two different types of itemset sequences:

1. frequent sequential patterns,
2. data sequences.

The frequent sequential patterns are extracted from synthetic data generated by the generator IBM quest⁴. In the second stage of our experimentation, we decided to make a test on data sequences because in such sequences, we are more likely to have conflicts of mapping. This allows us to study the impact of conflict on runtimes. In the tests depending on the number of itemsets and items, data sets are made up of 1000 sequential patterns (or data sequences). At each stage, we calculate the similarity matrix, (*i.e.* we realize 1,000,000 comparisons of similarity).

⁴ www.almaden.ibm.com/cs/projects/iis/hdb/Projects/data-mining/datasets/syndata.html

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Precision (S^2MP)	0.55	0.71	0.54	0.54	0.55	1	1	0.53	1	0.57
Recall (S^2MP)	0.5	1	0.6	0.6	0.5	0.6	1	0.7	1	0.4
Precision (Edit)	0.52	0.61	0.46	0.60	0.50	0.5	0.5	0.37	0.9	0.16
Recall (Edit)	0.9	0.6	0.6	0.3	0.5	0.6	0.7	0.3	0.9	0.2

Table 4: Precision and recall of clusters using S^2MP and using Edit distance when the contents of itemsets are more important than the order of itemsets – (e.g. patterns extracted from the DNA chips data)

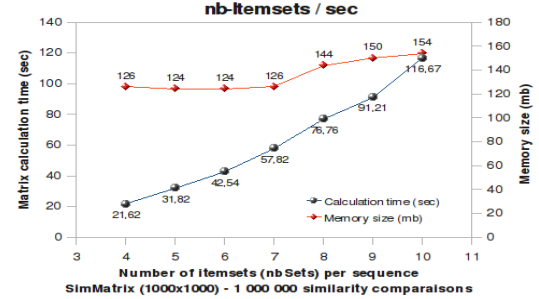


Figure 4: Calculation time and memory size depending on the number of itemsets.

Results. Figures 4 and 5 represent the evolution of the running time of 1,000,000 comparisons and the size of memory used according to the number of itemsets and items per sequence. According to the curves, the size of memory used does not change significantly when the number of itemsets (or items) per sequence increases. The time for calculating the similarity matrix (*1,000,000 comparisons*) when there are 10 itemsets per sequence is satisfactory (116 sec). This means that the time for calculating similarity between two sequential patterns, each one with 10 itemsets is equal to 116μ sec. Our experiments show that the number of items per sequence does not affect the runtime as much as the number of itemsets per sequence.

We show the time of calculating similarity matrix on the Figure 6 and the size of memory used on Figure 7 when the number of sequences increases. In each case, there is $n \times n$ similarity comparisons where n is the number of sequences in the data set. We run this test on three types of sequences: the sequences with 5 itemsets, with 7 itemsets and sequences with 9 itemsets. In cases where there are 5000 sequences (*i.e.* 25,000,000 similarity comparisons), and each sequence contains 9 itemsets, the execution time is only 1974 sec.

Figure 8 shows the results of experimentation on data sequences. For each case, we noted the number of resolved conflicts when calculating the similarity matrix. The X-axis represents the different data sets. For each, the number of itemsets and the average number of items per sequence are marked. There are 1000 sequences in each data set (thus 1,000,000 similarity comparisons). The curves represent the running time of the calculating similarity matrix for each case and the size of used memory. For example, where there are 20 itemsets and on average 109 items per sequence and 103437 solved conflicts, the calculating time of 1,000,000 similarity comparisons is equal to 961 sec. We note that the size of memory used is almost constant.

With these experiments, we have shown that our measure is efficient in term of runtime and size of memory for data sequences and frequent sequential patterns.

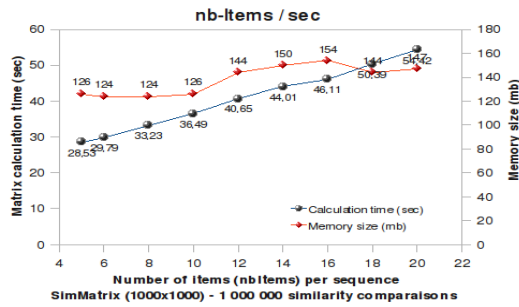


Figure 5: Calculation time and memory size depending on the number of items.

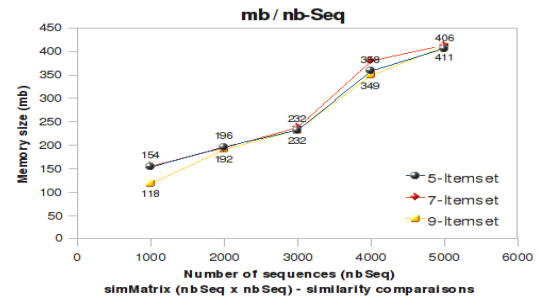


Figure 7: Size of memory used for calculating the similarity matrix based on the number of sequences.

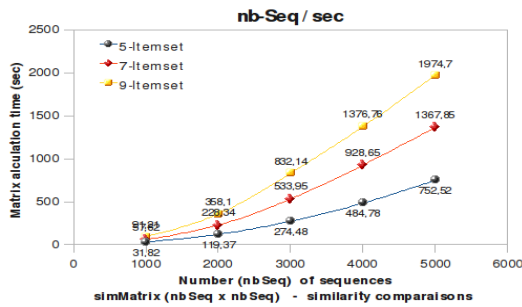


Figure 6: Time for calculating the similarity matrix based on the number of sequences.

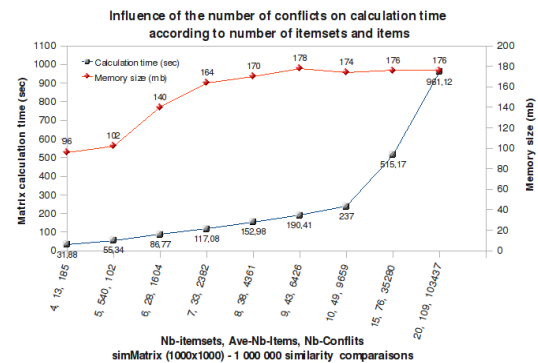


Figure 8: Influence of conflicts on the runtime of calculating the similarity

6 Conclusion

In this paper, we have defined a similarity measure (S^2MP) adapted to sequential patterns taking into account all the characteristics of sequential patterns and in particular their semantics. The degree of similarity is the result of the aggregation of two scores. These scores measure the similarity of sequential patterns in terms of itemsets and their positions in the sequences (*orderScore*) but also in terms of items contained in the corresponding itemsets (*aveWeightScore*). The combination of two independent scores allows a modular measurement. It is therefore adaptable and parametrizable depending on the context, different definition of similarity and the meaning of itemset in the application domain. S^2MP overcomes the disadvantages of *LCS* and *Edit distance* in the case of sequential patterns.

Experiments show that S^2MP is more accurate than Edit distance. The clusters obtained by S^2MP are more precise and more complete than the clusters obtained by Edit distance. The experiments show also that S^2MP can be calculated very quickly even when we compare many sequences with several itemsets.

Several areas and methods as the clustering of sequential patterns, outliers detection, extraction of sequential patterns under similarity constraint, compression of sequential patterns, visualisation and querying the sequential patterns, etc are possible applications of S^2MP .

References

- Agrawal, R., Faloutsos, C. & Swami, A. N. (1993), Efficient Similarity Search In Sequence Databases, in D. Lomet, ed., 'Proceedings of the 4th International Conference of Foundations of Data Organization and Algorithms (FODO)', Springer Verlag, Chicago, Illinois, pp. 69–84.
- Agrawal, R. & Srikant, R. (1995), Mining sequential patterns, in P. S. Yu & A. S. P. Chen, eds, 'Eleventh Inter-

national Conference on Data Engineering', IEEE Computer Society Press, Taipei, Taiwan, pp. 3–14.

- Bozkaya, T., Yazdani, N. & Ozsoyoglu, Z. M. (1997), Matching and indexing sequences of different lengths, in 'CIKM', pp. 128–135.

- Capelle, M., Masson, C. & Boulicaut, J.-F. (2002), Mining frequent sequential patterns under a similarity constraint, in 'IDEAL', pp. 1–6.

- Garofalakis, M. N., Rastogi, R. & Shim, K. (1999), SPIRIT: Sequential pattern mining with regular expression constraints, in 'The VLDB Journal', pp. 223–234.

- Guralnik, V. & Karypis, G. (2001), A scalable algorithm for clustering sequential data, in 'ICDM '01: Proceedings of the 2001 IEEE International Conference on Data Mining', IEEE Computer Society, Washington, DC, USA, pp. 179–186.

- Hartigan, J. (1975), *Clustering Algorithms*, John Wiley and Sons, Inc.

- Jianhua Zhu, Z. W. (2005), Fast: A novel protein structure alignment algorithm, Vol. 58, Bioinformatics Program, Boston University, Boston, Massachusetts; Biomedical Engineering Department, Boston University, Boston, Massachusetts.

- Kum, H.-C. (2004), Approximate Mining of Consensus Sequential Patterns, PhD thesis, University of North Carolina.

- Kum, H.-C., Pei, J., Wang, W. & Duncan, D. (2003), Approxmap: Approximate mining of consensus sequential patterns, in 'SDM'.

- Lee, K.-H., Choy, Y.-C. & Cho, S.-B. (2004), 'An efficient algorithm to compute differences between structured documents', *Knowledge and Data Engineering, IEEE Transactions on* **16**(8), 965–979.

- Levenshtein, V. I. (1966), Binary codes capable of correcting deletions, insertions, and reversals, Technical Report 8.
- Mannila, H. & Ronkainen, P. (1997), Similarity of event sequences, *in* 'TIME '97: Proceedings of the 4th International Workshop on Temporal Representation and Reasoning (TIME '97)', IEEE Computer Society, Washington, DC, USA, p. 136.
- Moen, P. (2000), Attributs, Event Sequence, and Event Type Similarity Notions for Data Mining, PhD thesis, University of Helsinki, Finland.
- Morzy, T., Wojciechowski, M. & Zakrzewicz, M. (1999), Pattern-oriented hierachical clustering, *in* 'Advances in Databases and Information Systems', pp. 179–190.
- Pei, J., Han, J., Mortazavi-asl, B., Pinto, H., Chen, Q., Dayal, U. & chun Hsu, M. (2001), Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth, pp. 215–224.
- Plantevit, M., Goutier, S., Guisnel, F., Laurent, A. & Teisseire, M. (2007), Mining unexpected multidimensional rules, *in* 'DOLAP '07: Proceedings of the ACM tenth international workshop on Data warehousing and OLAP', pp. 89–96.
- Sequeira, K. & Zaki, M. J. (2002), Admit: anomaly-based data mining for intrusions, *in* 'KDD', pp. 386–395.
- Tversky, A. (1977), *Psychological Review* (84), 327–352.

Mining medical specialist billing patterns for health service management

Yin Shan¹, David Jeacocke, D. Wayne Murray, Alison Sutinen

Program Review Division, Medicare Australia
134 Reed St. North, Tuggeranong
ACT 2900, Australia

{Yin.Shan, David.Jeacocke, Wayne.Murray, Alison.Sutinen}@medicareaustralia.gov.au

Abstract

This paper presents an application of association rule mining in compliance in the context of health service management. There are approximately 500 million transactions processed by Medicare Australia each year. Among these transactions, there exist a small proportion of suspicious claims. This study applied association rule mining to examine billing patterns within a particular specialist group to detect these suspicious claims and potential fraudulent individuals. This work identified both positive and negative association rules from specialist billing records. All the rules identified were examined by a subject matter expert, a practicing clinician, to classify them into two groups, those representing compliant patterns and non-compliant patterns. The rules representing compliant patterns were then used to detect potentially fraudulent claims by examining whether claims are consistent with these rules. The individuals whose claims frequently break these rules are identified as potentially high risk. Due to the difficulty of direct assessment on high risk individuals, the relevance of this method to fraud detection is validated by analysis of the individual specialist's compliance history. The results clearly demonstrate that association rule mining is an effective method of identifying suspicious billing patterns by medical specialists and therefore is a valuable tool in fraud detection for health service management.

Keywords: association rule, negative association rule, health data mining, fraud detection, open source data mining.

1 Introduction

There has been an increasing interest in mining health service management data (Becker, Kessler and McClellan, 2005, Lin *et. al.*, 2008, Yang and Hwang, 2006). This is partially due to the fact that public health systems in many countries have consumed a significant portion of governments' expenditure and can be subject to abuse. At the same time, it provides an extremely rich dataset and many challenging research questions, such as

detecting fraudulent practice, or inappropriate billing, to facilitate more efficient use of the resources. In Australia, a government agency, Medicare Australia, administers Medicare, a fee for service national health funding system for Australians. It is also responsible for undertaking reviews to ensure the integrity of associated health programs administered under Medicare. There have been a series of studies that have applied a range of data mining techniques to the Medicare Australia data for various compliance purposes, such as applying Neural Networks and Boosted Regression Trees to detect fraudulent behaviour by general practitioners (Pearson, Murray and Mettenmeyer 2005), K-nearest neighbour method for fraud detection (He, Graco and Yao 1999) and positive association rule mining to better understand medical practice patterns (Semenova, 2004).

Although there have been an increasing number of applications of association rule mining, they usually focus on positive rules and discovering common patterns. There is very limited research on association rule mining in detecting anomalous patterns for compliance purposes in the medical service domain. This study applied association rule mining for fraud detection in a specialist population. In addition to conventional positive rule mining, negative association rule mining was also applied in this study. We were able to show negative association mining rules to be particularly useful in this application and possibly other fraud detection problems.

Rules describe typical patterns of practice, which may reflect compliant or non-complaint patterns. All rules must therefore be evaluated by a subject matter expert to determine their relevance. Rules representing non-compliance may imply some commonly entrenched incorrect specialist billing practices. This is very useful in improving compliance, in particular, it may assist with clarifying billing regulations, for example through targeted educational interventions.

Rules representing compliant practices are valuable to identify specialists who may not be practicing in accordance with their peers and may therefore be billing either inappropriately or fraudulently. Individual specialists, whose claims frequently break these rules, may present a high risk of inappropriate or fraudulent billing patterns. It would be impossible to definitively validate whether specialists identified by association rules were engaged in inappropriate practices, without a comprehensive review of their billing practices being undertaken by a panel of peers. Thus an indirect approach to validation was undertaken in this study. The

Copyright (c)2008, Australian Computer Society, Inc. This paper appeared at conference Seventh Australasian Data Mining Conference (AusDM 2008), Glenelg, Australia. Conferences in Research and Practice in Information Technology, Vol. 87. John F. Roddick, Jiuyong Li, Peter Christen and Paul Kennedy, Eds. Reproduction for academic, not-for profit purposes permitted provided this text is included.

effectiveness of association rule mining to detect fraud or inappropriate practice was evaluated by analysing specialists' compliance history. We were able to show that with this effectiveness measure, association rule mining was a valid and promising method of identifying potentially fraudulent billing patterns.

The remaining sections of this paper are organised as follows. The problem domain is briefly introduced in Section 2. Section 3 provides a short description of positive and negative association rule mining. Experiments are reported in Section 4. The evaluations of the results numerically and by subject matter experts are covered in Section 5. Section 6 presents the conclusions and future research.

2 Specialist Billing Patterns

Specialists claim a significant portion of Medicare benefits. There are dozens of small specialist groups based on their specialties, consisting of tens to hundreds of specialists each. In contrast General Practitioners (GPs) as one professional group are much larger in number (over 25,000 nation-wide) and exhibit relatively much less variation in their practice patterns. Because of their unique practice styles and small group sizes, specialist groups impose interesting challenges to automatic fraud detection approaches.

One of the main compliance tasks in specialist groups is to ensure specialists bill items according to the intent specified under the Medicare system. Ideally, billing patterns may be identified as anomalous through a clear difference in the pattern of services rendered by other specialists in the same specialty group. The discovery of anomalous billing patterns may identify a range of issues from fraud, inappropriate billing to billing arising from new technologies and procedures. In each case, the billing pattern discovered may assist Medicare Australia to determine the effective advice to policy makers and compliance response to ensure the integrity of its programs. If these patterns can be identified in time and correctly, the response can be made to benefit both Medicare in reducing inappropriate payments and sometimes to the profession to determine areas of the Medicare Benefits Schedule (MBS) (Australian Government, 2007) requiring clarification or new Medicare items.

3 Association Rules

Association rule mining (Agrawal, Imielinski, and Swami, 1993, Agrawal and Srikant, 1994) has drawn a lot of attention because of its effectiveness and intuitive representation. This has resulted in many efficient algorithms being proposed. For completeness in this paper, we only present a brief description. More details can be found in other literature sources (Agrawal, Imielinski, and Swami, 1993).

Let $I = \{i_1, i_2, \dots, i_m\}$ be a set of literals, called items. Let D be a set of transactions, where each transaction T is a set of items such that $T \subseteq I$. A transaction T contains X if $X \subseteq T$. An association rule is in the form $X \rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \emptyset$. The rule $X \rightarrow Y$ holds in the transaction set D with confidence c if $c\%$ of transactions in D that contain X also contain Y . The rule $X \rightarrow Y$ has

support s in the transaction set D if $s\%$ of transactions in D contain $X \cup Y$.

The association rule defined above, describing the presence of items, can't completely meet our needs. In the compliance domains, it is natural to ask complementary questions. For example, one common non-compliant billing practice is to bill some additional items in conjunction with commonly billed items for the same service. If a rule can tell us what should *not* be billed with other commonly billed items, such non-compliant "add-on" billing can be easily identified. This type of association rules which can describe the absence of items are called negative association rules (Savasere, Omiecinski and Navathe 1998, Zhang and Zhang 2002).

To avoid confusion, the previously defined association rules are called positive association rules. Negative association rules are in the form $X \rightarrow \neg Y$, which can be interpreted as that if X is present, it is unlikely that Y would be present too. Because the formal full definition of negative association rule is similar to that of the positive ones, it is omitted here. Negative association rules also have similar measures of confidence and support as in for the positive rules and therefore they aren't reiterated here.

4 Experiments

The data set used in this study was drawn from Medicare Australia's Enterprise Data Warehouse, covering billing records of a specialist group for the second quarter of 2007 (1 April, 2007 – 30 June, 2007 inclusive). The data was organised in episodes which were defined as all the items claimed or billed for one patient on one day by one specialist. Obviously, an episode corresponds to a transaction in the context of association rule mining. This data set contained 63010 episodes (transactions). Removing those episodes which contained only one item, resulted in 32476 episodes deemed suitable for further study.

Because of the nature of the specialist practice examined, there were as many as 620 procedural MBS items presented in this dataset. This results in as many as 7989 unique billing episodes because of the distinctive needs of each individual patient. After conducting a series of empirical studies, we determined that the settings of 80% confidence and 0.1% support produced optimal results.

In total, 215 association rules, including both positive and negative rules, were identified. These rules were presented to the subject matter expert for evaluation and specialists were checked against these rules to study the effectiveness of the method in identifying potential non-compliant individuals.

The association rule discovery was undertaken using the Christian Borgelt, Artamonova and others' open source implementation (Borgelt and Kruse 2002, Artamonova, Frishman and Frishman, 2007) of Apriori (Agrawal, Imielinski and Swami 1993)

5 Evaluation

There were three components to the evaluation undertaken. The first involved examination of each of the rules identified by the subject matter expert, to determine

whether the services reflected by the items in one rule might conceivably be billed together. Thus, all rules were examined to assess their clinical relevance. While only a single subject matter expert was utilised in this preliminary study, consideration was given to whether the items could be appropriately billed under Medicare rather than whether they considered the management approach was consistent with their own practice.

The second aspect of the evaluation involved the comparison of an individual specialist against rules representing compliant patterns, to determine whether billing patterns of the specialists observed were likely to reflect non-compliant billing patterns and the number of occasions on which this occurred was recorded. The specialists who broke rules on a great number of occasions were identified as high risk.

Finally the compliance histories of the specialists identified as high risk were examined. This provides us with a good indication of the effectiveness of association rules in identifying inappropriate or fraudulent claims patterns.

5.1 Rule evaluation

In total, 215 rules were identified, including 192 negative rules and 23 positive rules. It was not surprising that more negative rules were discovered because for positive rule discovery only the presence of items are considered while for the negative rule discovery both presence and the absence of the items are considered. Another observation was that although there were over 20 positive rules identified, there were only a very small number of unique items involved.

The negative rules were much stronger than positive rules in terms of confidence. The minimum confidence of negative rules was 95.95% while it was only 80.25% for positive rules. This fits with the clinical context, based on the description of items in the Medicare Benefit Schedule, where some services were explicitly stated that they should not be billed together on the same day. So the negative rules consistent with these patterns should be very strong. Another example this study highlighted, was that there were several very strong negative rules, indicating procedural items should not be billed on the same day with an initial attendance. Clinical observation tells us it is extremely unlikely that a specialist would perform procedures on the very first consultation with the patient. Thus, it is not surprising these negative rules describing these clinic scenarios are very strong.

On the other hand, the clinic scenarios indicated by positive rules are not as definitive as those by negative rules. For positive rules, close examination reveals that these rules themselves, not violations of them, can represent inappropriate billing. However, it is also possible that these rules may reflect new patterns of billing by specialists, possibly related to a new specialist technology or technique, resulting in a small number of specialists starting billing differently from their peers.

5.1.1 Common compliant patterns

Negative rules represented common patterns that were generally considered consistent with the billing rules covered under the Medicare Benefit Schedule. It is very

encouraging that some of negative rules correspond very well to some unusual combinations in this specialist group, which have been alerted to Medicare Australia's clinical experts by other sources.

The subject matter experts found it was more intuitive to interpret the negative rules and the implication of their violation than those of positive rules. The subject matter expert concluded that violations of these rules are likely to be good indications of non-compliant billing. Violations of these negative rules suggest billing additional items not normally billed by the majority of specialists. Those specialists who violate these rules frequently are thus markedly different from their peers. Therefore, concerns may need to be raised regarding the appropriateness of the services provided by these specialists.

Of the 192 negative rules identified, 30 rules had a confidence value of 1.0, which was considered neither numerically interesting, nor in the opinion of the subject matter expert, as being useful for compliance purposes. These rules were thus removed. For the remaining 162 negative rules, the subject matter expert classified them into three groups based on the likelihood inappropriate billing (see Table 1). High rating indicates the rules make strong sense in the domain. If any of these rules was broken, it was almost certain to suggest an incorrect billing to Medicare Australia. The low rating indicates that although breaking these rules might be inappropriate other appropriate billing explanations may also exist. In other words, low rating rules might not be sufficient for detecting inappropriate billing. It is worth mentioning although these low rating rules may not be sufficient for direct identification of inappropriate billing, they still provide valuable information on profiling specialists for related compliance activities. As can be seen from Table 1, more than half of the rules (56.18%) discovered, comprising high and medium rate rules, are regarded as suitable for detecting inappropriate billing.

Rating	Number of Rules
High	53 (32.72%)
Medium	38 (23.46%)
Low	71 (43.83%)

Table 1: The risk rating of the negative rules

5.1.2 Common non-compliant patterns

An unusual finding relating to the positive association rules, was that all of the positive rules were unexpected to some extent, i.e., positive rules can not be fully explained by the subject matter expert. As mentioned, there may be several possibilities to explain the occurrence of these frequent patterns. One possibility is that these rules indicate inappropriate billing practices. It is also possible that these rules reflected new billing patterns by specialists where the service rendered may not yet be reflected within the MBS billing structure. The third possibility is that these rules describe incorrect billing due to uncertainty or misunderstanding among specialists in relation to the correct billing method, other than deliberately taking advantage of MBS benefit.

Although there are 23 positive rules, these related to mainly two sets of items. Further research is proposed to determine the nature of these two sets of items so as to better understand the clinical context. Once confirmed by further research, these positive rules would be very valuable in assisting Medicare's educational intervention or government policy responses.

5.2 Relevance to compliance

To identify the specialists with anomalous, potentially fraudulent, behaviour, the rules were matched against all the episodes each specialist rendered. This allowed the total number of occasions where rules were broken by each specialist to be identified. The number of rule violations provided an indication of how much one specialist deviate from their peers.

As listed in Table 1, 162 negative rules are rated from low to high by the subject matter expert. Rules rated medium or high may be directly related to non-compliant practices. Therefore, all the specialists were checked against these high or medium rating rules. The specialists who broke these rules on the greatest number of occasions were identified as high risk. The best way of validate whether the individual specialists identified by this method were truly non-compliant, would be a review by a panel of peers or investigation possibly followed by legal action, which is prohibitively costly and time consuming. Luckily, there is a database available, called PRISM maintained by Medicare Australia, containing records of medical practitioners who have been approached in relation to previous compliance activities. Therefore, an alternative performance validation method is to match specialists identified by association rule mining against their compliance history in PRISM. This provides a reasonable estimate of the effectiveness of association rule mining in detecting non-compliant practice. Validation was against records within PRISM not necessarily linked to outcomes, however it is known that the majority of these records relate to specialists for compliance related issues.

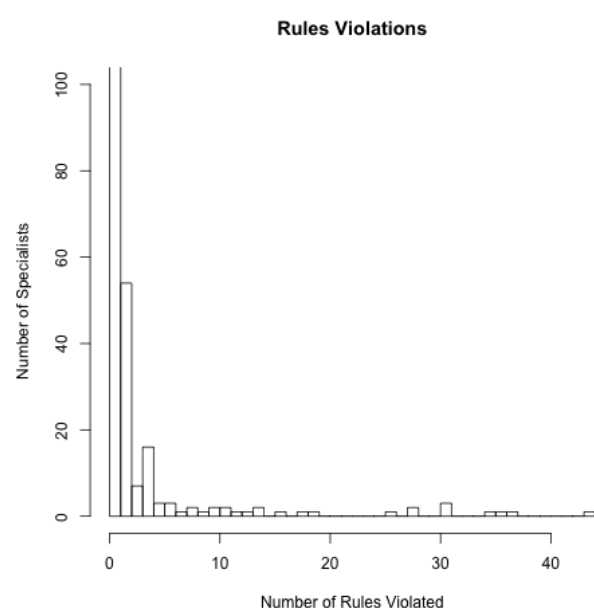


Figure 1: Number of Rule Violations

There were 779 specialists included in this study, based upon their derived specialty. Among them, there are only 129 specialists breaking rules on one or occasions. As can be seen in Figure 1, for those specialists who broke rules, most of them only broke rules on one occasion. The highest number of occasions of breaking rules is 44. The total number specialists who did not break any rules was 650 (83.44% of all specialists identified), which greatly exceeds the upper limit of y-axis in Figure 1.

As demonstrated in Figure 1, amongst those specialists who broke one or more rules, the vast majority of them break rules on less than five occasions and only an extremely small number of specialists broke the rules on more than 20 occasions. Therefore for further analysis, the specialists were divided into three overlapping groups based upon the number of occasions they had broken rules. These three classes were that specialists who broke rules in:

- 1) 1 or more occasions;
- 2) 5 or more occasions;
- 3) more than 20 occasions.

The accuracies of the association rules in detecting likely inappropriate billing are listed in Table 2. There were 10 specialists who broke these rules on more than 20 occasions. Amongst these, 5 had compliance records in PRISM resulting in an estimated accuracy of 50%. For specialists who broke more than 5 rules, the estimated accuracy was 25.81%, compared to an accuracy of 29.46% for those specialists who broke more than 1 rule. In order to put these accuracies into perspective, we constructed a baseline classifier, which randomly samples the data. As 163 of the 779 specialists have more than one compliance record, this results in an accuracy of 20.92%. Therefore, it is clear that the association rules mining method utilised in this study outperforms the random sampler. The fact that breaking even one or more rule can give us an accuracy of 29.46%, better than the baseline classifier (20.92%), suggests that breaking even one negative rule may be a good indication of non-compliant practice.

The compliance data base contains information about practitioners engaged in possible fraud or inappropriate practices. Often practitioners may be identified as having concerns in multiple areas. It was not possible to ensure that all practitioners from the PRISM data base were selected for compliance related issues. Some cases may have reflected past targeting strategies. This may have resulted in misclassification of practitioners. Provided, as may be assumed likely, this misclassification was non-differential it might be expected that the overall accuracy levels related to the number of rules violated would be higher, as this would have a dilutive effect.

No previous compliance activities have been undertaken in relation to the newly identified rules from this analysis.

Rules Violated	Specialists Identified	Specialists with compliance records	Accuracy
≥ 20	10	5	50.00%
≥ 5	31	8	25.81%
≥ 1	129	38	29.46%
Baseline	779	163	20.92%

Table 2: Accuracy of association rule in detecting potentially inappropriate practises, as measured by the percentage of specialists with past compliance records.

We are aware that specialists may have different numbers of records in their compliance history, which suggests some specialists have multiple incidents of non-compliant practice or have been engaged in multiple compliance activities. To measure the severity of a specialist's possible non-compliant practice, we calculated the average number compliance records listed in Table 3. For all the specialists who had compliance records, on average they had 1.47 records per specialist. For the three classes of specialists identified above, they have on average 1.53, 1.63 and 1.80 records per specialist, respectively. It is clear there is a close relationship between the number of occasions where rules were broken and severity of non-compliance of a specialist, measured by average number of records. In combination, these findings suggest that association rule mining can not only identify potential non-compliant specialists but also give us a good indication of the severity of their non-compliant behaviour.

Rules Violated	Specialists with compliance records	Average No. of records
≥ 20	5	1.80
≥ 5	8	1.63
≥ 1	38	1.53
All records	163	1.47

Table 3: Average number of compliance records per specialist.

We also checked the specialists against all the negative rules, not just the high and medium rating ones. It was unexpected that this gave similar accuracy to only checking against high and medium rating negative rules. This probably suggests that any broken rules may flag possible fraudulent activities. However, we will not be able to explore this further in the paper.

5.3 Negative rules vs. Positive rules

It was reported by the subject matter expert that negative rules may have certain advantages in compliance. Negative rules represent the absence of items being billed. In the compliance context, it may often be the case that more items than necessary are billed for financial gain. Such billing patterns are well captured by negative rules.

6 Conclusions and future research

This paper presents a novel application of association rule mining and demonstrates how both positive and negative association rule mining can be used with the aims of detecting fraud and inappropriate practice in the health service management domain.

The results were validated in several ways. The subject matter experts have confirmed the clinical relevance of the rules discovered. The individual specialists identified have good overlap with specialists who have compliance records. This demonstrates the effectiveness of this method for fraud detection and compliance. For further validation, this method is compared to the baseline classifier. This method significantly outperformed the baseline classifier. It is worth mentioning we have also demonstrated that this method may give a good indication of severity of the potential non-compliant activity as well.

This research clearly demonstrates that methods used were effective and we see immediate potential in the use of these methods to identify other relevant specialist groups to support compliance activities within Medicare Australia. Medicare Australia has run this technique against a limited number of medical specialties at this point and further validation will be undertaken based on compliance intervention feedback and future specialty group analyses. It is envisaged that this technique might be applied to a broad range of specialty and practitioner groups covered the Medicare system.

Further work is proposed as follows to enhance the use and evaluation association rules mining in compliance in the field of health service management.

A more comprehensive validation may be conducted. Information from other sources regarding the high risk specialists identified can be collected and limited cases might be audited to provide a more comprehensive assessment on the effectiveness and the accuracy of the association rule mining in detecting fraud and inappropriate billings.

It is possible that some of the rules identified by this type of analysis maybe false alarms, reflecting appropriate though infrequent practice. For this reason it is important that the outcomes of this form of analysis are further reviewed by a subject matter expert.

This technique is a substantial improvement over random auditing of practitioners engaged in specialised areas of practice.

This research focuses on the analysis of episodes, which only considers all items rendered in one given day. Therefore, there is no particular order recorded among items involved in one episode. However, the chronological order of the items may be crucial in determining the appropriateness of the billing. Therefore, it may be promising if the episode defined in the paper is expanded to "total episodes of care", covering 28 days, where the chronological information is recorded. With this added time dimension, this would be an interesting challenge for temporal data mining and may help further enhance the detection of non-compliant billings by Medicare.

7 Acknowledgements

The authors would like to thank Christian Borgelt and Irena Artamonova for their assistance in providing their implementation of association rule mining packages. The authors also wish to acknowledge the input of the three anonymous referees whose insightful comments improved the final version of this paper.

8 References

- Agrawal, R., Imielinski, T. and Swami, A. (1993): Mining association rules between sets of items in large databases. *Proc. ACM SIGMOD International Conference on Management of Data*, Washington DC, USA, **22**:207-216, ACM Press.
- Agrawal, R. and Srikant, R. (1994) Fast algorithms for Mining Association Rules. *Proceedings of the International Conference on Very Large Data Bases*, Santiago, Chile, 487 – 499.
- Artamonova, I., Frishman. G., Frishman D. (2007) Applying negative rule mining to improve genome annotation. *BMC Bioinformatics*. (Jul 21);**8** (1):261.
- Becker, D. and Kessler, D. and McClellan, M. (2005) Detecting Medicare abuse. *Journal of Health Economics*. **24**(1): 189-210.
- Borgelt, C. and Kruse, R. (2002). Induction of Association Rules: Apriori Implementation. 15th Conference on Computational Statistics (Compstat 2002, Berlin, Germany) Physica Verlag, Heidelberg, Germany. <http://www.borgelt.net/apriori.html>
- He H., Graco, W. and Yao X. (1998) Application of Genetic Algorithm and k-Nearest Neighbour Method in Medical Fraud Detection. In Second Asia-Pacific Conference on Simulated Evolution and Learning (SEAL '98), Canberra, Australia. pp. 74-81, LNAI 1585 Springer
- Lin, C., Lin, C-M., Li, S-T. and Kuo, S-C. (2008) Intelligent physician segmentation and management based on KDD approach. (2008) *Expert Systems with Applications*. **34**(3): 1963—1973. Pergamon Press, Inc. Tarrytown, NY, USA
- Medicare Benefit Schedule Book (2007) Australian Government, ISBN 1-74186-363-5, Department of Health and Ageing, Australian Government. Canberra. ISBN 1-74186-363-5
- Pearson, R., Murray, W. and Mettenmeyer, T. (2006) Finding Anomalies in Medicare. *electronic Journal of Health Informatics*. **1**(1): e2. (www.ejh.net/ojs/index.php/ejhi/issue/view/1)
- Savasere A., Omiecinski E., and Navathe S. B. (1998) Mining for Strong Negative Associations in a Large Database of Customer Transactions. *Proceedings of the International Conference on Data Engineering*, February
- Semenova T. (2004) Discovering patterns of medical practice in large administrative health databases. *Data Knowledge Engineering*. **51**(2): 149—160, Elsevier Science Publishers B. V. ISSN 0169-023X Amsterdam, The Netherlands
- Yang, W-S. and Hwang, S-Y. (2006) A process-mining framework for the detection of healthcare fraud and abuse. *Expert Systems with Applications*. **31**: 56–68. Elsevier
- Zhang, C., Zhang, S. (2002) Association Rule Mining: Models and Algorithms. *Lecture Notes in Computer Science*, **2307**. ISBN: 978-3-540-43533-4

Comparison of visualization methods of genome-wide SNP profiles in childhood acute lymphoblastic leukaemia

Ahmad Al-Oqaily¹Paul J. Kennedy¹Daniel Catchpoole²Simeon Simoff³

¹Faculty of Engineering and IT,
University of Technology, Sydney,
PO Box 123, Broadway, NSW 2007,
Australia,

²The Oncology Research Unit,
The Children's Hospital at Westmead,
Locked Bag 4001, Westmead NSW 2145,
Australia,

³School of Computing and Mathematics
University of Western Sydney,
Locked Bag 1797, Paramatta,
Australia,
Email: aaoqaily@it.uts.edu.au

Abstract

Data mining and knowledge discovery have been applied to datasets in various industries including biomedical data. Modelling, data mining and visualization in biomedical data address the problem of extracting knowledge from large and complex biomedical data. The current challenge of dealing with such data is to develop statistical-based and data mining methods that search and browse the underlying patterns within the data. In this paper, we employ several data reduction methods for visualizing genome-wide Single Nucleotide Polymorphism (SNP) datasets based on state-of-art data reduction techniques. Visualization approach has been selected based on the trustworthiness of the resultant visualizations. To deal with large amounts of genetic variation data, we have chosen to apply different data reduction methods to deal with the problem induced by high dimensionality. Based on the trustworthiness metric we found that neighbour Retrieval Visualizer (NeRV) outperformed other methods. This method optimizes the retrieval quality of Stochastic neighbour Embedding. The quality measure of the visualization (i.e. NeRV) showed excellent results, even though the dataset was reduced from 13917 to 2 dimensions. The visualization results will assist clinicians and biomedical researchers in understanding the systems biology of patients and how to compare different groups of clusters in visualizations.

Keywords: biomedical datasets, single nucleotide polymorphisms, SNP visualization.

1 Introduction

Data mining and knowledge discovery have been applied to datasets in various industries including biomedical informatics (Azuaje & Dopazo 2005). Data mining and visualisation in biomedical informatics addresses the problems of extracting knowledge from data originating from multiple sources, encoded in different formats or protocols, and pro-

cessed by multiple systems. As identified by (Bertone & Gerstein 2001), the problems have only recently been reviewed in a systematic way (Azuaje & Dopazo 2005). The major challenges in data mining in the area stem from the fact that biomedical data requires data structures that are multidimensional. It is our intention to construct models which incorporate large amounts of biomedical data in a manner which will alleviate the error induced by high dimensionality.

Methods and approaches applied in this paper rely on the information extracted from biomedical datasets, derived from cancer patients. This data includes genome-wide single nucleotide polymorphism genotyping data (genetic variations).

The domain of this paper is Childhood Acute Lymphoblastic leukaemia (ALL), which is the most common childhood malignancy. It represents 24% of all new cancers that occurred in children between 1995 and 1999 (240 ALL/985 Cancer patients) (Coates & Tracey 2001). Nearly all children with ALL achieve an initial clinical remission, so the major obstacle to cure is patient relapse, i.e. the recurrence of evident disease. The approaches and methods we apply to ALL data can also be extended to other complex diseases such as heart diseases, diabetes and inflammatory diseases.

Information visualization is considered as a direct way to help browse the datasets. It is possible to combine visual exploration with other data exploration tools such as clustering analysis and data comparisons. The result of data explorations can be confirmed on the visualization. The main challenge in visualizing genetic variation datasets stems from the high dimensionality of the data, which may includes tens of thousands of SNPs. In this paper, several visualization methods will be applied to genetic variation datasets, for example manifold-based reduction methods.

Traditional dimensionality reduction techniques include Principal Components Analysis (PCA) (Hotelling 1933) which tries to preserve the variance in the data, and Multidimensional Scaling (MDS) which tries to preserve pairwise distances between data points. These methods are used to find a low space representation of the high dimensionality space which preserves the global structure of the data. However, these methods are not adequate to handle high dimensionality data which could have nonlinear relationships.

Therefore, in the last decade a large number

Copyright ©2008, Australian Computer Society, Inc. This paper appeared at the Seventh Australasian Data Mining Conference (AusDM 2008), Glenelg, Australia. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 87, John F. Roddick, Jiuyong Li, Peter Christen and Paul Kennedy, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

of nonlinear techniques for dimensionality reduction have been proposed. Some of these methods are used to find a lower dimensionality manifold of the data or a nonlinear embedding manifold space in the higher-dimensional data space. The main advantage of these methods is that they are able to preserve the local relationships of the data, which can be advantageous for the task of information visualization. Several of these methods will be described in section 4. The main difference between manifold estimation and visualization is that visualization is limited to two or three dimensions (Venna & Kaski 2007a). Thus, it is difficult to know the exact number of dimensions to uncover the underlying structure of the data. Therefore, we need to apply different manifold-based methods on the given dataset in order to choose the most appropriate method.

In this work, we will study the results of applying different dimensionality reduction methods to genome-wide SNP profiles of leukaemia patients to determine which is the best method for visualizing this type of data. The results will be compared based on measures such as trustworthiness and continuity of the visualizations.

The rest of this paper is organized as follows. Section 2 points to related work that has been applied for visualizing biomedical data, specifically SNP data. Section 3 describes the dataset used in this study and the preprocessing steps applied to the data. Next, in section 4 we describe methods and techniques that will be used to visualize the ALL data. Section 5 describes in detail the experiments and results for different methods. In section 6 we further discuss these results. Finally, in section 7 we conclude the paper and describe the future directions for our research.

2 Related Work

The current interest in genetic variation studies is focused on disease-gene association analyses. Such analyses are important in identifying which variants are associated with a specific disease. Identification of genetic variants that contribute to susceptibility of diseases such as cancer will assist in the development of diagnostic and therapeutics (Carlson, Eberle, Kruglyak & Nickerson 2004). To identify these markers, at a statistically significant level, it is necessary to obtain genetic information from a large scale population sample of affected and unaffected individuals, which is termed a population based study. However, recent advances in biomedical technologies and genetics studies have made association studies a powerful approach for mapping complex-disease genes by conducting studies based on the whole genome. These studies are called genome-wide association (GWA) studies, in which a dense set of SNPs across the genome is genotyped to survey the most common genetic variations for a role in a disease or to identify the heritable quantitative trait that is a risk factor of a disease (Hirschhorn & Daly 2005).

Genome-wide association studies are mainly conducted using statistical methods, which are used to discover genetic factors that contribute to susceptibility to disease. Factors that show a significantly high statistical level of association are chosen for further analyses. However, in this paper, we are heading in a different direction. Data mining approaches will be used here. These approaches mainly concentrate on visualizing genome-wide SNP datasets based on state-of-art data reduction techniques. The visualization results will assist clinicians and biomedical researchers in understanding the different structure of patients and how to compare different group of patients' clustering in the visualization.

3 Data

In this section we describe in detail the dataset used and the preprocessing steps applied.

3.1 Single Nucleotide Polymorphism (SNP) data

The human genome was found to contain a large amount of genetic variation in the form of sequence polymorphisms. Polymorphism is a variation of DNA sequence that has an allele frequency of at least 1% of the population (Cavalli-Sforza 1974). There are several types of polymorphism in the human genome: SNPs, repeated polymorphisms and insertions or deletions, ranging from a single base-pair to thousands of base-pairs in size (Tabor, Risch & Myers 2002). Single Nucleotide polymorphisms (SNPs) are the simplest but most abundant type of genetic variation among individuals with between 1 to 10 million existing in the human genome (Donnelly 2004). These common SNP are thought to account for around 90% of human polymorphism (Carlson, Eberle, Rieder, Smith, Kruglyak & Nickerson 2003, Reich, Gabriel & Altshuler 2003).

Genetic variations, especially SNPs, are known to be the key feature of discovering disease-genes. In the case of complex disease, identifying multiple genetic variants would be possible by conducting association analysis between a specific variant and a disease. This association involves examining all genetic differences in a large number of affected individuals with unaffected controls (Risch & Merikangas 1996).

3.2 Genetic Variation and Childhood ALL

Chromosomal imbalances have long been known to be key features of leukaemia. Further, the human genome was found to contain a large amount of genetic variation in the form of sequence polymorphisms. Non-synonymous (ns)SNPs occurring within coding regions are those which produce an amino acid change but are not considered a "mutation" as a functional protein is still transcribed. Such nsSNPs are known to affect the functional efficiency of genes (Aplenc & Lange 2004). For example, drug metabolism and patient response to chemotherapy. SNP's which are found throughout non-coding intronic genome regions are used in major disease linkage and haplotyping studies including the HapMap Project (Altshuler, Brooks, Chakravarti, Collins, Daly, Donnelly et al. 2005) whilst identification of minor regions of amplification or deletion within the genome are facilitated through assessment of SNP copy number (Herr, Grützmann, Matthaei, Artelt, Schröck, Rump & Pilarsky 2005). However, genetic variation of the human genome is a promising resource for studying complex diseases such as cancer. Large number of genetic variations, scattered across the human genome, represent a remarkable opportunity to investigate the etiology, inter-individual differences in treatment response and outcomes of specific cancer such as leukaemia (Erichsen & Chanock 2004). Thus, we are in a position of utilizing such a tool (i.e. SNPs data) to analyze genetic contributions to complex diseases. Such analyses could have big influences on the prevention and early intervention strategies of a disease.

3.3 ALL Dataset

Genome-wide SNP data incorporates large scale mapping of SNPs and subsequent collation into databases. Generation of SNP data has been facilitated by high

throughput microarray-based technologies ((Barker, Hansen, Faruqi, Giannola, Irsula, Lasken, Latterich, Makarov, Oliphant, Pinter et al. 2004), (Leykin, Hao, Cheng, Meyer, Pollak, Smith, Wong, Rosenow & Li 2005), (Irving, Bloodworth, Bown, Case, Hogarth & Hall 2005)). DNA from a cohort of 139 childhood ALL patients are generated with the Illumina Bead Array system (Fan et al. 2003) using the non-synonymous beadchip to assess 13,917 SNPs across the genome within exon-centric loci.

3.4 Data preprocessing

The SNP dataset contains information about 13,917 SNPs which scattered across the whole genome. These SNPs are classified as non-synonymous (functional) SNPs which affect the functionality of genes. Each individual's genome has two alleles of a given SNP. For most cases there are two alleles for each SNP (Bi-allelic). At a specific SNP, a person can have one of the several genotypes. When they are the same the SNP is called homozygous and when they are different the SNP is called heterozygous. For a single SNP, one is designated the major allele and the other the minor allele, based on their observed frequency in a general population (Crawford & Nickerson 2005). Each SNP can have four different values (nominal): two homozygous, one heterozygous or missing. That is, the four possibilities for alleles A and B of the i th SNP are two homozygous (AA or BB), one heterozygous (AB) or missing NA (not determined). All SNPs were transformed into numerical data based on Minor Allele Frequency, as described in (Price, Patterson, Plenge, Weinblatt, Shadick & Reich 2006).

Let \mathbf{G} be a matrix of genotype data, g_{ij} is the genotype for SNP i and individual j where $i = 1$ to M and $j = 1$ to N . The row mean $\mu_i = (\sum_j g_{ij})/N$ is subtracted from each entry in each row i , to obtain row sums equal to 0. Missing entries are excluded from the computation of μ_i and are subsequently set to 0. Each row i is then normalized by dividing each entry by $\sqrt{p_i(1-p_i)}$ where p_i is a posterior estimate of the unobserved underlying allele frequency of SNP i defined by

$$p_i = (1 + \sum_j g_{ij}) / (2 + 2N) \quad (1)$$

with missing entries excluded from the computation. We denote the resulting matrix as \mathbf{G} -normalized. The new matrix is regarded as normalized version of data matrix. The mean of each row i is equal to 0.

4 Methods and Approach

In this section, we describe some of the dimensionality reduction methods for visualizing the similarity relationship between patients. Firstly, we will describe the main classical methods for dimensionality reduction i.e. Principal Component Analysis (PCA), and Multidimensional Scaling (MDS). Some other methods based on MDS will be described. Then, other recently proposed methods that focus on finding the manifold or embedding of data will be described. Lastly, we will describe the measures for the goodness of visualization that we use in the experiments.

The problem of dimensionality reduction can be defined as follows. Given a dataset matrix $\mathbf{X} \in \mathbb{R}^{N \times d}$ consisting of N data vectors x_i ($i = 1, 2, \dots, N$) with dimensionality d which can be considered as points in a high-dimensional data space. Dimensionality reduction methods transform the data set \mathbf{X} with dimensionality d into a new data set $\mathbf{Y} \in \mathbb{R}^{N \times p}$ with dimensionality p ($p \ll d$), while preserving the geometry

of the data as much as possible. The low-dimensional representation of x_i is denoted by y_i , where y_i is the i th row of the p -dimensional data matrix \mathbf{Y} . For visualization purposes the dimensionality representation of the *output space* needs to be two or at most three dimensions, whereas the original space or *input space* can be thousands of dimensions.

Generally, the task of visualization methods is to construct a low-dimensional representation (i.e. output space) y_i of the input space, in such a way that the original relationships (or similarities) of the data are preserved. However, lower-dimensional representation of the data in 2 or 3-D dimensions might not be able to preserve all the information of the original (higher-dimensional) datasets and a compromise must be made by applying different data reduction methods and then selecting the best method based on how well a given method preserves the information of the original data (Venna & Kaski 2007a).

4.1 Principal Components Analysis

Principal components analysis (PCA) constructs a low-dimensional representation of the data that maximally preserves as much variance in the data as possible (Hotelling 1933). This is done by finding the linear projection or direction where the data has maximum variance. The projection can be found by solving the eigenvalue problem of the covariance matrix \mathbf{C}_x of the data using the general eigen-decomposition problem

$$\mathbf{C}_x \mathbf{a} = \lambda \mathbf{a} \quad (2)$$

It can be shown that the linear projection is formed by the p principal components of the covariance matrix. The new representation of data points x_i can then be found by projecting (or mapping) the original data with

$$y_i = \mathbf{A} x_i \quad (3)$$

The low-dimensional data representations y_i of the data point x_i are computed by projecting data matrix \mathbf{X} using matrix \mathbf{A} , which contains the eigenvectors corresponding to the two or three largest eigenvalues. The new representation of the data can be visualized using the projected matrix \mathbf{Y} .

PCA has been successfully applied in a large number of domains. However, the main limitation of PCA is that it does not work well when the data lies in a nonlinear manifold. However, PCA is advantageous when the variance of the data is mainly concentrated in a few directions.

4.2 Multidimensional Scaling

Multidimensional Scaling (MDS) (Torgerson 1952) represents approaches that are commonly used with nonlinear mapping methods. There are several different variants of MDS (Cox & Cox 2001), but they all share a common goal which is to find the low-dimensional representation of the data that preserves the pairwise distance of the data as much as possible. The quality of the mapping is represented by a stress function (or cost function), which tries to minimize the errors of the pairwise distances between the low-dimensional and high-dimensional representations of the data.

The classical version of MDS is very closely related to PCA. The solution of linear MDS can be found by solving an eigen-decomposition problem. When the dimensionality of the sought space is the same and the distance measure is Euclidean distance, the projection of the original data using PCA is similar to the configuration of points that calculated by squared Euclidean distance matrix of the data (Gower 1966).

Other variants of MDS which have a more effective stress function are the raw stress function and Sammon cost function. The raw stress function can be defined by

$$\phi(Y) = \sum_{ij} (\|x_i - x_j\| - \|y_i - y_j\|)^2 \quad (4)$$

where $\|x_i - x_j\|$ is the Euclidean distance between x_i and x_j in data points in the original data space, and $\|y_i - y_j\|$ is the Euclidean distance between y_i and y_j data points in the low-dimensional space. This cost function is able to find nonlinear relationships in the data. The Sammon cost function is slightly different to the raw stress in that it gives small distances a larger weight, which emphasises the local relationships in the data. In addition, there exist other variants of MDS, called non-metric MDS, which aim to preserve ordinal relations in data, rather than the pairwise distance (Kruskal 1964). Nevertheless, Multidimensional Scaling has been widely used for data visualization, such as Functional Magnetic Resonance Imaging (fMRI) analysis and molecular modelling (Tagaris, Richter, Kim, Pellizzer, Andersen, Ugurbil & Georgopoulos 1998, Venkatarajan & Braun 2004). The success of MDS has led to the proposal of new variants such as Curvilinear Component analysis (Demartines & Herault 1997) and Stochastic neighbours Embedding (SNE) (Hinton & Roweis 2003). These methods have shown the capability to produce good quality visualizations. Extended versions of these methods will be described in the following sections.

4.3 Stochastic neighbour Embedding

Stochastic neighbour Embedding (SNE) proposed by (Hinton & Roweis 2003) is a probability-based embedding method. SNE tries to find the low-dimensional representation of data points that preserve neighbourhood identities. The SNE algorithm tries to preserve the probability distribution of the pairwise distances of data points in the input space, so that the probability of a data point i being a neighbour of point j in the output space is the same as in the input space.

For each data point x_i and its potential neighbours, X_j , the algorithm starts by computing p_{ij} , the probability that point x_i and x_j are neighbours in the input space using

$$p_{ij} = \frac{\exp(-d(x_i, x_j)^2)}{\sum_{i \neq k} \exp(-d(x_i, x_k)^2)} \quad (5)$$

where $d(x_i, x_k)^2$ is the pairwise distance between data points i and j . The distance can simply be the squared Euclidean distance or it can be the scaled squared Euclidean distance if we have a high-dimensional data

$$d(x_i, x_k)^2 = \frac{\|x_i - x_j\|^2}{2\sigma_i^2} \quad (6)$$

In low-dimensional output space the images y_i of all data point x_i is defined as q_{ij} , which express the probability of the point y_i being a neighbour of point y_j .

$$q_{ij} = \frac{\exp(-d(y_i, y_j)^2)}{\sum_{i \neq k} \exp(-d(y_i, y_k)^2)} \quad (7)$$

The aim of the embedding is to match the two probability distributions p_{ij} and q_{ij} as well as possible.

The embedding of points y_i can be achieved by minimizing a cost function which is the Kullback–Leibler divergence between the probability distribution of the input (p_{ij}) and output (q_{ij}) distribution over neighbours of each data point. The cost function is

$$E_i[D(p_i, q_i)] = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \quad (8)$$

Stochastic neighbour Embedding has been successfully applied to several datasets (eg. (Nguyen & Worring 2004) or (Memisevic & Hinton 2005)). Results show that good optima can be achieved.

Stochastic neighbour Embedding was originally designed as a data reduction method that tries to preserve neighbourhood identities. However, SNE can be also seen as an information retrieval algorithm. A new restructured method called neighbour Retrieval Visualizer (NeRV) was proposed by (Venna & Kaski 2007b). This method is motivated by visual neighbour retrieval, unlike SNE, which tries to optimize recall (i.e. misses). The method balances the error caused by precision (i.e. false positive, see section 4.7).

In information visualization, high precision is more important than recall. Minimizing precision is associated with preserving the neighbourhood of points in the output space. Recall on the other hand, tries to preserve the neighbourhood of points in the input space. Stochastic neighbour Embedding updates the original SNE method by assigning a relative cost λ to recall and $(1 - \lambda)$ to precision. Then, the total function to be optimized is

$$\begin{aligned} E &= \lambda E_i[D(p_i, q_i)] + (1 - \lambda) E_i[D(q_i, p_i)] \\ &= \lambda \sum_{i,j} p_{ij} \log \frac{p_{ij}}{q_{ij}} + \\ &\quad (1 - \lambda) \sum_i \sum_j q_{ij} \log \frac{q_{ij}}{p_{ij}} \end{aligned} \quad (9)$$

That is, by setting the parameter $\lambda \in [0, 1]$ the choice can be focused on either the probabilities that are high in the input space (recall) or in the output space (precision). When $\lambda = 1$ the method is equal to SNE and when $\lambda = 0$, the method focuses completely in avoiding false positives (precision). This method can be described as retrieving points based on the visualization display. In our experiment we apply this method with choice of λ that emphasizes the underlying structure of the data that maximizes precision. In addition, SNE will be applied for comparison purposes.

4.4 Curvilinear Component Analysis

Curvilinear Component Analysis (CCA) (Demartines & Herault 1997) is a variant of MDS. Whereas MDS tries to find the configuration of points that preserve the pairwise distances as much as possible, CCA tries to find the configuration of points that preserve a subset of the distances that are neighbours in the output space. The cost function of CCA concentrates on preserving the distance of points in the reduced space. This can be done by introducing a weighted function F that depends on the distance between the points in the output space (or visualization), yielding a cost function

$$E = \frac{1}{2} \sum_i \sum_{i \neq j} (d(x_i, x_j) - d(y_i, y_j))^2 F(d(y_i, y_j), \sigma_i) \quad (10)$$

Generally, $F(d(y_i, y_j), \sigma_i)$ is chosen as a bounded and monotonically decreasing function, in order to favor preserving the local geometry of the data. Decreasing exponential, sigmoid, or Lorentz functions can be suitable choices, and a simple step function can also be applied.

$$F(d(y_i, y_j), \sigma_i) = \begin{cases} 1 & Y_{ij} \leq \sigma_i \\ 0 & Y_{ij} > \sigma_i \end{cases} \quad (11)$$

The minimization of the cost function can be achieved using a form of stochastic gradient decent algorithm. During the optimization process, σ_i is set to cover all or at least most of the data points (as the case of MDS), and it is slowly decreased to reach the optimal value.

Curvilinear Component Analysis has been successfully applied to various nonlinear-dimensionality problems in data representation such as for gene expression data and computer vision (Buchala, Davey, Frank & Gale 2004, Venna & Kaski 2007a). An extension of CCA, Curvilinear Distance Analysis (CDA), was introduced by (Lee, Lendasse & Verleysen 2004). The main difference of CDA compared to CCA is to replace the Euclidean distance used by CCA with geodesic distance. Geodesic distance is based on graph theory and uses the minimum spanning tree to find the distance.

The main drawback of CCA is that the cost function may have several local optima. Although this can cause undesired results when applying CCA, solutions found by CCA have showed quite reasonable results (Venna & Kaski 2007a).

Recently, a method called Local Multidimensional Scaling (LocalMDS) was proposed (Venna & Kaski 2006). This method is regarded as a derivative of CCA. Similarly to NeRV, LocalMDS has the indirect ability to control the tradeoff between precision and recall, which helps for data visualization. The cost function of CCA tries to preserve the distance of points that are neighbours in the output space, by ignoring the error in distance between points that are far from each other in the reduced space. Thus, CCA could increase the errors caused by recall, which can result in lower visualization quality. In LocalMDS, a term is added to the cost function to increase recall. This can be achieved by penalizing the errors of distance between points that are close by in the input space. The tradeoff between the two types of errors helps in having a more efficient display of the local similarities of the data. The cost function of LocalMDS is defined as

$E =$

$$\sum_i \sum_{i \neq j} [(1 - \lambda)(d(x_i, x_j) - d(y_i, y_j))^2 F(d(y_i, y_j), \sigma_i) + \lambda(d(x_i, x_j) - d(y_i, y_j))^2 F(d(x_i, x_j), \sigma_i)] \quad (12)$$

where $\lambda \in [0, \dots, 1]$ controls the tradeoff between precision and recall. During the optimization the radius of the area of influence around data point x_i , σ_i , is slowly reduced to reach the optimal value. $F(d(x_i, x_j), \sigma_i)$, similarly to CCA, emphasizes the local distance in the input space. F is equal to one when $d(x_i, x_j) < \sigma_i$ and 0 otherwise. The final radius is set equal to the distance of k -NN of a data point x_i in the original space.

When $\lambda = 0$ the cost function will be that of the basic CCA method. A good choice of λ ranges from 0 to 0.5. The cost function can be optimized using stochastic gradient descent methods similarly to CCA. In our experiments we apply LocalMDS with a

choice of λ that emphasizes the underlying structure of the data to maximize precision. In addition, CCA will be applied for comparison purposes.

4.5 Laplacian Eigenmap

Laplacian Eigenmap (LE) finds a low-dimensional representation of data by preserving the local structure of the data (Belkin & Niyogi 2002). Laplacian Eigenmap is regarded as a geometrically motivated dimensionality reduction. The output space reflects the intrinsic geometric structure of the manifold. In Laplacian Eigenmap, the local structure can be preserved by keeping the local structure between each datapoint and its k nearest neighbours. Therefore, the local structure of LE algorithms can be relatively insensitive to outliers and noise, and as a result the algorithm implicitly emphasizes the natural clusters in the data (Belkin & Niyogi 2002).

Laplacian Eigenmap computes a low-dimensional representation of the data in which the nearest neighbours of a datapoint in the original space should be mapped to nearest neighbours of that datapoint in the reduced space (He, Yan, Hu, Niyogi & Zhang 2005). This can be done in a weighted manner applied to graph partitioning, i.e., using a weighted criterion such as a heat kernel (Gaussian function) enables us to choose the weight of the graph in such a way that keeps the local similarity of the graph. The embedding map is constructed by computing the eigenvectors of the graph Laplacian. The algorithm's procedures are as follows.

The LE algorithm first constructs the adjacency graph G in which every node (datapoint) x_i is connected to its k nearest neighbours. For all nodes i and j in the graph G that are connected by an edge, a weight is calculated using different methods such as a Gaussian kernel or a simple approach where $W_{ij} = 1$ if node i and j are connected by an edge. This leads to a sparse matrix W in which $W_{ij} > 0$ if node i and j are connected and $W_{ij} = 0$ otherwise.

To compute the low-dimensional representation $y = y_1, y_2, \dots, y_n^T$, Laplacian Eigenmap minimizes the following objective function

$$\phi(Y) = \sum_{ij} (y_i - y_j)^2 w_{ij} = \text{tr}(Y^T \mathbf{L} Y) \quad (13)$$

where $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the Laplacian matrix, \mathbf{D} is diagonal matrix, with elements $D_{ii} = \sum_j W_{ij}$ being the column (or row, since W is symmetric) sums of W . The Laplacian matrix is symmetric and positive semidefinite.

Minimizing the objective function tries to put datapoints that are connected in the graph G as close together as possible. There is a trivial solution to the objective function which collapses all the new representations of the graph G into a single location. This can be prevented by adding an orthogonality constraint $Y^T \mathbf{D} Y = 1$.

The configuration of points in the low-dimensional space can be solved by finding the eigenvectors and eigenvalues of the generalized eigenvector problem

$$\mathbf{L} y = \lambda \mathbf{D} y \quad (14)$$

The low-dimensional embedding of the original data points can be formed by the d eigenvectors y_i that correspond to the smallest non-zero eigenvalues, after discarding the smallest eigenvector that corresponds to the zero eigenvalues, which represent the case where all data points are represented by a single location.

Laplacian Eigenmap has been successfully applied to number of domains such as clustering and face recognition (Ng, Jordan & Weiss 2002, Shi & Malik 2000, He et al. 2005). Variants of Laplacian Eigenmaps have been extended to supervised and semi-supervised data analysis (Costa & Hero 2005, Belkin & Niyogi 2004). A linear variant of Laplacian Eigenmap is proposed by (He & Niyogi 2004).

Laplacian Eigenmap has two main drawbacks. Firstly, in most applications it is not possible to see the structure within clusters from the visualization. Secondly, this method is mainly used for data representation or visualization and can not compute the projection for a new test point. However, this problem can be solved using techniques proposed by (Bengio, Paiement & Vincent 2004) called an out-of-sample extension.

4.6 Locally linear Embedding (LLE)

The LLE algorithm (Roweis & Saul 2000) is similar to Laplacian Eigenmap, which tries to preserve the local geometry of the data by finding the *local linear* approximation of the manifold. This is based on the assumption that a data point and its neighbours lie in or close to a locally linear subspace on the manifold. In LLE, the local geometry of this subspace can be characterized by calculating the linear coefficients (weights) that reconstruct each data point from its k nearest neighbours. In the low-dimensional space of the data, LLE attempts to retain the reconstruction weights in the linear combination as much as possible (van der Maaten, Postma & van den Herik 2007).

The algorithm works in two stages. First, the local coordinate of each data point is calculated based on its k nearest neighbours, and the total reconstruction error to be optimized is then measured by the cost function

$$\epsilon(W) = \sum_{i=1}^N \left\| X_i - \sum_{j=1}^k W_{ij} X_j \right\|^2 \quad (15)$$

which adds up the squared distance between all data points and their reconstruction. The weight W_{ij} summarizes the contribution of the j th data point to the i th reconstruction. The reconstruction error is minimized subject to the constraints that $W_{ij} = 0$ if datapoints i and j are not neighbours and $\sum_j W_{ij} = 1$.

In the second stage, the task is to find the low-dimensional representations y_i that preserve the local geometry of the data as described by the local coordinate of each data point. In other words, the reconstruction weights W_{ij} that reconstruct each datapoint x_i from its neighbours in the high-dimensional data space also reconstruct each data point y_i in the low-dimensional space. To do so, the p -dimensional reduced space \mathbf{Y} can be computed based on minimizing the cost function

$$\epsilon(Y) = \sum_{i=1}^N \left\| y_i - \sum_{j=1}^k W_{ij} y_j \right\|^2 \quad (16)$$

(Roweis & Saul 2000) showed that the optimization function described in (16) can be solved by the eigenvectors that correspond to the p nonzero eigenvalues of matrix \mathbf{M} , where $\mathbf{M} = (\mathbf{I} - \mathbf{W})^T (\mathbf{I} - \mathbf{W})$ and \mathbf{I} is the identity matrix.

A linear variant of LLE algorithm was proposed recently (Kokiopoulou & Saad 2005, Kokiopoulou & Saad 2007).

4.7 Comparing visualisations

As we have discussed, the first step in exploring the structure of a given dataset is to have the data visualized. In many previous works, visualization methods are compared through examining the produced figures. Some quantitative criteria should be designed, to compare the visualization results without considering the human as a part of visualization.

One of the crucial tasks in data visualization is how to assess the quality of produced visualizations or the tools that are used. The quality measure is used to assess how well the visualization of a given tool can represent the underlying data. The local structure of the data is the most important component of the visualization. The usability of the visualization can be measured by how accurately the data is represented and how readable it is.

The first question that comes to mind is how trustworthy is the visualization. The local similarity or structure of a data is the most crucial part of the visualization. When looking at the visualization, the first insight is how points are similar and how points group together. Looking at a visualization a user can possibly get insight into some question such as, are the unknown data points similar to the known ones? How is the data clustered? Are there denser areas and more sparse ones? Questions like these cannot be answered without having a visualization that is capable of answering these questions.

There are a number of methods that have been implemented to assign a quantity to a visualization. Some of these methods calculate the correlation coefficient between the distance vectors (i.e., the vectors that compare the distance between all pairs of points) of the original space with that of the lower dimensional space. It was proven that this measure can provide a good measurement of quality of the visualization procedures (Tan, Steinbach & Kumar 2005).

Others methods measure how trustworthy the local structure of the visualizations is (Kaski, Nikkila, Oja, Venna, Toronen & Castren 2003, Venna & Kaski 2001). Based on these methods the low-dimensional representation is trustworthy if the k nearest neighbours of a point in the reduced space (or in the visualization) are also neighbours of the point in the original space. The proportion of points that are in the neighbourhood in the visualization but not in the original space is quantified as the precision (or loss of precision, i.e., one minus precision). This number is usually not informative. However, the magnitude of the error can be used to rank the data points based on their distance instead of just counting the number of errors.

Reducing the dimensionality of a data can result in losing some of the similarity relationships between data points. Two general errors can be caused in applying a reduction method. First, data points that are not neighbours in the input space can be mapped close by in the reduced space, causing points to be incorrectly identified as neighbours. These kind of errors can reduce the *precision*. Secondly, data points that are neighbours in the input space can be mapped far away in the reduced space, causing discontinuities in the mapping and can distort the neighbour relations. This kind of error is called *recall*. The two kind of errors (i.e. precision and recall) are used in information retrieval literature in which the error is quantified based on the proportion of the points that caused the errors.

The main limitation of using *precision* and *recall*, as it is used in information retrieval, is that each of the errors is equally bad. However, in the visualization context this kind of measurement is not intuitive, whereas the distance between data points are known.

Intuitively, a data point that comes into the neighbourhood of another from far away causes a larger error than one that comes from closer. By ranking data points based on their similarity we can have two new quality measures: *trustworthiness* and *continuity* (Kaski et al. 2003) which quantify the errors of a visualization tool by the neighbourhood ranks of each data point.

The *trustworthiness* of a visualization can be defined as follows. Let N be the number of data samples and $r(x_i, x_j)$ be the rank of the sample x_j in the ordering according to the distance from data sample x_i in the original space. Let $U_k(x_i)$ be a set of data samples of size k that are in the neighbourhood of sample x_i in the visualization space but not in the original space. The measure of trustworthiness is defined as

$$M_{Tru}(k) = 1 - A(k) \sum_{i=1}^N \sum_{j \in U_k(i)} (r(x_i, x_j) - k) \quad (17)$$

where $A(k) = 2/(Nk(2N-3k-1))$ scales the measure between zero and one. The errors reach the maximum value when the ranks in the input and output space are reversed. The trustworthiness measure is closely related the precision (as in information retrieval). However, the trustworthiness measure is a special kind of precision measure for the case where the objects are ranked based on their relevance (Venna & Kaski 2006).

On the other hand, discontinuities are used to quantify whether neighbours in the original space remain neighbours in the visualization. If neighbour's points are pushed out in the displayed visualization, discontinuities arise in the visualization. The errors caused by discontinuities may be quantified similarly to the errors caused by trustworthiness.

Let $V_k(x_i)$ be the set of data samples that are neighbours of the data sample x_i in the original space but not in the output space and $\hat{r}(x_i, x_j)$ be the rank of data sample x_j in the ordering according to the distance from x_i in the visualization. The effects of discontinuities of the mapping are measured by:

$$M_{disc}(k) = 1 - A(k) \sum_{i=1}^N \sum_{j \in V_k(i)} (\hat{r}(x_i, x_j) - k) \quad (18)$$

Therefore, a data sample that is mapped far away from the neighbourhood in the reduced space will cause a larger error than a data sample that was mapped just out of the neighbourhood. In the recall measure both errors are considered equally severe.

The trustworthiness and continuity measures can be used to assess the quality of a data reduction method or to compare the performance of different data reduction method on a data set for visualization. Based on the quality results, the best run, method or parameters for a data set can be selected (Venna & Kaski 2007b). Comparing the data reduction methods can be tested on a large range of neighbourhood size. Small k can be important for the quality of visualization but a range of neighbourhood size can give an overview of the overall performance of different methods.

The performance of different data reduction methods is made by plotting the trustworthiness and continuity measures as a function of the neighbourhood size k . In any given data reduction method, a tradeoff must be made between trustworthiness and continuities. Seeking for a high trustworthiness will typically lead to a lower continuity and vice versa.

5 Experiments and Results

The purpose of the experiments is to gain insight and to understand the behavior of different dimensionality reduction methods in biomedical data and, more specifically, SNP data of children with acute lymphoblastic leukaemia. As described above, a dataset of 139 patients with 13917 non-synonymous SNPs (dimensions) was used.

The performance of dimensionality reduction methods will be compared on visualizing the SNP dataset. The following methods will be included in our experiments: Principal Component Analysis (PCA), Laplacian Eigenmap (LE), Locally Linear Embedding (LLE) and methods based on Multidimensional scaling, which include an extended version of Curvilinear Component Analysis (CCA) called Local MDS (LocalMDS) (Venna & Kaski 2006) and an extended version of Stochastic neighbour Embedding (SNE) called neighbour Retrieval Visualizer (NeRV) (Venna & Kaski 2007b). In the following subsection, the experimental settings and the results of experiments on SNP dataset are described.

5.1 Experimental setting

All methods except PCA have a parameter k for setting the number of nearest neighbours. This parameter was tested with values of k ranging from 5 to 30. The best k was selected based on the best result. However, small neighbourhood size can be related to the data points that are most likely to be relevant. The performance of the resulting visualizations was tested based on the trustworthiness and continuities of the reduced dimension, as described in section 4.7.

Some of the applied methods such as LocalMDS and NeRV that may fall into local optima were run several times (in our case 10 times) with different random initialization and the best run was selected. Random mapping was computed based on the average of 10 different random projections. Two types of distance metric were used to calculate the distance of data in the input space: Euclidean distance and Gaussian function. In this study, we employed both of these with different parameters and we set the dimensions of the output space equal to two for visualization purposes.

5.2 Results

Data reduction methods were compared using the trustworthiness and continuity measures of the resulted visualization. Figure 1 and 2 shows the trustworthiness and continuity results of the applied methods. The following subsection will get insight on different aspect of the results. For visualization purposes, trustworthiness is more important than continuity. In each case the result with the best trustworthiness was reported.

5.2.1 Trustworthiness and continuity

Trustworthiness and continuity are the first aspects that we examined. In terms of exploring the result of a visualization, the local neighbourhood of each data point is the first insight a human analysis looks at. Therefore, a visualization is trustworthy if the visualization preserves small neighbourhoods as much as possible. Thus, attention should be paid to small sizes of k (e.g. k between 5 and 15). It is clear from figure 1 that, in terms of trustworthiness, the NeRV method is the best. Unexpectedly, PCA is also quite good at preserving the locality of the reduced neighbourhood (Trustworthiness). On the other hand, state-of-art

data reduction methods such as LLE and LE were not able to produce reasonable results. In fact, LE was the worst method compared in our experiments on this dataset and is the most similar one to random mapping. The LLE and LE results suggest that the minimum number of dimensions that are required to uncover the manifold of the data is greater than two.

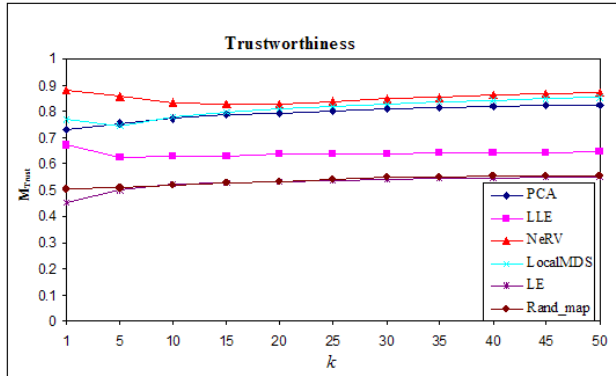


Figure 1: Trustworthiness of the mapping as a function of k that applied to SNP dataset, where k is the size of neighbourhood. Small neighbourhood sizes are the most important ones. PCA: Principal Component Analysis, LLE: Locally Linear Embedding, NeRV: neighbour Retrieval Visualizer, LocalMDS: Local Multidimensional scaling, LE: Laplacian Eigenmap rand_map: random mapping.

In this initial analysis LocalMDS was expected to perform similarly to NeRV method but the result shows slightly different behavior. In this data set PCA performs similarly to LocalMDS, even though the original cost function of LocalMDS emphasizes trustworthiness and should perform better.

In terms of continuity, as can be seen in figure 2, the result of NeRV method is again the best. But this time LocalMDS is slightly better than PCA which is different than the case of Trustworthiness where both methods are similar. Once more, manifold-based methods, LLE and LE, perform very badly on this data set. In section 6 we will suggest reasons why this occurred.

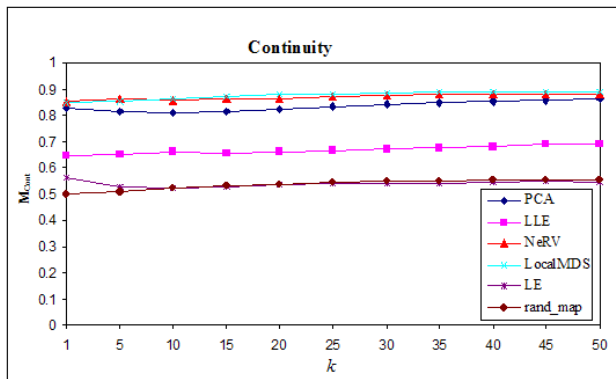


Figure 2: Continuity of the mapping as a function of k that applied to SNP dataset, where k is the size of neighbourhood. Small neighbourhood sizes are the most important ones. PCA: Principal Component Analysis, LLE: Locally Linear Embedding, NeRV: neighbour Retrieval Visualizer, LocalMDS: Local Multidimensional scaling, LE: Laplacian Eigenmap, rand_map: random mapping.

For the NeRV method, different parameters were set to explore the performance of this method on the

dataset. As can be seen in figure 3, we have applied the NeRV method for different neighbourhood sizes ranging from 5 to 30. If a small size of neighbourhood is considered around each point on the output space, a neighbourhood size of 5 or 15 produces the best results on this data. The same results were also found by LocalMDS, as can be seen on figure 4. Next, we set $k = 15$, and ran NeRV on a range of $\lambda = 0$ to 1. The result can be seen in figure 5. This shows that the best trustworthiness occurs when λ equals 0.0 or 0.1. This result confirms the performance of NeRV compared to SNE, where λ equals one, which is the case when NeRV is equivalent to SNE, the trustworthiness attains the lowest performance. Thus, NeRV shows the capability of producing a high trustworthy visualization result which is based on balancing the tradeoff between the continuity and trustworthiness of visualization. A large value of λ , close to one, gives a lower trustworthy result and vice versa.

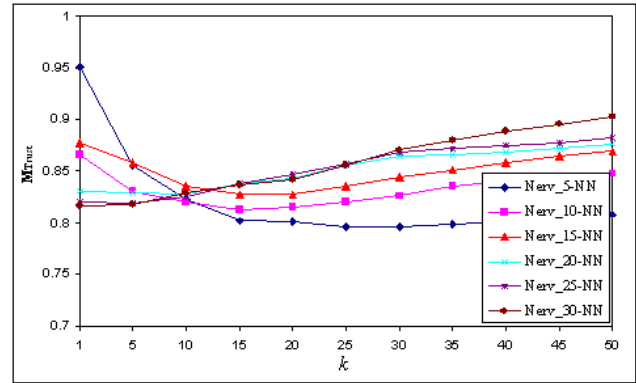


Figure 3: Trustworthiness of NeRV mapping as a function of k that applied to SNP dataset, where k is the size of neighbourhood. The neighbourhood size used by NeRV is ranging from 5 to 30.

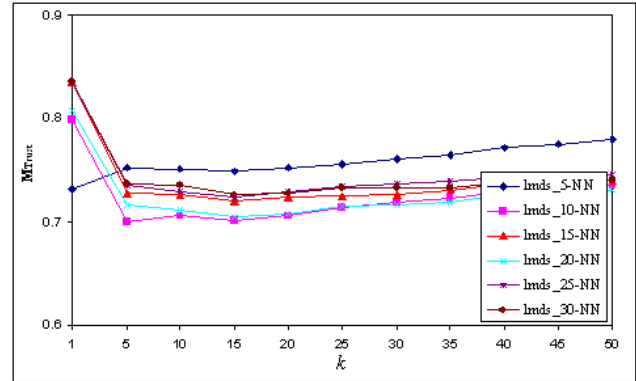


Figure 4: Trustworthiness of LocalMDS mapping as a function of k that applied to SNP dataset, where k is the size of neighbourhood. The neighbourhood size used by LocalMDS is ranging from 5 to 30.

5.2.2 Euclidean distance and Gaussian function

In our experiments two types of dissimilarity measure were used: Euclidean distance and Gaussian function. In the case of Gaussian function a parameter σ is used as a control parameter. The parameter σ was set to 0.001, 0.01, 0.1, 0.5, 1, 10, 100, 200, 500, 1000 and 2000 for different runs. The settings of $\sigma = 0.1$ and 0.5 gave the best performance (result not shown). On

our data set the use of Euclidean distance seems to give slightly similar results to the Gaussian function. This is can be due to the high-dimensionality of the data.

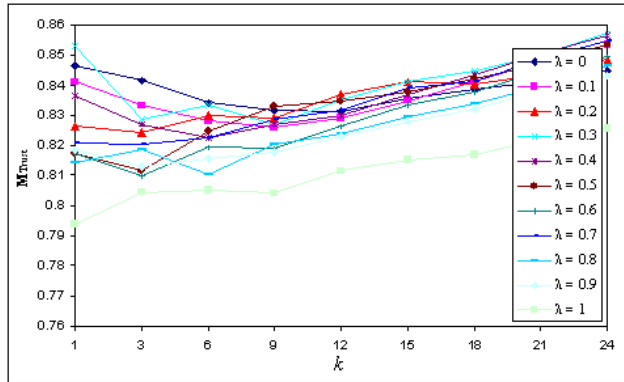


Figure 5: Trustworthiness of NeRV mapping as a function of k that applied to SNP dataset, where k is the size of neighbourhood. The Lambda used by NeRV is ranging from 5 to 30.

5.2.3 Quality of the visualization

The first thought that comes to mind is that reducing the dimensionality of the data from 13917 down into just two dimensions will not show how the data points are similar to each other in the high-dimensional space. On the other hand, without the use of a quality measure, we will not be able to assess the quality of different data reduction methods. In the previous sections, we summarized the different data reduction methods that we have employed, with different parameter settings. The comparison of the produced results was calculated in term of trustworthiness and continuity measures. The best method was selected based on the balance between these two measures and more emphasis was put on the trustworthiness measure.

Based on the results of the methods and parameter settings, the neighbour Retrieval Visualizer method, with k nearest neighbour equal to 15 and $\lambda = 0.1$, produced the best result. Figure 6 shows the visualization of data with the NeRV method. From the visualization we can see different clusters of data points (patients). The left most point in Figure 6 marks two outliers of patients sitting on top of one another. The clusters of patients require further scrutiny by domain experts. In contrast, the visualization produced by LocalMDS, as can be seen in figure 7, does not show any kind of structure on the data. This result confirms the ability of NeRV to produce better results.

6 Summary and discussions

In Summary, different data reduction methods were utilized for visualizing genetic variation (SNP) data as a way to discover the underlying relationships between patients. State-of-art data reduction methods have been employed. The result was selected based on the trustworthiness of the visualizations. The task of visualization was formulated as an information retrieval problem where the result of the visualization describes the local structure of the data. The quality measure of the visualization is tested based on a quantitative error of the number of misses and false positives.

We tested several different dimensionality reduction methods. These include PCA, Laplacian Eigen-

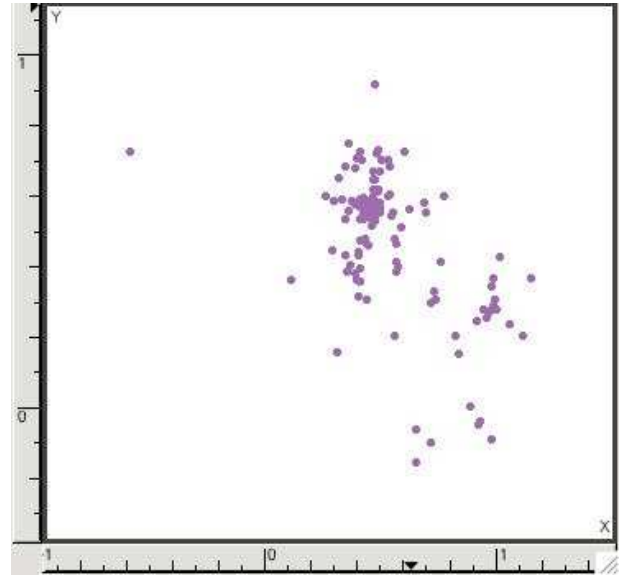


Figure 6: Visualization of SNP data using NeRV method with $k = 15$ and $\lambda = 0.1$.

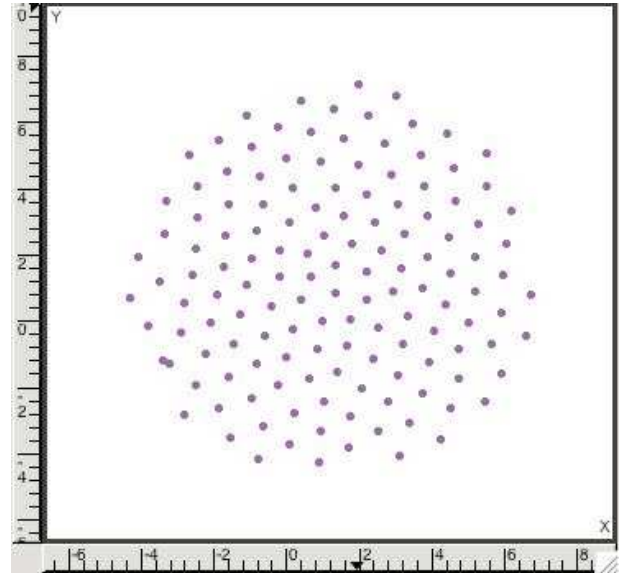


Figure 7: Visualization of SNP data using LocalMDS method with $k = 15$ and $\lambda = 0.2$

map (LE) and Locally Linear Embedding (LLE) that are designed to extract data manifolds and more generally methods that are designed for dimensionality reduction, Stochastic neighbour Embedding (SNE) and Curvilinear Component analysis (CCA). Recently, extended version of the SNE and CCA methods called neighbour Retrieval Visualizer and Local MDS methods, respectively, were introduced. These data reduction methods were run on the data with different parameter settings. An extended method of Stochastic neighbour Embedding (SNE) called neighbour Retrieval Visualizer (NeRV) has shown the best performance on this data set. This method balances the tradeoff between the trustworthiness and continuity of the visualization. The result shows that a neighbourhood of size 15 was the best for our data. A parameter λ which controls the tradeoff between trustworthiness and continuity was selected to be 0.1. This parameter emphasizes the trustworthiness of the visualization which is more important for visualization.

The result did not show any differences be-

tween using different distance matrices (Euclidean and Gaussian function) due to high-dimensionality of the data. Manifold-based data reduction methods, i.e. LLE and LE, perform surprisingly badly and the Laplacian Eigenmap method similarly performs worse as random mapping of the data. This result was not expected for these methods due to the high performance of these methods in other datasets. We hypothesize that the reason is that these methods are designed to discover the intrinsic dimensionality of the data manifold which can be more than two dimensions. Lastly, the performance of PCA was comparable to the result of LocalMDS although the latter method is considered as a nonlinear dimensionality reduction method. This behavior suggests that the dataset has a linear relationship, which is difficult to comprehend due to the high-dimensionality of the data.

7 Conclusion and Future Work

In this paper, we employed several data reduction methods for visualizing a biomedical dataset. This dataset describes the genetic variation of Acute Lymphoblastic leukaemia patients. Visualization approaches were compared based on the trustworthiness metric of the resultant visualization. To deal with large amounts of genetic variation data, we have chosen to compare the performance of different dimensionality reduction methods on the given dataset. Based on this comparison neighbour Retrieval Visualizer (NeRV) showed the best results and outperformed other methods. Even though the dimensionality of the dataset was reduced from 13917 to 2 dimensions, the quality measure of the visualization (i.e. NeRV) still shows excellent results. The visualization results will assist clinicians and biomedical researchers in understanding the different structure of patients and how to compare different group of clustering in the visualization.

The result from using NeRV shows the feasibility of this method in visualizing genetic variation data. The main limitation of the employed methods is the distance measure that has been used (Euclidean distance or Gaussian function). These methods might not be appropriate for the given dataset due to the high-dimensionality of the data. The future direction of this work is to employ other distance measures that can be more appropriate in discriminating the major characteristics of the dataset. In particular, prior knowledge or domain-driven dissimilarity measures may improve the performance of the data reduction methods in the examined dataset.

8 Acknowledgments

This work was supported by the Australian Rotary Health Research Fund (ARHRF). ARHRF/District 9680 Funding partners scholar.

References

- Altshuler, D., Brooks, L., Chakravarti, A., Collins, F., Daly, M., Donnelly, P. et al. (2005), 'A haplotype map of the human genome', *Nature* **437**(7063), 1299–1320.
- Aplenc, R. & Lange, B. (2004), 'Pharmacogenetic determinants of outcome in acute lymphoblastic leukaemia', *British Journal of Haematology* **125**(4), 421–434.
- Azuaje, F. & Dopazo, J. (2005), *Data analysis and visualization in genomics and proteomics*, Hoboken, NJ: John Wiley & Sons, Ltd.
- Barker, D., Hansen, M., Faruqi, A., Giannola, D., Irsula, O., Lasken, R., Latterich, M., Makarov, V., Oliphant, A., Pinter, J. et al. (2004), 'Two Methods of Whole-Genome Amplification Enable Accurate Genotyping Across a 2320-SNP Linkage Panel', *Genome Research* **14**(5), 901.
- Belkin, M. & Niyogi, P. (2002), 'Laplacian eigenmaps and spectral techniques for embedding and clustering', *Advances in Neural Information Processing Systems* **14**, 585–591.
- Belkin, M. & Niyogi, P. (2004), 'Semi-Supervised Learning on Riemannian Manifolds', *Machine Learning* **56**(1), 209–239.
- Bengio, Y., Païement, J. & Vincent, P. (2004), 'Out-of-Sample Extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering', *Advances in Neural Information Processing Systems 16: Proceedings of the 2003 Conference*.
- Bertone, P. & Gerstein, M. (2001), 'Integrative data mining: the new direction in bioinformatics', *Engineering in Medicine and Biology Magazine, IEEE* **20**(4), 33–40.
- Buchala, S., Davey, N., Frank, R. & Gale, T. (2004), 'Dimensionality reduction of face images for gender classification', *Intelligent Systems, 2004. Proceedings. 2004 2nd International IEEE Conference* **1**.
- Carlson, C., Eberle, M., Kruglyak, L. & Nickerson, D. (2004), 'Mapping complex disease loci in whole-genome association studies', *Nature* **429**, 446–452.
- Carlson, C., Eberle, M., Rieder, M., Smith, J., Kruglyak, L. & Nickerson, D. (2003), 'Additional SNPs and linkage-disequilibrium analyses are necessary for whole-genome association studies in humans', *Nature Genetics* **33**(4), 518–521.
- Cavalli-Sforza, L. (1974), 'The genetics of human populations.', *Sci Am* **231**(3), 80–9.
- Coates, M. & Tracey, E. (2001), 'Cancer in New South Wales. Incidence and mortality 1998 and Incidence for Selected Cancers 1999', *NSW Central Cancer Registry, Cancer Research and Registers Division, NSW Cancer Council* pp. 42–43.
- Costa, J. & Hero, A. (2005), 'Classification Constrained Dimensionality Reduction', *Proceedings (ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing* **5**.
- Cox, T. & Cox, M. (2001), *Multidimensional Scaling*, CRC Press.
- Crawford, D. & Nickerson, D. (2005), 'Definition and Clinical importance of Haplotypes', *Annual Review of Medicine* **56**(1), 303–320.
- Demartines, P. & Herault, J. (1997), 'Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets', *Neural Networks, IEEE Transactions on* **8**(1), 148–154.
- Donnelly, J. (2004), 'Pharmacogenetics in Cancer Chemotherapy: Balancing Toxicity and Response.', *Therapeutic Drug Monitoring* **26**(2), 231.

- Erichsen, H. & Chanock, S. (2004), 'SNPs in cancer research and treatment', *British Journal of Cancer* **90**, 747–751.
- Fan, J. et al. (2003), 'Highly Parallel SNP Genotyping', *Cold Spring Harbor Symposia on Quantitative Biology* **68**(1), 69–78.
- Gower, J. (1966), 'Some distance properties of latent root and vector methods used in multivariate analysis', *Biometrika* **53**(3-4), 325–338.
- He, X. & Niyogi, P. (2004), 'Locality Preserving Projections', *Advances in Neural Information Processing Systems 16: Proceedings of the 2003 Conference*.
- He, X., Yan, S., Hu, Y., Niyogi, P. & Zhang, H. (2005), 'Face Recognition Using Laplacianfaces', *IEEE Transaction on pattern analysis and machine intelligence* pp. 328–340.
- Herr, A., Grützmann, R., Matthaei, A., Artelt, J., Schröck, E., Rump, A. & Pilarsky, C. (2005), 'High-resolution analysis of chromosomal imbalances using the Affymetrix 10K SNP genotyping chip', *Genomics* **85**(3), 392–400.
- Hinton, G. & Roweis, S. (2003), 'Stochastic neighbor embedding', *Advances in Neural Information Processing Systems* **15**, 833–840.
- Hirschhorn, J. & Daly, M. (2005), 'Genome-wide association studies for common diseases and complex traits', *Nature Reviews Genetics* **6**(2), 95–108.
- Hotelling, H. (1933), 'Analysis of a complex of statistical variables into principal components', *Journal of Educational Psychology* **24**(6), 417–441.
- Irving, J., Bloodworth, L., Bown, N., Case, M., Hogarth, L. & Hall, A. (2005), 'Loss of Heterozygosity in Childhood Acute Lymphoblastic Leukemia Detected by Genome-Wide Microarray Single Nucleotide Polymorphism Analysis'.
- Kaski, S., Nikkila, J., Oja, M., Venna, J., Toronen, P. & Castren, E. (2003), 'Trustworthiness and metrics in visualizing similarity of gene expression', *BMC Bioinformatics* **4**(1), 48.
- Kokopoulou, E. & Saad, Y. (2005), 'Orthogonal Neighborhood Preserving Projections', *IEEE Int. Conf. on Data Mining* pp. 1–8.
- Kokopoulou, E. & Saad, Y. (2007), 'Orthogonal Neighborhood Preserving Projections: A Projection-Based Dimensionality Reduction Technique', *IEEE Transactions on pattern analysis and machine intelligence* pp. 2143–2156.
- Kruskal, J. (1964), 'Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis', *Psychometrika* **29**(1), 1–27.
- Lee, J., Lendasse, A. & Verleysen, M. (2004), 'Nonlinear projection with curvilinear distances: Isomap versus curvilinear distance analysis', *Neurocomputing* **57**, 49–76.
- Leykin, I., Hao, K., Cheng, J., Meyer, N., Pollak, M., Smith, R., Wong, W., Rosenow, C. & Li, C. (2005), 'Comparative linkage analysis and visualization of high-density oligonucleotide SNP array data', *feedback*.
- Memisevic, R. & Hinton, G. (2005), 'Improving dimensionality reduction with spectral gradient descent', *Neural Networks* **18**(5-6), 702–710.
- Ng, A., Jordan, M. & Weiss, Y. (2002), 'On spectral clustering: Analysis and an algorithm', *Advances in Neural Information Processing Systems 14: Proceedings of the 2002 [sic] Conference*.
- Nguyen, G. & Worring, M. (2004), 'Optimizing similarity based visualization in content based image retrieval', *Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on* **2**.
- Price, A., Patterson, N., Plenge, R., Weinblatt, M., Shadick, N. & Reich, D. (2006), 'Principal components analysis corrects for stratification in genome-wide association studies', *Nature Genetics* **38**, 904–909.
- Reich, D., Gabriel, S. & Altshuler, D. (2003), 'Quality and completeness of SNP databases', *Nature Genetics* **33**(4), 457–458.
- Risch, N. & Merikangas, K. (1996), 'The Future of Genetic Studies of Complex Human Diseases', *Science* **273**(5281), 1516.
- Roweis, S. & Saul, L. (2000), 'Nonlinear Dimensionality Reduction by Locally Linear Embedding'.
- Shi, J. & Malik, J. (2000), 'Normalized Cuts and Image Segmentation', *IEEE Transaction on pattern analysis and machine intelligence* pp. 888–905.
- Tabor, H., Risch, N. & Myers, R. (2002), 'Candidate-gene approaches for studying complex genetic traits: practical considerations.', *Nat Rev Genet* **3**(5), 391–7.
- Tagaris, G., Richter, W., Kim, S., Pellizzer, G., Andersen, P., Ugurbil, K. & Georgopoulos, A. (1998), 'Functional magnetic resonance imaging of mental rotation and memory scanning: a multidimensional scaling analysis of brain activation patterns', *Brain Research Reviews* **26**(2-3), 106–112.
- Tan, P., Steinbach, M. & Kumar, V. (2005), *Introduction to Data Mining*, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.
- Torgerson, W. (1952), 'Multidimensional scaling: I. Theory and method', *Psychometrika* **17**(4), 401–419.
- van der Maaten, L., Postma, E. & van den Herik, H. (2007), 'Dimensionality reduction: A comparative review', *Disponibile sur internet*.
- Venkatarajan, M. & Braun, W. (2004), 'New quantitative descriptors of amino-acids based on multidimensional scaling of a large number of physicochemical properties', *Journal Molecular Modeling* **7**(12), 445–453.
- Venna, J. & Kaski, S. (2001), 'Neighborhood preservation in nonlinear projection methods: An experimental study', *Artificial Neural Networks—ICANN* pp. 485–491.
- Venna, J. & Kaski, S. (2006), 'Local multidimensional scaling', *Neural Networks* **19**(6-7), 889–899.
- Venna, J. & Kaski, S. (2007a), 'Comparison of visualization methods for an atlas of gene expression data sets', *Information Visualization* **6**(2), 139–154.
- Venna, J. & Kaski, S. (2007b), 'Nonlinear Dimensionality Reduction as Information Retrieval', *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics (AISTATS* 07)* pp. 568–575.

Classification of Brain-Computer Interface Data

Omar AlZoubi^{1,2}, Irena Koprinska¹ and Rafael A. Calvo²

¹School of Information Technology

²School of Electrical and Information Engineering

University of Sydney, NSW 2006, Australia

oalz5092@mail.usyd.edu.au, irena@it.usyd.edu.au, rafa@ee.usyd.edu.au

Abstract

In this paper we investigate the classification of mental tasks based on electroencephalographic (EEG) data for Brain Computer Interfaces (BCI) in two scenarios: off line and on-line. In the off-line scenario we evaluate the performance of a number of classifiers using a benchmark dataset, the same pre-processing and feature selection and show that classifiers that haven't been used before are good choices. We also apply a new feature selection method that is suitable for the highly correlated EEG data and show that it greatly reduces the number of features without deteriorating the classification accuracy. In the on-line scenario that we have designed, we study the performance of our system to play a computer game for which the signals are processed in real time and the subject receives visual feedback of the resulting control within the game environment. We discuss the performance and highlight important issues.

Keywords: classification of EEG data, brain-computer interfaces, correlation-based feature selection

1 Introduction

A BCI is a system which allows a person to control special computer applications (e.g. a computer cursor or robotic limb) by only using his/her thoughts. The idea is to provide a new communication channel to people who are paralyzed but are cognitively intact, e.g. people suffering from the so called locked-in syndrome. BCIs have been a very active area of research, especially over the past ten years (Wolpaw et al. 2000, Dornhege et al. 2007). The research is based on recording and analyzing EEG brain activity and recognizing EEG patterns associated with mental states. For example, imagining a movement of the right hand is associated with a pattern of EEG activity in the left side of the motor cortex. Other frequently used mental tasks are the movement of the left hand, movement of the toes and movement of the tongue. Mental tasks are carefully chosen so that they activate different parts of the brain, which makes them easier to detect.

The increasing success of BCI systems is partially due to a better understanding of the dynamics of brain

oscillations that generate EEG signals. In the brain, networks of neurons form feedback loops responsible for the oscillatory activity recorded in the EEG. Normally the frequency of such oscillations becomes slower with increased synchronization. Sensorimotor activity such as body movements or mental imagery (e.g. imagining body movement) changes the oscillatory patterns resulting in amplitude suppression called event related desynchronization or amplitude enhancement called event related synchronization on the Rolandic mu rhythm (7-13 Hz) and the central beta rhythms above 13 Hz. This phenomenon has been known since the 1940's (Jasper and Penfield 1949).

Supervised classification methods are employed to learn to recognize these patterns of EEG activities, i.e. to learn the mapping between the EEG data and classes corresponding to mental tasks such as movement of the left hand (Lotte et al. 2007). From data mining point of view this is a difficult learning task due to two main reasons. Firstly, the EEG data is noisy and correlated as many electrodes need to be fixed on the small scalp surface and each electrode measures the activity of thousands of neurons (Lee et al. 2005; Thulasidas, Guan, and Wu 2006)). Selecting the optimal frequency band and extracting a good set of features are still open research problems. Secondly, the quality of the data is affected by the different degree of attention of the subject and changes in their concentration.

Traditionally, classical linear classifiers such as the Fisher's linear discriminant have been favoured (Lotte et al. 2007; Blankertz et al. 2001, Müller et al. 2004). More recently, a variety of machine learning classifiers have been applied, e.g. neural networks such as multi-layer perceptrons (Anderson and Sijercic 1996; Kubat, Koprinska and Pfurtscheller 1998), probabilistic classifiers (Barreto, Frota and Medeiros 2004), lazy learning classifiers such as k-nearest neighbor (Blankertz, Curio and Müller 2002) and state of the art classifiers such as support vector machines (Lee et al. 2005). However, as noted by Lotte et al. (2007), it is hard to evaluate these classifiers as the experimental setup, the pre-processing and feature selection are different in the reported studies. In addition, Lotte et al. (2007) also note that some of the classical classification algorithms such as decision trees and also ensembles of classifiers haven't been evaluated. Thus, the first goal of our study is to evaluate a variety of classification techniques on a multi-class BCI classification task, using a benchmark dataset, and also under the same conditions, i.e. using the same pre-processing and feature selection methods.

For this purpose we used the BCI2000 (Schalk and McFarland 2004) and Weka (Witten and Frank 2005) software and also data from the latest BCI competition

(BCI III) (Blankertz et al. 2006). The BCI2000 is a recently developed publicly available software platform for EEG data recording and signal processing. Weka is a publicly available Java-based open-source library for machine learning and data mining. We have integrated the WEKA's classifiers in BCI2000 which allows for evaluation of a wide range of classifiers. On the other hand, the BCI competitions provide publicly available datasets recorded at the leading BCI laboratories and can be used for benchmark evaluation which hasn't been done. We chose dataset IIIa from the BCI III competition, a four class problem.

We also introduce a new pre-processing and feature selection method that is appropriate for the highly correlated and noisy EEG data. It is based on common spatial patterns and correlation-based feature selection. We evaluate the performance of 13 classifiers and compare the results with the top three competition results.

While the classification of BCI competition data is an off-line task, our second goal is to evaluate the performance of our BCI system (using the most successful classifiers from the previous off-line task) in an online experiment. We chose the simple pong computer game. A vertical panel (target) appears in either the right or left side of the screen, and a ball appears in the middle of the screen. The goal is to move the ball towards the target. Three subjects took part in this experiment. We discuss the performance of our system and highlight important issues.

Thus, the contribution of our paper can be summarised as follows:

- We propose a new pre-processing and feature extraction method appropriate for the noisy and correlated nature of the EEG data.
- We integrated BCI2000 with Weka and evaluated a number of classification algorithms using the same experimental setup, pre-processing and feature selection, and also using a benchmark dataset from the BCI competition. We also compared the performance against the BCI competition submissions.
- We designed and conducted an online experiment to evaluate our BCI system in a realistic application.

The paper is organised as follows. Section 2 presents the off-line scenario, i.e. the classification of the BCI competition data. Section 3 presents the on-line classification of BCI data. We describe the datasets, pre-processing and feature selection, present and discuss the results. Section 4 concludes the paper and suggests avenues for future work.

2 Task 1: Off-Line Classification of BCI Competition Data

2.1 Data Acquisition

We used dataset IIIa from the BCI III competition (BCI Competition III 2008). It contains data from 3 subjects: K3b, K6b and L1b and was collected as follows (Schlögel 2005). Each subject, sitting in front of a computer, was asked to perform imaginary movements of the left hand, right hand, tongue or foot during a pre-specified time interval. As mentioned before, when a person imagines such movements, there are associated changes in the EEG

data called event-related synchronization or de-synchronization. 60 electrodes were placed on the scalp of the subject recording a signal sampled at 250 Hz and filtered between 1 and 50 Hz using a Notch filter.

Each trial starts with a blank screen. At $t=2s$, a beep is generated and a cross "+" is shown to inform the subject to pay attention. At $t=3s$ an arrow pointing to the left, right, up or down is shown for 1s and the subject is asked to imagine a left hand, right hand, tongue or foot movement, respectively, until the cross disappears at $t=7s$. This is followed by a 2s break, and then the next trial begins. For each subject 60 trials per class were recorded.

Two data files are available for each subject: training and testing.

2.2 Pre-processing and Feature Selection

Firstly, we applied the Common Spatial Patterns (CSP) method (Müller-Gerking, Pfurtscheller and Flyvbjerg 1999) to the raw EEG data. The standard CSP is applicable to two class problems; it transforms the original signal into a new space where the variance of one of the classes is maximised while the variance of the other is minimized. We used an extension for more than two classes by considering one class versus the rest. The result of the application of CSP to the original 60 signals, for each class versus the others, is a new set of 60 signals ordered based on how informative they are to predict the class. We selected the first 5 projections which resulted in 20 signals (5 channels \times 4 projections). Then we applied 3 frequency band filters for 8-12 Hz, 21-20 Hz, 20-30 Hz. Finally, we extracted 7 features: max, min and mean voltage values, voltage range, number of samples above zero volts, zero voltage crossing rate and average signal power. This resulted in 420 ($5 \times 4 \times 3 \times 7$) discrete numeric features for each subject.

The number of instances in the training and test sets was equal for all subjects and was 180 for K3b, 120 for K6b and 120 for L1b. Each instance was labelled with one of the four classes and the distribution of the classes was equal in both training and test data. Thus, given a set of 120 or 180 instances labelled into 4 classes, each of them 420 dimensional, the goal is to build a classifier for each subject able to distinguish between the 4 classes. This highlights another difficulty in classifying BCI data: the curse of dimensionality - small number of training instances but highly dimensional. It is generally accepted that the number of training instances per class should be at least 10 times more than the features and that more complex classifiers require a larger ratio of sample size to features (Jain, Duin and Mao 2000).

Good feature selection is the key to the success of a classification algorithm. It is needed to reduce the number of features by selecting the most informative and discarding the irrelevant and redundant features. As EEG data is known to be highly correlated, a feature selection method which exploits this property seems appropriate. We applied a simple, fast and efficient method, called Correlation-Based Feature Selection (CFS) (Hall 2000). It searches for the "best" sub-set of features where "best" is defined by a heuristic which takes into consideration 2 criteria: 1) how good the individual features are at predicting the class and 2) how much they correlate with

the other features. Good subsets of features contain features that are highly correlated with the class and uncorrelated with each other. The search space is very big for employing a brute-force search algorithm. We used the best first (greedy) search option starting with an empty set of features and adding new features. It is important also to note that the feature selection was done using the training data only.

As a result, 56 features were selected for B3b, 19 for K6b and 15 for L1b, which is a feature reduction of 86.7% for B3b, 95.5% for K6b and 96.4% for L1b. Thus, this drastic feature reduction confirms that the BCI data is highly correlated. It also reduces the effect of the curse of dimensionality: the ratio of the number of instances per class to the number of features is reduced from 45/420 to 45/56 for K3b, from 30/420 to 30/19 for K3b and from 30/420 to 30/15 for L1b.

2.3 Classifiers

We evaluated 13 classifiers using their WEKA's implementations (Witten and Frank 2005). They are summarized in Table 1.

	classifier	Description and parameters
1	ZeroR	Predicts the majority class in the training data; used as a baseline.
2	1R	A rule based on the values of 1 attribute i.e. one level decision tree, (Holte 1993).
3	Decision Tree (DT)	A classical divide and conquer learning algorithm (Quinlan 1986). We used J48.
4-5	K Nearest Neighbor (k-NN)	A classical instance-based algorithm (Aha and Kibler 1991); uses normalised Euclidean distance. We used k=1 and 5.
6	Naïve Bayes (NB)	A standard probabilistic classifier.
7	Radial-bases Network (RBF)	A 2-layer network. Uses Gaussians as basis functions in the first layer (number and centers set by the k-means algorithm) and a linear second layer and learning algorithm (Moody and Darken 1989).
8	Support Vector Machine (SVM)	Finds the maximum margin hyperplane between 2 classes. We used Weka's SMO with polynomial kernel. SMO is based on Platt's optimisation algorithm (Platt 1998).
9	Logistic Regression (LogReg)	Standard linear regression. Weka's implementation is based on (Le Cessie and van Houwelingen 1992).
10	Ada Boost	An ensemble of classifiers. It produces a series of classifiers iteratively; new classifiers focus on the instances which were misclassified by the previous classifiers; uses weighed vote to combine individual decisions (Freund and Shapire1996). We used boosting of 10 decision trees (J48).
11	Bagging	An ensemble of classifiers. Uses random sampling with replacement to generates training sets for the classifiers; decisions are combined with majority vote (Breiman 1996). We combined 10 decision trees (J48).
12	Stacking	A 2 level ensemble of classifiers (Wolpert 1992). We used 1-NN, NB and DT (J48) as level-1 classifiers and 1-NN as a level-2 classifier.
13	Random Forest (RF)	An ensemble of decision trees based bagging and random feature selection (Breiman 2001). We used t=10 trees.

Table 1: Classifiers used

2.4 Results and Discussion

Figures 1, 2 and 3 show the classification results in terms of accuracy on the test set for the three subjects, under 2 conditions: without and with CFS feature selection. More specifically, each of the 13 classifiers was trained on the training set and tested on the test set; and the accuracy on the test set is reported. The testing set was not used in any way during the training or feature selection.

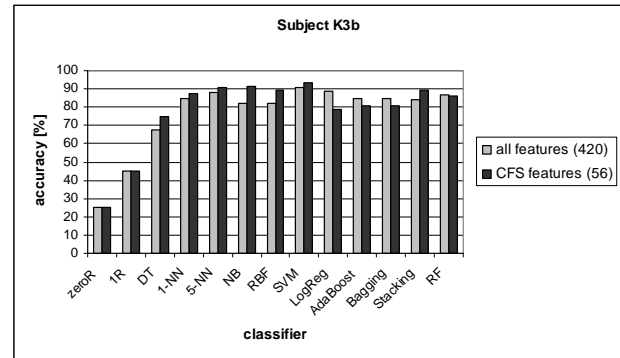


Figure 1: BCI competition data IIIa, subject K3b - accuracy on test set [%] for various classifiers

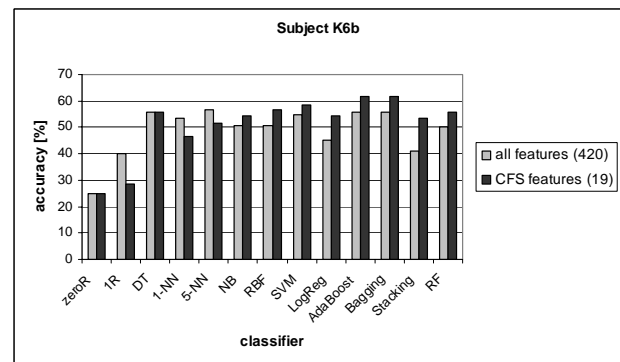


Figure 2: BCI competition data IIIa, subject K6b - accuracy on test set [%] for various classifiers

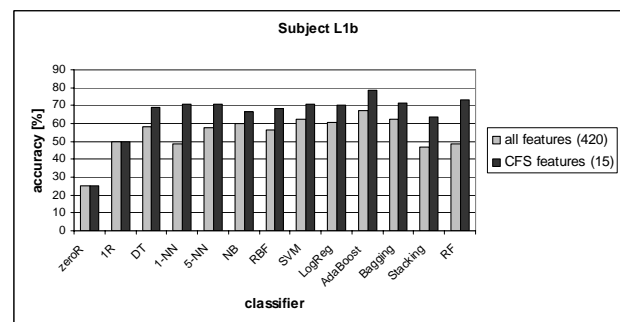


Figure 3: BCI competition data IIIa, subject L1b - accuracy on test set [%] for various classifiers

It can be seen that all classifiers, with and without feature selection, outperform the baseline (ZeroR's 25% accuracy). Comparing across the subjects, the accuracy is highest for K3b and lowest for K6b. This is consistent with Lee (2005) who note that the three subjects have different amount of experience in BCI training, with K3 being the most experienced subject, L1 having little experience and K6 being a beginner.

The results from Figures 1-3 also show that CFS was a very successful feature selector for all subjects despite the fact that it was very aggressive and discarded a large number of features. It improved or maintained the accuracy for all classifiers except the following 3: LogReg, AdaBoost and Bagging for K3b.

Table 2 shows our best results and the results of the top 3 competition submissions. They were achieved using CFS feature selection and SVM for K3b, AdaBoost and Bagging for K6b and AdaBoost for L1b. As it can be seen our results are the second best for each subject, thus they are comparable with the best submitted results.

BCI team	K3b	K6b	L1b
Hill & Schröder (resampling 100Hz, detrending, Informax ICA, Welch amplitude spectra, linear PCA, SVM)	96.11	55.83	64.17
Guan, Zhang & Li Fisher ratios of channel-frequency-time bins, feature selection, mu- and beta- passband, multiclass CSP, SVM)	86.67	81.67	85.00
Gao, Wu & Wei (surface laplacian, 8-30Hz filter, multi-class CSP, SVM+kNN+LDA)	92.78	57.50	78.33
Ours (multi class CSP, CFS)	93.89 (SVM)	61.67 (Ada Boost or Bagging)	78.33 (Ada Boost)

Table 2: BCI competition data IIIa – comparison between the 3 top competition submissions as reported in Blankertz et al. (2006) and our best results, accuracy [%] on test set

In the rest of this section we focus on the approach with feature selection and discuss the results in more details. The top three classifiers were as follows:

- for K3b: SVM (accuracy on test set: 93.33%), NB (91.67%) and 5-NN (90.56%)
- for K6b: AdaBoost and Bagging (both 61.67%) and SVM (58.33%)
- for L1b: AdaBoost (78.33%), Stacking (73.33%) and Bagging (71.67%).

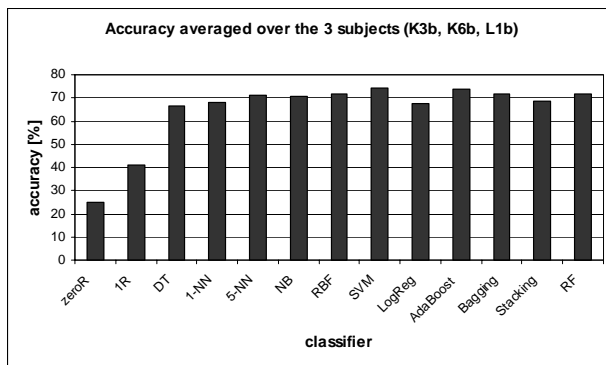


Figure 4: BCI competition data IIIa, accuracy on test set [%] averaged over the 3 subjects for various classifiers using CFS feature selector

Figure 4 shows the average accuracy on test set across the 3 subjects. The best overall classifier was SVM (74.16%), closely followed by AdaBoost (73.70%), RF (71.76%), Bagging and RBF (both 71.48%). Thus, our results confirm the good generalization ability of SVM but they also show that other algorithms such as ensembles of classifiers and RBF networks produce similar results. Below we discuss each of these 4 classifiers.

SVM were previously shown to classify successfully BCI data but comparisons are difficult as the data, pre-processing and experimental setups were different. The same data was used by Schlögel et al. (2005), the team which collected the BCI IIIa competition data. They applied different pre-processing (adaptive autoregressive processing) extracting 180 features and then compared SVM with linear discriminant analysis (LDA) and k-NN and found SVM to significantly outperform the other 2 algorithms achieving accuracy of 77.24% for K3b, 52.4% for K6b and 53.9% for L1b. This accuracy is significantly lower than ours and BCI competition results, especially for the first and third subjects and it may be due to the different experimental evaluation: they used leave-one-out cross validation while we and the BCI competition participants used 1 training/test run, consistent with the rules. In addition, the reason why SVM performed well in their case may be due to the high number of features they retained (180 as opposite to 15-56 in our case) – dealing with high dimensional data is one of the main strengths of SVM and weakness of k-NN.

In our experiments AdaBoost, Bagging and RF were ensembles of decision trees and the results showed that they outperformed the single decision tree classifier with 2-11% (the only exception was RF on K6b which achieved the same accuracy as DT). Boosting of decision trees is a highly successful classifier, frequently used for comparison in machine learning. Bagging is a less complex and faster than boosting; there is empirical evidence (Dietterich 2000, Opitz 1999) that it is more robust to noisy data than bagging which may explain its good performance on the BCI data. RF combines bagging with random feature selection. It is RF is faster than DT as it considers less number of features when selecting an attribute to split on and also does not prune the trees. Our results are also consistent with Breiman (1996) who showed that RF runs faster than AdaBoost and gives comparable accuracy results.

While RF and Bagging hasn't been used previously to classify BCI data, Boosting of backpropagation neural networks was used by Boostani and Moradi (2004) and was found to be significantly outperformed by linear discriminant analysis. Again, comparison is not possible as the task was different (two imaginary movement classes), the datasets were different and the feature extraction was different (based on band power, Hjort parameters and fraction dimension).

We also found RBF to be a successful classifier. It is a powerful nonlinear classifier, fast to train (in contrast to the slow training of the backpropagation network), accurate in classification and tolerant to noise (similarly to the backpropagation network). While the backpropagation networks have been widely used for

classification of BCI data, RBF hasn't received enough attention. There seem to be only one published study (Hoya et al. 2003) in the context of BCI data classification - letter imaginary tasks, with principle component analysis and independent component analysis for feature selection and RBF for classification.

In terms of training time, SVM was the slowest classifier (e.g. 1.03s to build a classifier for K3B) as the 4-class problem is decomposed into 4 binary problems, followed by AdaBoost (0.42s) and the remaining 3 classifiers (0.04-0.19s). In general, the current BCI applications are trained off-line which means that accuracy is more important than training time; they require fast classification of new data which is true for all classifiers except lazy classifiers such as k-NN. However, the need to incrementally retrain the classifier to adapt to the incoming data or subject is recognised as one of the desirable features of the future BCI applications, in which case the training time and the development of incremental versions of the algorithms become very important.

In summary, our experiments show that: 1) CFS is an appropriate and successful feature selector for classification of BCI data; 2) SVM, ensembles of classifiers such as AdaBoost, Bagging and RF of decision trees, and also RBF were the best classifiers. While SVM has been widely used in previous BCI data classification, the remaining 4 classifiers haven't received enough attention although they have many attractive properties; 3) Our classification results are comparable with the top BCI competition results.

3 Task 2: On-line Classification of BCI Data for Playing the Pong Game

There are two stages in this task: 1) collecting data and building classifiers and 2) playing of the pong game on-line using these trained classifiers. Below we discuss the experimental setup and data acquisition for each of them.

3.1 Collecting Data and Building Classifiers

Firstly, we need to collect labelled data that will be used as training data to build a classifier able to recognise "move the cursor right" from "move the cursor left". This classifier will be then used to play pong on line, i.e. given an EEG signal, it will classify it as one of the two movements and the cursor will be moved accordingly.

The training data has been collected as follows. The subjects sit in front of a computer in a relaxing chair with armrests. They wear EEG recording cap; for this experiment we used three electrodes as described below. In each trial, a cursor (ball) appears in the centre of the screen, and a target (vertical panel) in either the left or right side. The task for the subjects is to move the cursor to the target within a given time (4-5s) by imagining such movement to the right and left (e.g. by imagining movement of the right and left hand, respectively). Thus, we collect data associated with imaginary movement to the right and left, and it is labelled with the correct class based on the target.

Data from 3 subjects was collected: So, Si and Sp. The subjects were firstly given a few minutes to play the game and this data was not recorded. Then, they took part in 1 or more sessions lasting approximately 240s for which

the data was recorded. Each session was divided into trials. Our subjects performed different number of sessions: Si – 1 session, Sp and So – 2 sessions each. Thus, all subjects are beginners, with So being the most experienced and Si the less experienced. To ensure consistency, data from one session was used as a training set; in case of multiple sessions (e.g. for So and Sp), this was the data from the last session. Each session consisted of the same number of trials.

A novel wearable dry electrode EEG recording equipment (Gargiulo et al 2008) was used for the recordings. Three electrodes were used for data acquisition, C3, C4 and Cz according to the international 10-20 system. Sampling has been done at 256Hz with a sample block size of 40Hz. A feature vector was extracted from each trial and it consisted of 11 features for each electrode. These features were derived using a standard BCI EEG data pre-processing using BCI2000 and described below. An autoregressive filter was applied and 11 coefficients were obtained for 11 bins of equal size for frequencies between 0 and 31.5 Hz (BCI2000 ARFilter). Then, a common average spatial filter (BCI2000 Spatial Filter) was applied. This type of filter was shown to produce a good signal to noise ratio and perform well in BCI mental tasks applications (McFarland et al. 1997).

This resulted in 33-dimensional datasets with 3918 instances for Si, 2174 instances for Sp and 4109 instances for So. We used this data to build and evaluate classifiers off-line for each subject.

3.2 On-line Experiment

In this experiment we used the classifiers built in the previous step to play the pong game on-line. The experimental setup and signal pre-processing were the same as before but now the classifier's prediction is used to move the cursor. More specifically, as before, the subject sits in front of the computer, the same pong game is run and the subject is instructed to imagine left and right movements in order to hit the target. This imaginary movement signal is fed to the classifier; it predicts the class (left or right), the cursor moves accordingly and the subject can see the movement i.e. the subject receives visual feedback. This feedback looks continuous to the subject, i.e. the cursor moves smoothly, although the classification of the signal is discrete.

Performance is calculated as the number of hits, i.e. how many times the target was hit. To hit the target (i.e. move the cursor from the centre to the target) several consecutive signals need to be correctly classified.

3.3 Results and Discussion

For the *off-line* evaluation of the classifiers we used 10-fold cross validation (Witten and Frank 2005). Figures 5, 6 and 7 show the accuracy results for subjects Si, Sp and So, respectively.

The following observations can be made:

1) Subjects So and Sp performed much better than subject Si. For So, all classifiers outperformed the baseline ZeroR=50.3%. For Sp all but 1 classifier (LogReg with feature selection) outperformed the baseline ZeroR=50.87%. In contrast Si was not able to

distinguish between left and right, obtaining results below or around the baseline of 61.2% in most of the cases.

This can be explained with the lack of experience of subject Si – Si didn't have any prior training. After the session Si also reported to have had difficulty to concentrate and not being able to consistently imagine the same movement for each of the mental tasks.

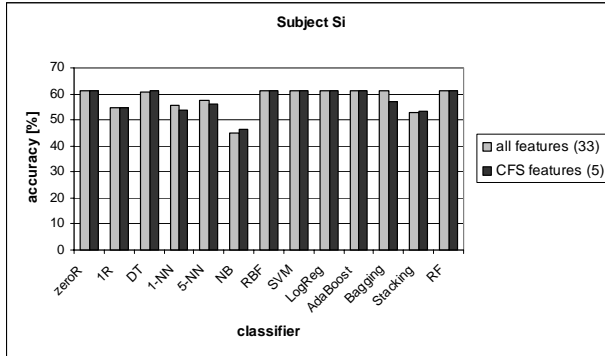


Figure 5: Off-line evaluation of the classifiers built for the pong game. Accuracy [%] using 10-fold cross-validation for subject Si

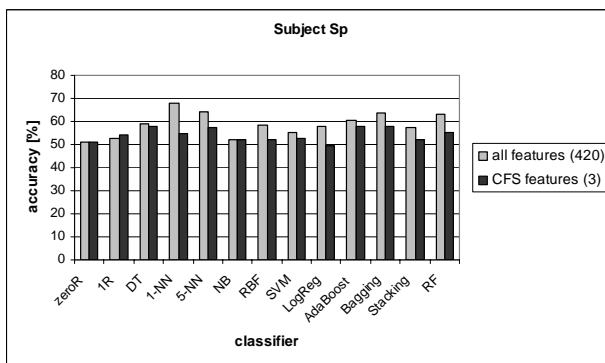


Figure 6: Off-line evaluation of the classifiers built for the pong game. Accuracy [%] using 10-fold cross-validation for subject Sp

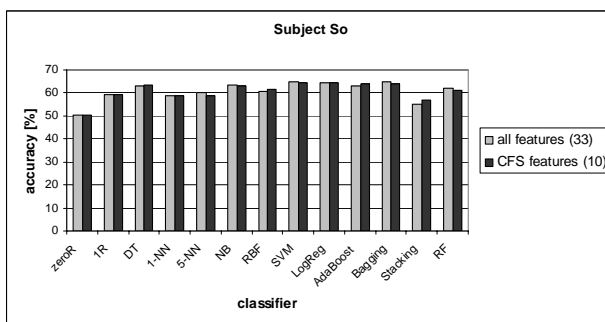


Figure 7: Off-line evaluation of the classifiers built for the pong game. Accuracy [%] using 10-fold cross-validation for subject So

2) For So the most accurate classifier was 1-NN without feature selection (67.66%). For Sp, the most accurate classifier was SVM without feature selection (64.76%), closely followed by Bagging and LogReg. These results are significantly above the baseline. The improvement over the baseline is lower in comparison to the BCI competition data (see Section 2.4), however this

is not a fair comparison as the tasks and subjects were different.

3) A closer examination of the confusion matrices revealed that there were differences in the misclassifications between the classifiers. For example, for subject So, some of the classifiers (1R, NN and Stacking) misclassified equal number of examples from the two classes; for the majority of the other classifiers class 2 was more difficult than class 1.

4) The CFS in this case was less effective than in the BCI competition data. The number of extracted features was much smaller in this case (33 versus 420). For Sp CFS further reduced the number of features to only 3 which decreased the performance while a reduction to 10 features for So maintained the performance

Table 3 summarises the results from the *on-line* experiment using 3 of the off-line trained classifiers for each subject (SVM, AdaBoost and NB) without feature selection.

Subject	AdaBoost	SVM	NB
Si	52.17	50.00	50.00
Sp	55.88	55.56	60.98
So	63.89	68.25	70.59

Table 3. Playing the pong game on-line: target hit rate (%) for subjects So, Si and Sp using different classifiers

The following observations can be made:

1) Across the subjects, the on-line results are consistent with the off-line evaluation of the classifiers: subject So achieved the best results followed by Sp and Si. Thus, more accurate classifiers (created and evaluated using previously recorded data) produced better results in the on-line experiment on new data.

2) Across the three classifiers, NB was the best classifier outperforming SVM and AdaBoost for each subject. However, it should be noted that this comparison between the three classifiers is not completely fair. As this is an on-line task, a separate session was conducted for each classifier which means that the same subject “generated” different test data for each classifier. Thus, the test data is not the same; there may be differences caused by fatigue of the subjects, electrodes’ and subjects’ artifacts.

3) A hit rate of 70% would be valuable in practical on-line application. In our experiment it was achieved by only one of our subjects but recall that all of the 3 subjects didn't receive enough training.

4) A comparison between the off-line and on-line experiments shows that the ranking of the three classifiers is different. More specifically, the classifier ranking based on average accuracy for the 3 subjects was AdaBoost, SVM, Naïve Bayes in the off-line evaluation and Naïve Bayes, SVM, AdaBoost in the on-line evaluation. Thus, the classifier that worked best in an off-line evaluation was not the best one in the on-line scenario. However, this comparison is not completely fair for 2 reasons. Firstly, as discussed above, in the on-line experiment the classifiers used different test sets which may affect their ranking. Secondly, different performance

measures are used in the off-line and on-line evaluation: accuracy and hit rate, respectively as discussed below.

It is important to note that Table 3 reports the “hit rate” which is the performance index reported by the BCI2000 software. Hit rate is the ratio of the trials when the target was hit to the total number of trials. The hit of the target is just the final step in a sequence of steps, where each step is a classification task. Thus, the hit rate is a very coarse measure of performance which doesn't tell us how the individual steps were classified. For example, a hit rate of 100% may correspond to accuracy of 80% meaning that the target was always hit but the individual steps were not always correctly classified. Or alternatively, for example, the hit rate can be low and the accuracy high when all the steps except the last one were correctly classified within the given time but the target was not hit. This highlights the need for consistency in the reporting of the BCI classification results and the use of a more informative performance index than the hit rate. Accuracy is a better choice but it doesn't provide information where the misclassifications were. Reporting the confusion matrix or using recall, precision and their combinations (e.g. the F1 measure) would be a better choice.

In summary, our experiments show that: 1) For some subjects it is possible to train a classifier and use it on-line to control a cursor achieving a hit rate of 70%. For other subjects, the on-line classification was not successful and some of the reasons were insufficient subject training and difficulty of the subject to concentrate both during the collection of the training data and the on-line classification; 2) The classifiers that worked best during the training are not necessarily the classifiers that perform best on-line although such comparison is difficult due to the different performance measures used during the training and on-line classification and also the different test sets used during the on-line classification.

4 Conclusions

In this paper we study classification of mental tasks for EEG-based BCI. We consider 2 scenarios: off-line and on-line classification of BCI data.

In the on-line scenario we used 4-class benchmark dataset from the BCI competition to evaluate a number of classification algorithms under the same conditions, i.e. using the same pre-processing and feature selection. The need for such consistent evaluation has been identified in previous research, e.g. Lotte (2007). Our evaluation included algorithms that have not been previously applied for classification of BCI data or have received very little attention such as RF, RBF, Bagging, Stacking and Boosting. The results showed that these classifiers, in addition to the popular SVM, produced best results and are good choices for classification of BCI data. We also applied a new feature selector (CFS) which exploits the high correlation of the EEG data. The results showed that it was very successful: it discarded a large number of features (87-96%) while improving or maintaining the classification accuracy for almost all classifiers. The results also showed that our classification results using CSP for signal processing, CFS for feature selection, and

SVM, AdaBoost or Bagging for classification, are comparable to the top competition results.

We designed an on-line experiment to test the performance of our BCI system in a realistic application. Data from 3 subjects was collected and used to train classifiers, which were then used to control the cursor on-line in a computer game. On the positive side, one of the subjects was able to achieve a hit rate slightly above 70% which would be very valuable for practical BCI applications and could possibly be improved with more training. However, the results also highlighted several issues. Firstly, the on-line task is more difficult than the off-line BCI classification setup used in the BCI competition data as there is visual feedback which may affect the performance. Secondly, it is difficult to compare the off-line training performance of the classifiers with their performance for on-line classification as different performance measures are used (accuracy and hit rate) and also as the subject generates different test set for the on-line testing as discussed in Section 3.3.

There are several avenues for future work. First, we plan to use more than 3 electrodes in the on-line task and apply the CSP pre-processing and CFS feature selection. Second, the BCI2000 software could be extended to report accuracy and show the confusion matrices in addition to the hit rate which will allow for consistent comparison. Third, more research is needed to choose the right mental tasks for an on-line BCI application and also to study the effect of the feedback and the potential benefits of using on-line classification algorithms.

5 Acknowledgements

We are very grateful to Jorge Villalon and Benjamin Harding for the integration of Weka with BCI2000 and the CSP implementation and to Gaetano Gargiulo for the help with the EEG data recording. This work was supported in part by the University of Sydney bridging support grant U1189

6 References

- Aha, D., and Kibler D. (1991): Instance-based learning algorithms. *Machine Learning* 6:37-66.
- Anderson C.W. and Sijercic Z. (1996): Classification of EEG signals from four subjects during five mental tasks. *Proc. International Conference on Engineering Applications of Neural Networks (EANN)*.
- Barreto, G.A., Frota R.A. and Medeiros F.N.S. (2004): On the classification of mental tasks: a performance comparison of neural and statistical approaches. *Proc. IEEE Workshop on Machine Learning for Signal Processing*.
- BCI2000 ARFilter: http://www.bci2000.org/wiki/index.php/User_Reference:ARFilter. Accessed July 2008.
- BCI Competition III: http://ida.first.fraunhofer.de/projects/bci/competition_iii/. Accessed July 2008.
- Blankertz B., Müller K.-R., Krusienski, D., Schalk G., Wolpaw J.R., Schlögl, A., Pfurtscheller, G., Millan J., Schröder, M., Birbaumer, N. (2006): The BCI2000 Competition III: validating alternative approaches to

- actual BCI problems. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*.
- Blankertz B., Curio, G. and Müller K.-R. (2001): Classifying single trial EEG: towards brain computer interfacing. In *Advances in Neural Information Processing Systems*, T.G. Diettrich, S. Becker and Z. Ghahramani, ed., **14**: 157-164.
- Boostani, R. and Moradi, M.H. (2004): A new approach in the BCI research based on fractal dimension as feature and Adaboost as classifier. *Journal of Neural Engineering* **1**:212-217.
- Breiman, L. (2001): Random forests. *Machine Learning* **45**: 5-32.
- Breiman, L. (1996): Bagging predictors. *Machine Learning* **24**(2): 123-140.
- Diettrich, T.G. (2000): An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting and randomization. *Machine Learning* **40**(2) 139-158.
- Dornhege, G., Millán, J. del R., Hinterberger, T., McFarland, D.J. and Müller K.-R. (2007): *Toward Brain-Computer Interfacing*. Cambridge, MIT Press.
- Freund Y. and Schapire R.E. (1996): Experiments with a new boosting algorithm. *Proc. International Conference on Machine Learning*, 148-156, Morgan Kaufmann, San Francisco.
- Gargiulo, G., Bifulco, P., Calvo, R.A., Cesarelli, M., Fratini, A., Jin, C. and van Schaik, A. (2008): A wearable dry-electrode-capable Bluetooth personal monitoring system. *Proc. 4th European Conference for Medical and Biomedical Engineering*, Antwerp, Belgium.
- Garrett, D., Peterson, D.A., Anderson C.W. and Thaut M.H. (2003): Comparison of linear, non-linear and feature selection methods for EEG signal classification, *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, **11**(2): 141-144.
- Jain, A. K, Duin, R.P.W. and Mao, J. (2000): Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and machine Intelligence* **22**(1): 4-37.
- Jasper, H.H. and Penfield, W. (1949): Electro-corticograms in man: effect of the voluntary movement upon the electrical activity of the precentral gyrus. *Arch. Psychiat. Z. Neurol.* **183**: 163-174.
- Hall, M. (2000): Correlation-based feature selection for discrete and numeric class machine learning. *Proc. 17th International Conference on Machine Learning (ICML)*, 359-366, Morgan Kaufmann.
- Holte, R.C. (1993): Very simple classification rules perform well on most commonly used datasets. *Machine Learning* **11**: 63-91.
- Hoya, T., Hori G., Bakardjian H., Nishimura T., Suzuki T., Miyawaki Y., Funase A. and Cao J. (2003): Classification of single trial EEG signals by combined principal + independent component analysis and probabilistic neural network approach. *Proc. 4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA)*, Japan.
- Kubat, M., Koprinska, I. and Pfurtscheller G. (1998). Learning to classify biomedical signals. In *Machine Learning and Data Mining: Methods and Applications*, R.S. Michalsi, M. Kubat and I. Bratko (ed.), Wiley.
- Le Cessie, S. and van Houwelingen, J.C. (1992): Ridge Estimators in Logistic Regression. *Applied Statistics*, **41**(1):191-201.
- Lee, F., Scherer, R., Leeb, R., Neuper, C., Bischof, H. and Pfurtscheller, G. (2005): A comparative analysis of multi-class EEG classification for brain computer interface. *Proc. 10th Computer Vision Winter Workshop (CVWW)*, Technical University of Graz, Austria.
- Lotte, F., Congedo M., Lecuyer A., Lamarche F. and Arnaldi B. (2007): a review of classification algorithms for EEG-based brain-computer interfaces. *Journal of Neural Engineering* **4**: R1-R13.
- McFarland D.J., McCane I. M., David S.V., Wolpaw J. R. (1997): Spatial filter selection for EEG-based communication. *Electroencephalography and Clinical Neurophysiology* **103**(3) 386-394.
- Moody J. and Darken C. (1989): Fast training in networks of locally-tuned processing units. *Neural Computation* **1**: 284-294.
- Müller, K-R., Anderson C.W and Birch G.E (2003): Linear and nonlinear methods for brain-computer interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **11**(2): 165-169.
- Müller, K-R., Krauledar, M., Dornhege, G., Curio, G. and Blankertz, B. (2004): Machine learning techniques for brain computer interfaces. *Biomedical Technology* **49**: 11-22.
- Opitz, D. and Maclin, R. (1999): Popular ensemble methods: an empirical study. *Journal of Artificial Intelligence Research* **11**:169-198.
- Platt, J. (1998): Fast Training of Support Vector Machines using Sequential Minimal Optimization. In *Advances in Kernel Methods - Support Vector Learning*, B. Schoelkopf, C. Burges, and A. Smola, eds., MIT Press.
- Quinlan, J.R. (1986). Induction of decision trees. *Machine Learning* **1**(1): 81-106.
- Müller-Gerking, J., Pfurtscheller, G. and Flyvbjerg H. (1999): Designing optimal spatial filters for single-trial EEG classification in a movement task. *Clinical Neurophysiology*, **110**(5): 787-798.
- Schlögl, A., Lee, F., Bischof, H. and Pfurtscheller, G. (2005): Characterization of four-class motor imagery EEG data for the BCI competition 2005. *Journal of Neural Engineering* **2**: L14-L22.
- Schalk, G., McFarland D. J., et al. (2004): BCI2000: A general-purpose Brain-Computer Interface (BCI) System. *IEEE Transactions on Biomedical Engineering* **51**(6): 1034-1043.
- Thulasidas, M., Guan C. and Wu, J. (2006): Robust classification of EEG signal for brain-computer interface. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **14**:24-9.

- Witten, I. H and Frank, E. (2005): *Data mining: practical machine learning tools and techniques*. Second edition, San Francisco, Morgan Kaufmann.
- Wolpert, D.H. (1992): Stacked generalization. *Neural Networks* **5**: 231-259.
- Wolpaw, J.R., Birbaumer, N, Heetderks, W.J., McFarland, D.J, Peckham, P.H., Schalk, G., Donchin, E., Quatrano, L.A, Robinson, C.J., Vaughan, T.M. (2000): Brain-computer interface technology: a review of the first international meeting. *IEEE Transactions on Rehabilitation Engineering* **8**:164-173.

Kernel-based Visualisation of Genes with the Gene Ontology

Hamid Ghous¹ Paul J. Kennedy¹ Daniel R. Catchpoole²
 Simeon J. Simoff³

¹ Faculty of Engineering and Information Technology,
 University of Technology, Sydney,
 PO Box 123, Broadway NSW 2007, AUSTRALIA,
 Email: Hamid.Ghous@student.uts.edu.au, paulk@it.uts.edu.au

² Tumour Bank, The Children's Hospital at Westmead,
 Locked Bag 4001, Westmead NSW 2145, AUSTRALIA,
 Email: DanielC@chw.edu.au

³ School of Computing and Mathematics, University of Western Sydney,
 NSW, AUSTRALIA,
 Email: s.simoff@uws.edu.au

Abstract

With the development of microarray-based high-throughput technologies for examining genetic and biological information en masse, biologists are now faced with making sense of large lists of genes identified from their biological experiments. There is a vital need for "system biology" approaches which can allow biologists to see new or unanticipated potential relationships which will lead to new hypotheses and eventual new knowledge. Finding and understanding relationships in this data is a problem well suited to visualisation. We augment genes with their associated terms from the Gene Ontology and visualise them using kernel Principal Component Analysis with both specialised linear and Gaussian kernels. Our results show that this method can correctly visualise genes by their functional relationships and we describe the difference between using the linear and Gaussian kernels on the problem.

Keywords: kernel-based visualization, Gene Ontology, biomedical datasets.

1 Introduction

It is well recognised that improvements in health are universally driven by gains in understanding of the biology behind human disease. With the completion of the Human Genome Project and the development of microarray-based high-throughput technologies for examining genetic and biological information en masse, biologists are now seeking to assess systems of biological information rather than single genes. Consequently they have to deal with large amounts of information such as lists of hundreds or even thousands of genes. There is a vital need for tools which not only relate this mass of data to current knowledge through bioinformatic approaches but can assess this data to allow us to see new or unanticipated potential relationships within the system which will lead to new hypotheses and eventual new knowledge. In other words, "systems biology" approaches are required. Finding relationships in this morass of data is a problem that is well suited to unsupervised

data mining methods such as visualisation. Unsupervised learning methods deal with similarity measures between items of interest, in this case genes. To find similarities, lists of genes must be augmented with additional information. In this paper, we will augment genes with their associated terms from the Gene Ontology (GO) (Ashburner et al. 2000), which is a massive Internet database curated by biologists that defines over 25000 terms in a controlled vocabulary describing genes and gene products. These gene terms fit into three disjoint hierarchies or subontologies: cellular components, molecular functions and biological processes. Terms in the cellular component hierarchy are associated with the physical structure of gene products and generally relate to where the gene product is found in the cell. Terms in the molecular function hierarchy describe the biochemical activity of gene products. Finally, terms in the biological process hierarchy relate to the biological objective to which genes or gene products contribute. The Gene Ontology takes the form of a large database and allows GO terms to be found for specific genes and gene products. Other information, including cross references to other bio-databases, may also be found for genes. We calculate the similarity between genes using the similarity between their component terms.

1.1 Related Work

Several other researchers have explored the problem of applying unsupervised learning methods to lists of genes. Work generally falls into three main areas: describing groups of genes in terms of their annotations; measures for calculating similarity between genes using GO annotations; and clustering and visualisation of genes using GO annotations. The last two of these are similar because clustering and visualisation methods require similarities to be calculated for the genes.

Methods to describe groups of genes from GO annotations include methods that view the Gene Ontology as a simple collection of terms without exploiting too much the structural interrelationships (eg. Gattviks et al. (2003) or Shah & Fedoroff (2004)). Others including Beißbarth & Speed (2004) and Zeeberg et al. (2003) use statistical methods to analyze the GO categories. Cheng et al. (2004) use the Bootstrap Test on GO cliques to determine the statistical categorizing of GO categories. Lee et al. (2004) introduced an algorithm to find the significant biological features of a gene cluster or group of interest through the tree structure of the Gene Ontology. They applied a transformation of the GO directed acyclic graph structure

Copyright ©2008, Australian Computer Society, Inc. This paper appeared at the Seventh Australasian Data Mining Conference (AusDM 2008), Glenelg, Australia. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 87, John F. Roddick, Jiuyong Li, Peter Christen and Paul Kennedy, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

with a distance function. Their graph theoretic algorithm extracts common or representative GO terms for a gene cluster by taking the multi-functionality of genes into account. Popescu et al. (2004) construct a functional summary of clusters of genes using GO terms. They build a “most representative term” (MRT) for each cluster by making a hierarchical clustering of the genes and then applying fuzzy methods to find the GO terms of highest frequency. Liu et al. (2005) describe a tool called DYnGO, which allows users to conduct batch retrieval of GO annotations for a list of genes and semantic retrieval of genes and gene products sharing similar GO annotations. Results are shown in a tree format sorted by GO term.

Similarity measures usually use the hierarchical structure of the Gene Ontology. Mathur & Dinakarpandian (2007) describe an approach to computing gene product similarity by considering both the hierarchical nature of GO and the co-occurrence of GO terms in annotations. Their approach considers numbers of annotations and differences in the frequency of usage of GO terms with a set-based similarity function. Sanfilippo et al. (2007) categorise GO based similarity approaches into two main categories: similarities based on hierarchical relationships within a GO subontology (of which there are three subontologies) and similarities based on associative relationships of genes across the three subontologies. This latter approach predicts annotations in a subontology for a specific gene based on annotations for similar genes. They propose a method called cross-ontological analytics that merges these approaches. They also integrate textual data from biomedical literature with GO knowledge.

Clustering and visualisation approaches go further and apply gene similarity measures to understanding the natural structure of groups of genes and gene products. Lee et al. (2005) propose an ontology-based clustering algorithm (CLUGO) that identifies clusters of significant GO terms within a distribution of terms (eg. that arise from some previous clustering exercise). Kennedy & Simoff (2003) describe a technique for clustering genes based on GO terms using the MBSAS clustering algorithm. However, the method is sensitive to gene order and does not scale to large numbers of genes.

Speer et al. (2005) and Fröhlich et al. (2007) describe a kernel-based approach to clustering genes using Gene Ontology annotations. They define a kernel based on information-theoretic measures to calculate similarities between genes (based on the maximum similarity between terms). They state that their information-theoretic approach better models the variable branching and density of the GO graph and that it should perform better than link distance based measures like the one we use. They apply a dual k-means clustering to groups of genes and provide an R tool.

Like Fröhlich et al. (2007) we devise a kernel function. However, our measure is link distance based rather than information-theoretic. Also, Fröhlich et al. (2007) focusses on clustering, specifically a dual k-means clustering approach. Our motivation, on the other hand, is to visualise the genes with the longer term goal of explaining why a particular set group the way they do.

The rest of this paper is organised as follows. Section 2 elaborates on our approach to visualising a list of genes. Section 3 details the dataset used in this paper to validate our approach: a dataset derived from the KEGG (Kanehisa et al. 2008) database. Also, in section 3 we describe a series of experiments applying variants of our approach together with results. In section 4 we list potentially fruitful areas for future research. Finally, section 5 concludes the paper.

2 Method

This section describes our approach to visualisation of lists of genes. First we describe in more detail the Gene Ontology and the type of data we extract from it. Then we describe the unsupervised visualisation approach we apply, namely kernel Principal Component Analysis (kPCA).

2.1 Gene Ontology

The Gene Ontology provides a controlled vocabulary to describe genes and gene product attributes in many organisms. It is a collaborative effort beginning in 1998 and spans many organisms including but not limited to *Drosophila*, *Saccharomyces*, mouse and human.

The building blocks of the Gene Ontology are the terms. Each entry in GO has (i) a unique alphanumeric identifier (GO:#####); (ii) term name, e.g. cell, fibroblast growth factor receptor binding or signal transduction; (iii) synonyms (if applicable); and (iv) a definition. Each term is also assigned to one of the three hierarchies, which are structured as directed acyclic graphs. Most terms have a textual definition, with references stating the source of the definition. If any clarification of the definition or remarks about term usage is required, these are held in a separate comments field.

Each gene has one or more terms related to it and a term may have multiple parents on the tree. The terms provide us with a description of the functionality of a gene.

Table 1 shows three example genes with their related terms. Following each term name is the Gene Ontology accession number for the term. One of the challenges with using terms from the Gene Ontology is that terms give different amounts of information. For example, the gene *Aldh1a7* in Table 1 contains some very specific terms such as “retinal metabolic process” or “aldehyde dehydrogenase (NAD) activity” which give specific and useful information along with other terms such as “cytoplasm” or “metabolic process” which are more general (high in the hierarchy) and shared by many other genes. These latter terms do not confer much useful information. Also, some genes have been investigated thoroughly and have many annotations (such as *Aldh1a7*) whilst others are not well annotated (such as *Srpx2*). In short, the information associated with genes in the Gene Ontology is of mixed quality. This presents challenges for its use in augmenting lists of genes.

Table 1: Example of three genes from the Gene Ontology.

Gene Name	Term Name and Accession
<i>Aldh1a7</i>	cytoplasm (GO:0005737) oxidoreductase activity (GO:0016491) aldehyde dehydrogenase (NAD) activity (GO:0004029) metabolic process (GO:0008152) retinal metabolic process (GO:0042574)
<i>Srpx2</i>	electron transport (GO:0006118) extracellular region (GO:0005576)
<i>Tspan7</i>	biological process (GO:0008150) molecular function (GO:0003674) integral to membrane (GO:0016021) membrane attack complex (GO:0005579)

As illustrated in Fig. 1, GO terms are related in

two main ways: “is-a” and “part-of”. The “is-a” relationship is the main relationship seen in the Gene Ontology and represents a simple class-subclass relationship. For example, the figure shows that the term “extracellular space” is an “extracellular region part” and that an “extracellular region part” is a “cellular component”. Cellular component is the root of the hierarchy. Less commonly seen is the “part-of” relationship which signals containment. If C is “part-of” D it means that whenever C is present, it is always a part of D , but that C does not always have to be present. For example, in the figure “extracellular region part” is part of “extracellular region”.

The Gene Ontology database allows SQL queries of the terms associated with genes, the relationships between terms (parent and child) as well as finding the distance between terms in number of “hops”. There are also many web-based tools available to query the databases.

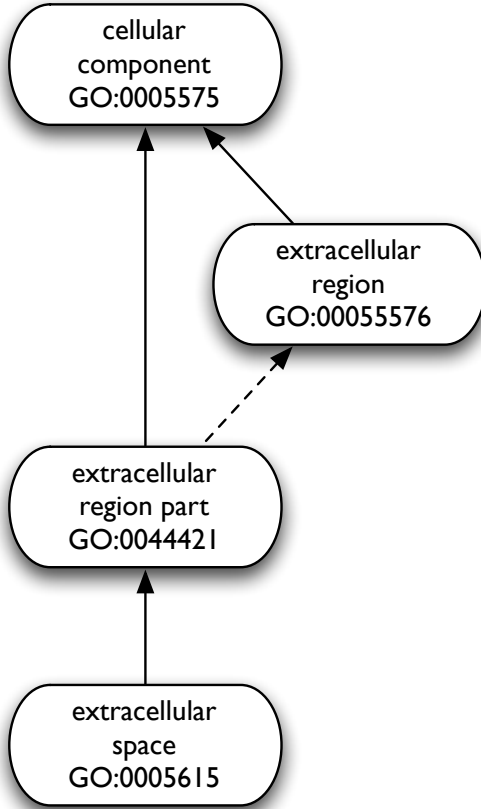


Figure 1: Example of small part of the hierarchical structure of GO terms. Solid lines represent “is-a” relationships and dashed line represents a “part-of” relationship.

2.2 Kernel Principal Component Analysis

The visualisation approach we use in this paper is a kernel-based extension to Principal Component Analysis (PCA) (Jolliffe 2004, Haykin 1999). Principal Component Analysis is a well known data transformation method that rotates a dataset into a different coordinate system. The coordinates of the transformed dataset (called principal components) are orthogonal linear combinations of the original coordinates. The principal components are ordered in descending order by the amount of variance they explain in the data. Generally, most of the variance in the dataset can be explained by many fewer coordinates than in the original dataset (e.g. less than ten) with the last principal

coordinates often associated with noise components of the original data. Consequently, PCA is often used for compression of data or feature selection. Principal Component Analysis allows visualization of datasets by plotting the first two or three principal components of the data. However, due to the fact that the principal components are linear combinations of the original dataset, PCA has the limitation that it can model only linear relationships in the data.

When applying PCA the dataset can be viewed as a matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ where n is the number of data items each containing d attributes and the d -dimensional row vector x_i represents each data item. The principal components of the dataset are the eigenvectors of the covariance or correlation matrix of \mathbf{X} ordered by decreasing value of the associated eigenvalue. So the first principal component is the eigenvector of the covariance/correlation matrix with the largest eigenvalue. The data is transformed into the principal component space by projecting each data item x_i along the principal components.

Several approaches have been devised to extend PCA to recognise nonlinear relationships among data attributes. One approach is kernel PCA (kPCA) (Müller et al. 2001, Haykin 1999, Shawe-Taylor & Cristianini 2004) which transforms the dataset \mathbf{X} into a feature space using a (nonlinear) kernel function κ before the PCA is done. Kernel PCA returns the principal components of the data items in the feature space. The input to kPCA is a Gram kernel matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$ which is a representation of the original dataset transformed with the kernel function. Each element k_{ij} of the kernel matrix can be viewed as a similarity between the data items x_i and x_j and is defined as

$$k_{ij} = \kappa(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle \quad (1)$$

where x_i and x_j are the data items, $\phi(x_i)$ is the transformation of x_i into the “feature” space and $\langle \cdot, \cdot \rangle$ is the dot product operator. Generally it is not necessary to compute $\phi(x_i)$ explicitly. Instead, \mathbf{K} is computed directly from the dataset. This is called the “kernel trick” and it means that the feature space can be very large without making generation of \mathbf{K} inefficient. It also means that non-vectorial data types can be handled using special kernels such as string kernels (e.g. (Leslie et al. 2004)). In kPCA the principal components are the eigenvectors of the kernel matrix.

Two common kernel functions are the *linear kernel* and the *Gaussian kernel*. The linear kernel is defined as

$$\kappa(x_i, x_j) = \langle x_i, x_j \rangle \quad (2)$$

and is simply the dot product of the two data items. The whole linear kernel matrix \mathbf{K} can be easily computed as $\mathbf{K} = \mathbf{X}\mathbf{X}'$ where \mathbf{X}' denotes the transpose of \mathbf{X} . Kernel PCA using the linear kernel is analogous to the standard (linear) PCA.

The Gaussian kernel explicitly considers the distance between data items and is defined as

$$\kappa(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \quad (3)$$

where σ is a control parameter governing the “width” of the Gaussian curve. The Gaussian kernel can also be viewed as a series of transformations applied to the linear kernel. Specifically, $\kappa(x_i, x_j) = \exp(\langle x_i, x_j \rangle / \sigma^2)$ and then normalised (Shawe-Taylor & Cristianini 2004).

In this study, we employ kPCA with both linear and Gaussian kernel matrices. However, as described

below, we use a slightly different linear kernel matrix to be able to use the Gene Ontology data. Also, before applying kPCA, we centre and normalise the data through the kernel matrix.

2.3 The Kernel Function for the Gene Ontology Data

Given a set of genes G , we query the Gene Ontology to find all GO terms directly associated with the genes. Define T as the set of GO terms directly associated with any of the genes in G .

From a list of genes we create a matrix $\mathbf{X} \in \mathbb{R}^{n \times t}$ where n is the number of genes (ie. $|G|$) and t is the number of GO terms (ie. $|T|$). Each element x_{ij} of \mathbf{X} has the value 1 if the gene i is directly associated with term j and 0 otherwise. This kind of scheme is similar to approaches used in computational linguistics where genes are replaced by documents and terms are replaced by words.

The linear kernel matrix would normally be defined as $\mathbf{X}\mathbf{X}'$ except that this ignores relationships between the GO terms. Therefore, we create an additional matrix $\mathbf{P} \in \mathbb{R}^{t \times t}$ called the proximity matrix with each element p_{ij} representing the proximity (or similarity) between GO term i and j . Terms with a close relationship have values close to 1, with the diagonal elements $p_{ii} = 1$. The proximity between GO terms is based on the number of links (or distance) between them and is defined as

$$p_{ij} = \frac{1}{d_{ij} + 1} \quad (4)$$

where d_{ij} is the minimum distance between terms i and j over the hierarchy. The distance can be extracted from the Gene Ontology. Clearly \mathbf{P} is symmetric.

The kernel matrix for the gene data, then, is defined as

$$\mathbf{K} = \mathbf{X}\mathbf{P}\mathbf{P}'\mathbf{X}' \quad (5)$$

The proximity matrix weights up GO terms in \mathbf{X} that are close to one another. Proximity matrices have been used before for text kernels (eg. (Shawe-Taylor & Cristianini 2004)).

The Gaussian extension to this kernel is straightforward as alluded to in the last section. Working from equation (5) we simply scale by σ^2 , take the exponent and normalise.

Consequently, in this paper we apply a linear kernel for comparing genes based on their GO terms using equation (5) and a Gaussian kernel based on the linear kernel.

3 Experiments

In this section we describe several experiments validating our method. First we describe the KEGG data set we used. Following this we give results for kernel PCA visualisation of the dataset using linear and Gaussian kernels.

3.1 Dataset

In this study we visualise a subset of genes from the Internet KEGG database. The KEGG dataset contains a list of genes classified into different classes of behaviour. The rationale behind using this dataset is to validate our approach with genes of known functional similarity.

The Kyoto Encyclopedia of Genes and Genomes (KEGG) (Kanehisa et al. 2008) is a biological resource which aims to link genomes to the biological

systems they govern. The resource takes the form of a series of interconnected databases of biological systems that interrelate (i) genes and proteins, (ii) chemical building blocks, (iii) molecular interaction pathway diagrams and (iv) hierarchies and relationships of biological objects. We are interested in the last of these databases (KEGG BRITE) which links genes into a functional hierarchy called the KEGG Orthology (KO). Importantly, this hierarchy is different to that of the Gene Ontology and has been constructed independently. This allows us to validate our visualisation by extracting genes that are similar according to their KO terms and then to visualise them using their GO terms. Consequently, our KEGG dataset contains a subset of genes from five classes of KO: ribosome (ko03010), RNA polymerase (ko03020), transcription (ko01210), pentose phosphate pathway (ko00030) and pentose and glucuronate interconversions (ko00040). Table 2 shows the interrelationships between the classes in terms of the parent KO terms to the selected classes. From these interrelationships we expect to see that classes 1, 2 and 3 are similar (with classes 2 and 3 more similar to one another than to class 1). Classes 4 and 5 should also be similar to one another but different to the other three classes.

A subset of genes was chosen from the lists given by KEGG. From the list of genes on KEGG, we chose those that were also accessible in the Gene Ontology database. The number of genes chosen for each class is given in Table 2.

3.2 Visualising the KEGG dataset

Initially we visualised the KEGG dataset with a linear kernel. This is equivalent to applying linear Principal Component Analysis to the dataset. For the genes listed in the KEGG dataset, we extracted the GO terms associated with the genes and generated \mathbf{X} and \mathbf{P} matrices as described in Section 2.3. Next, using equation (5) we generated the basic kernel matrix \mathbf{K} . Finally, we applied kernel PCA to this kernel as described in section 2.2.

Figure 2 shows a plot of the eigenvalues (λ) found for the principal components. These values are associated with the variance of the data explained by the corresponding principal component. As can be seen, the first principal component is very large compared to the rest. This is often a sign of one attribute dominating the principal component (or use of a covariance matrix rather than a correlation matrix). As described in section 2.2 above, we scale the kernel matrix which is equivalent to using a correlation matrix. Also, since we “fold” the original data attributes into kernel values we can no longer easily investigate the original principal component vectors to see whether one term dominates.

However, investigation of the terms associated with the genes suggests that the first principal component is a “size” component as described in Jolliffe (2004). “Size” components are found in (Jolliffe 2004) by checking the values of the principal component vector. When all (or most) values in the vector are strongly positive for each attribute then Jolliffe suggests that the principal component measures the general size of the data items. We cannot check the values of the principal component vector for each of the attributes because we are using a kernel-based approach rather than standard PCA and the original data attributes (ie. the terms) are hidden in the kernel value.

In Fig. 3 we plot the genes in the KEGG dataset according to principal component axes PC1 and PC2. The graph shows two groups of genes separated by principal component 1. The separation along PC1

Table 2: Description of the 5 classes of genes in our KEGG dataset. Column 1 contains the class identifier and the symbol used in our graphs. Column 2 gives the list of KO terms leading to the class and column 3 lists the number of genes in the class.

Class	KO structure	Count
1 +	genetic information processing : translation : ribosome	20
2 ×	genetic information processing : transcription : RNA polymerase	19
3 ○	genetic information processing : transcription	11
4 □	metabolism : carbohydrate metabolism : pentose phosphate pathway	11
5 ●	metabolism : carbohydrate metabolism : pentose and glucuronate interconversions	8

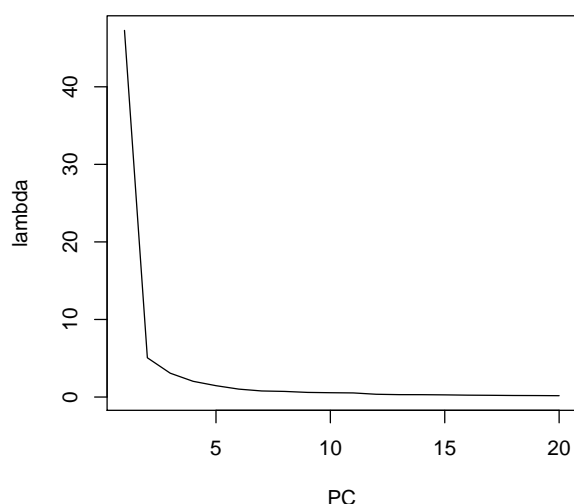


Figure 2: Sorted eigenvalues λ associated with the first 20 principal components of the visualisation of the KEGG dataset using the linear kernel. The first eigenvalue is very high compared to the rest.

does not reflect the KEGG class of the gene, although PC2 does discriminate by class to some extent. Investigation of the genes in each of the two clusters along PC1 show that genes on the left hand side have fewer terms associated with them compared to genes on the right hand side and that genes in the middle have a count of associated terms mid way between the extremes. For example, the four genes on the extreme left hand side of Fig. 3 are NUSG (with one associated GO term) and RPMF, RPSF and RPLD (each with 2 GO terms). The four genes on the extreme right hand side are RHO (27 terms), Elp3 (20 terms), Eda (18 terms) and Clpx (17 terms). This suggests that our interpretation of PC1 as a “size” component is the correct one for this dataset.

Consequently, in Fig. 4 we plotted the genes according to the next two principal components: PC2 and PC3. This figure shows that PCs 2 and 3 result in a visualisation that reflects the classes of genes. Genes that are similar to one another (ie. fall within a class) group together and those that are different are generally separated. Genes in classes 4 (□, pentose phosphate pathway) and 5 (●, pentose and glucuronate interconversions) group very closely together as expected. These are generally far apart from the genes in the other classes except for some overlap at the origin. Genes in class 1 (+, ribosome) cluster tightly and there is a closer relationship between genes in class 2 (×, RNA polymerase) and class 3 (○, transcription) than the translation related genes of class 1. Principal component 2 contrasts the carbohydrate metabolism related genes with the genetic information processing

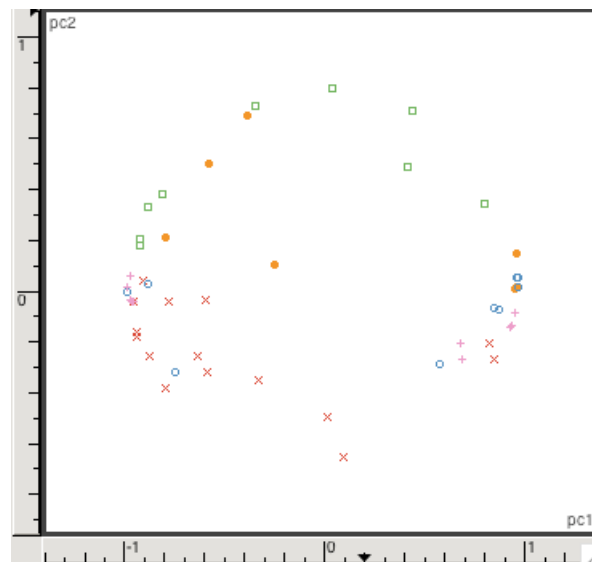


Figure 3: Plot of genes from KEGG dataset according to PC1 and PC2 using the linear kernel. Key: + = ribosome, × = RNA polymerase, ○ = transcription, □ = pentose phosphate pathway, ● = pentose and glucuronate interconversions.

related genes. This accords well with what we would *a priori* expect to be the main variance in the genes. Principal component 3 then contrasts the different kinds of genetic processing related genes.

Next, we applied the Gaussian kernel to the linear kernel \mathbf{K} generated above as described in section 2.3. We explored various settings of the σ parameter and empirically found that when $\sigma = 3$ it starts showing different clusters but $\sigma = 10$ gave reasonable visualisations where the genes did not end up on top of one another or spread out like the linear kernel. Plotting the eigenvalues (λ) does not make sense in this case because the principal components relate to the infinite dimensional feature space induced by the Gaussian kernel. Figure 5 plots the genes according to principal components 1 and 2. The genes at the ends of the tails in Fig. 5 are the same as those in Fig. 3 which again suggests that the first principal component contrasts the number of GO terms associated with the genes. Specifically, the gene marked ○ at the end of the left hand tail of Fig. 5 is RHO (27 terms). The next is ELP3 (20 terms) followed by EDA (18 terms) and Clpx (17 terms). These are the same as in the linear diagram and are ordered by the number of terms. At the other end are NUSG (1 term), RPMF, RPSF and RPLD (2 terms).

Figure 6 graphs the genes by the second and third principal components. As with the linear case, the genes cluster mostly according to functionality and KEGG class. Genes that were at the origin of the linear graph (Fig. 4), however, have moved to the left hand “spike” of the Gaussian graph (Fig. 6). The grouping of RNA polymerase genes (×) at the top left

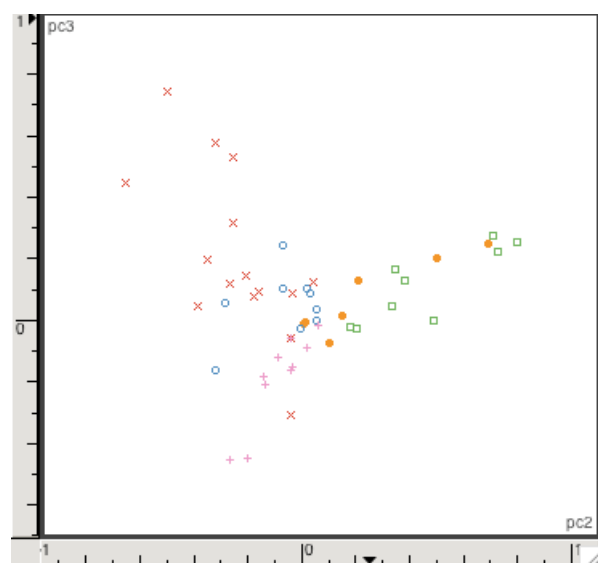


Figure 4: Plot of genes from KEGG dataset according to PC2 and PC3 using the linear kernel. Key: + = ribosome, \times = RNA polymerase, \circ = transcription, \square = pentose phosphate pathway, \bullet = pentose and glucuronate interconversions.

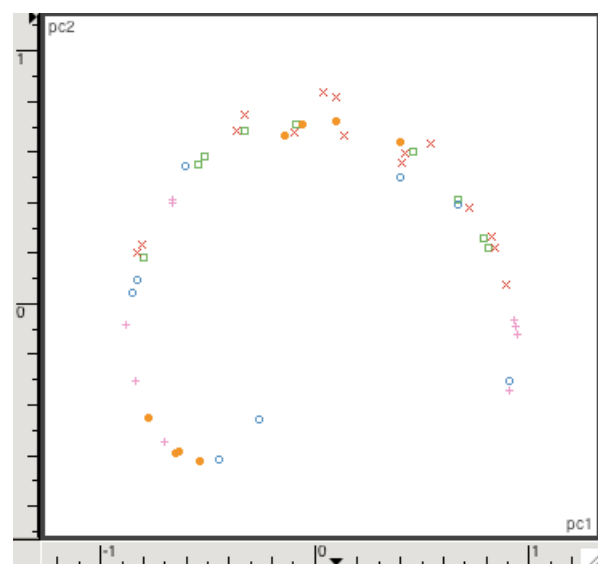


Figure 5: Plot of genes from KEGG dataset according to PC1 and PC2 using the Gaussian kernel with $\sigma = 10$. Key: + = ribosome, \times = RNA polymerase, \circ = transcription, \square = pentose phosphate pathway, \bullet = pentose and glucuronate interconversions.

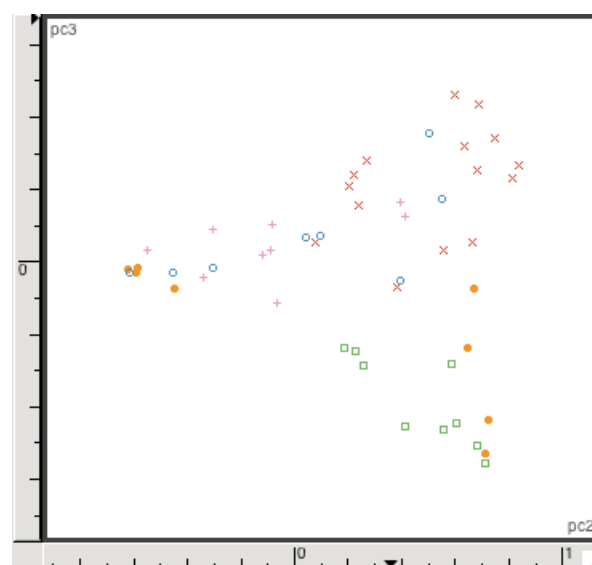


Figure 6: Plot of genes from KEGG dataset according to PC2 and PC3 using the Gaussian kernel with $\sigma = 10$. Key: + = ribosome, \times = RNA polymerase, \circ = transcription, \square = pentose phosphate pathway, \bullet = pentose and glucuronate interconversions.

of Fig. 4 still cluster separately (top right of Fig. 6) and the small group of pentose phosphate pathway (\square) and pentose and glucuronate interconversions (\bullet) genes at the right hand side of Fig. 4 have grouped more closely in the bottom right of Fig. 6.

We also examined visualisations of the data with different σ values. Specifically, we examined $\sigma = 0.1, 1, 1.5, 2, 2.5, 3, 5, 7, 25, 50, 75, 100, 500, 1000$. Using the value of 0.1 condensed the genes on top of one another. At value 2 genes starts to open up and at $\sigma = 3$ the genes start to make shape. Values of 50, 75, 100, 500 and 1000 look the same as the linear kernel, as expected. Figures 7 and 8 show visualisations using $\sigma = 1$ of the first two principal components and the second two components respectively. Many of the genes sit on top of one another so jitter (small random adjustments) has been added to the genes on these figures. With $\sigma = 1$, the first principal component no longer seems to act as a “size” component. However, the first two principal components contrast most of the ribosome (+) and transcription (\circ) genes from the others, as does the third principal component. The fourth principal component expresses the variance associated to the RNA polymerase (\times) genes. Although not shown, it is not until later principal components that the carbohydrate metabolism genes get distinguished from the others. Since there are many more genetic information processing genes in the dataset compared to the carbohydrate metabolism genes (see Table 2) it is expected that the earlier principal components are concerned with this variation. Also, the narrower focus of the σ gives a finer grained distinction between genes. This suggests that use of the Gaussian kernel rather than the linear kernel (ie. ordinary PCA) is important for distinguishing between different genes. The statistical properties of the Gaussian kernel are useful to the visualisation. Choice of the σ parameter is anticipated to be problematic for datasets where the relationship between genes is unknown and tuning of this parameter will be the subject of a future investigation.

Finally, we also explored use of a different distance function between terms to generate the proximity matrix \mathbf{P} . Rather than simply counting the links between the terms using equation (4) we instead

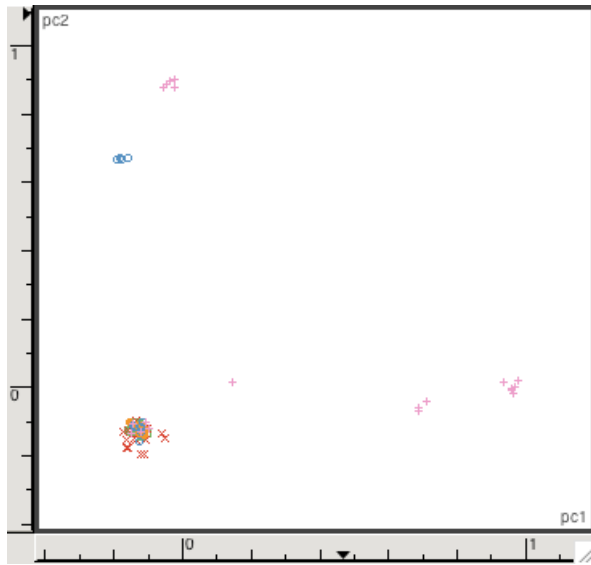


Figure 7: Plot of genes from KEGG dataset according to PC1 and PC2 using the Gaussian kernel with $\sigma = 1$ with a small amount of jitter applied to the values. Key: + = ribosome, \times = RNA polymerase, \circ = transcription, \square = pentose phosphate pathway, \bullet = pentose and glucuronate interconversions.

weighted down long paths. The motivation behind doing this is to emphasise close relationships between genes rather than distant relationships (where terms are related only through the very high level and overly general GO terms). The discounting distance function is defined as

$$d'_{ij} = \sum_{k=0}^{d_{ij}-1} c^k \quad (6)$$

where d_{ij} is the distance between terms reported by the Gene Ontology and $c \in [0, 1]$ is a discounting constant set to 0.9 in our experiments. However, the visualisations were very similar for both the linear and Gaussian kernels so we do not show them here. A more appropriate way to discount the distance would be to weight down the distance to the closest parent of the terms i and j following equation (6). However, the distance to the closest common parent term was not easily accessible from the Gene Ontology database, so we did not pursue the approach.

4 Future Work

There are several areas that we think warrant further investigation. The most important involves investigating how to decide whether one visualisation is “better” than another. This is useful because it allows tuning of parameters and should be used to decide on whether one algorithm is better than another. Along these lines we plan to investigate the “trustworthiness” metric of Venna & Kaski (2007) which uses notions based on precision and recall to compare visualisations of microarray data. Once a “ruler” for comparing visualisations is established we can turn to tuning of the σ parameter for the Gaussian kernel. We plan also to compare our similarity measure with others, most notably the information-theoretic one of Speer et al. (2005) to see which gives better visualisations. We plan also to examine other kernel-based visualisation methods and variants of the discounting distance function given in equation (6). Finally, but not least importantly, we will visualise datasets

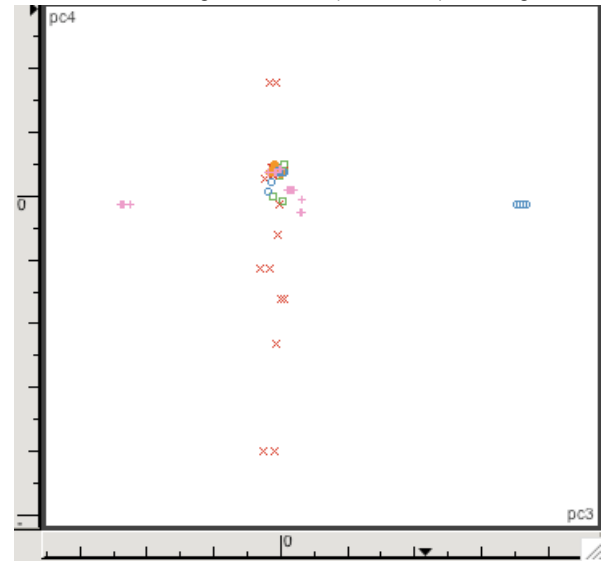


Figure 8: Plot of genes from KEGG dataset according to PC3 and PC4 using the Gaussian kernel with $\sigma = 1$ with a small amount of jitter applied to the values. Key: + = ribosome, \times = RNA polymerase, \circ = transcription, \square = pentose phosphate pathway, \bullet = pentose and glucuronate interconversions.

from experiments by biologists to gain a better understanding of their needs and the questions they want answered.

5 Conclusion

This paper describes an approach to visualising genes using kernel Principal Component Analysis. We define a specialised linear kernel based on computational linguistics and a Gaussian variant that was able to find similarities between genes using terms from the Gene Ontology. Functional relationships between genes chosen from classes within KEGG were correctly visualised with the technique.

References

- Ashburner, M., Ball, C., Blake, J., Botstein, D., Butler, H., Cherry, J., Davis, A., Dolinski, K., Dwight, S., Eppig, J. et al. (2000), ‘Gene ontology: tool for the unification of biology. The Gene Ontology Consortium.’, *Nat Genet* **25**(1), 25–9.
- Beißbarth, R. & Speed, T. (2004), ‘Gostat: finding statistically over expressed Gene Ontologies within groups of genes’, *Bioinformatics* **20**(9), 1464–1465.
- Cheng, J., Cline, M., Martin, J., Finkelstein, D., Awad, T., Kulp, D. & Siani-Rose, M. A. (2004), ‘A knowledge-based clustering algorithm driven by Gene Ontology’, *Journal of Biopharmaceutical Statistics* **13**(3), 687–700.
- Fröhlich, H., Speer, N., Poustka, A. & Beißbarth, T. (2007), ‘GOSim—An R-package for computation of information theoretic GO similarities between terms and gene products’, *BMC Bioinformatics* **8**, 166.
- Gat-Viks, I., Sharan, R. & Shamir, R. (2003), ‘Scoring clustering solutions by their biological relevance’, *Bioinformatics* **19**(18), 2381–2389.
- Haykin, S. (1999), *Neural networks: a comprehensive foundation*, 2nd edn, Prentice-Hall.

- Jolliffe, I. T. (2004), *Principal Component Analysis*, Springer Series in Statistics, second edn, Springer, New York.
- Kanehisa, M., Araki, M., Goto, S., Hattori, M., Hirakawa, M., Itoh, M., Katayama, T., Kawashima, S., Okuda, S., Tokimatsu, T. & Yamanishi, Y. (2008), 'KEGG for linking genomes to life and the environment', *Nucleic Acids Research* **36**, 480–484.
- Kennedy, P. J. & Simoff, S. J. (2003), CONGO: clustering on the Gene Ontology, in 'Proceedings Australasian Data Mining Workshop', pp. 181–198.
- Lee, I.-Y., Ho, J.-M. & Chen, M.-S. (2005), CLUGO: a clustering algorithm for automated functional annotations based on Gene Ontology, in 'Proceedings of Fifth IEEE International Conference on Data Mining', IEEE.
- Lee, S., Hur, J. & Kim, Y. (2004), 'A graph-theoretic modeling on GO space for biological interpretation of gene clusters', *Bioinformatics* **20**(3), 381–388.
- Leslie, C., Kuang, R. & Eskin, E. (2004), Inexact matching string kernels for protein classification, in B. Schölkopf, K. Tsuda & J.-P. Vert, eds, 'Kernel methods in computational biology', MIT Press, pp. 95–112.
- Liu, H., Hu, Z.-Z. & Wu, C. H. (2005), 'DynGO: a tool for visualizing and mining of Gene Ontology and its associations', *BMC Bioinformatics* **6**(201).
- Mathur, S. & Dinakarpanthian, D. (2007), 'A New Metric to Measure Gene Product Similarity', *Bioinformatics and Biomedicine, 2007. BIBM 2007. IEEE International Conference on* pp. 333–338.
- Müller, K., Mika, S., Rätsch, G., Tsuda, K. & Schölkopf, B. (2001), 'An introduction to kernel-based learning algorithms', *IEEE Transactions on Neural Networks* **12**, 181–201.
- Popescu, M., Keller, J., Mitchell, J. & Bezdek, J. (2004), Functional summarization of gene product clusters using Gene Ontology similarity measures, in 'Proceedings of IEEE Intelligent Sensors, Sensor Networks and Information Processing Conference', IEEE, pp. 553–558.
- Sanfilippo, A., Posse, C., Gopalan, B., Riensche, R., Beagley, N., Baddeley, B., Tratz, S. & Gregory, M. (2007), 'Combining Hierarchical and Associative Gene Ontology Relations With Textual Evidence in Estimating Gene and Gene Product Similarity', *Nanobioscience, IEEE Transactions on* **6**(1), 51–59.
- Shah, N. H. & Fedoroff, N. V. (2004), 'CLENCH: a program for calculating Cluster ENriCHment using the Gene Ontology', *Bioinformatics* **20**(7), 1196–1197.
- Shawe-Taylor, J. & Cristianini, N. (2004), *Kernel Methods for Pattern Analysis*, Cambridge University Press, Cambridge.
- Speer, N., Fröhlich, H., Spieth, C. & Zell, A. (2005), Functional grouping of genes using spectral clustering and gene ontology, in 'Proceedings of the IEEE International Joint Conference on Neural Networks', pp. 298–303.
- Venna, J. & Kaski, S. (2007), 'Comparison of visualization methods for an atlas of gene expression data sets', *Information Visualization* **6**, 139–154.
- Zeeberg, B., Feng, W., Wang, G., Wang, M., Fojo, A., Sunshine, M., Narasimhan, S., Kane, D., Reinhold, W., Lababidi, S. et al. (2003), 'GoMiner: a resource for biological interpretation of genomic and proteomic data', *Genome Biol* **4**(4), R28.

wFDT - Weighted Fuzzy Decision Trees for Prognosis of Breast Cancer Survivability

Umer Khan¹, Hyunjung Shin², Jong Pill Choi³, Minkoo Kim¹

¹Graduate School of Information and Communication Engineering, AJOU University, South Korea

²Department of Industrial and Information Systems Engineering, AJOU University, South Korea

³Centre for Genome Sciences, Division of Biomedical Informatics, National Institute of Health, South Korea

umer@ajou.ac.kr, shin@ajou.ac.kr, cjp@ajou.ac.kr, minkoo@ajou.ac.kr

Abstract

Accurate and less invasive personalized predictive medicine can spare many breast cancer patients from receiving complex surgical biopsies, unnecessary adjuvant treatments and its expensive medical cost. Cancer prognosis estimates recurrence of disease and predict survival of patient; hence resulting in improved patient management. To develop such knowledge based prognostic system, this paper examines potential hybridization of accuracy and interpretability in the form of Fuzzy Logic and Decision Trees, respectively. Effect of rule weights on fuzzy decision trees is investigated to be an alternative to membership function modifications for performance optimization.

Experiments were performed using different combinations of: number of decision tree rules, types of fuzzy membership functions and inference techniques for breast cancer survival analysis. SEER breast cancer data set (1973-2003), the most comprehensible source of information on cancer incidence in United States, is considered. Performance comparisons suggest that predictions of weighted fuzzy decision trees (wFDT) are more accurate and balanced, than independently applied crisp decision tree classifiers; moreover it has a potential to adapt for significant performance enhancement.

Keywords: Prognosis, knowledge based, hybridization, accuracy, interpretability, membership functions, inference, crisp and fuzzy

1 Introduction

According to National Cancer Institute of United States, estimated number of new breast cancer cases in 2008 is 182,460 (female) and 1,990 (male), while the estimation of deaths is 40,480 (female); 450 (male) (National Cancer Institute 2008). Based on current rates, 12.7 percent of women born in US today will be diagnosed with breast cancer at some time in their lives. Surgical biopsies confirm malignancy with high level of sensitivity, but are considered costly and can affect patient's psychology as well (Iliias, Elias and Ioannis 2007).

Copyright (c)2008, Australian Computer Society, Inc. This paper appeared at the Seventh Australasian Data Mining Conference (AusDM 2008), Glenelg, Australia. Conferences in Research and Practice in Information Technology, Vol. 87. John F. Roddick, Jiuyong Li, Peter Christen and Paul Kennedy, Eds. Reproduction for academic, not-for profit purposes permitted provided this text is included.

After confirmation of malignancy, oncologists get indulged into prognostic decision making. Surgery, radiation, chemotherapy, hormone therapy or any combination of them are considered to be the successful treatment methods. But again, selection of treatment method without considering the resulting tumour behaviour can lead to severe consequences. Therefore, being able to predict disease outcomes more accurately would help physicians make informed decisions regarding the potential necessity of adjuvant treatment. This may also lead to the development of individually tailored treatments to maximize the efficacy of treatment.

Ultimately, breast cancer mortality would also be decreased. This idea is the basic motivation behind the growing trend of focusing on accurate and less invasive personalized predictive medicine using machine learning techniques. This approach can spare many breast cancer patients from receiving complex surgical biopsies, unnecessary adjuvant treatments and its expensive medical cost (Yijun et al. 2007). Moreover, in situations where experienced oncologists are not available, predictive models created with data mining techniques can be used to support physicians in decision making with acceptable accuracy (Amir et al. 2007).

Prognosis helps in establishing a treatment plan by predicting the outcome of a disease. There are three predictive foci of cancer prognosis: 1) prediction of cancer susceptibility (risk assessment), 2) prediction of cancer recurrence and 3) prediction of cancer survivability. Focus of this paper is prediction of survivability of a particular patient suffering from breast cancer. "Survival" is generally defined as a patient remaining alive for a specified period of time after the diagnosis of disease. For this research effort, survival is considered as any incidence of breast cancer where the person is still living after 1825 days (5 years) from the date of diagnosis, as recommended by Dursun et al. 2004, Brenner and Gefellor 2002 and Cox DR. 1984.

In this research project, we surveyed various research efforts (Joseph et al. 2006, Ilias and Elias 2007, Dursun et al. 2004, Crockett et al. 2006, Andres and a-reyes 1999 and others will be mentioned later in this paper) in the application of different machine learning techniques to breast cancer prognosis. Some of the obvious trends which account for the motivation behind proposals and experiments presented in this manuscript are:

1. About 70% of all reported studies [Dursun et al. 2004] use Neural Networks which yield "Black Box" models for physicians to interpret.
2. Majority of reported studies in surveys like [Dursun et al. 2004] used machine learning

techniques independently without considering potential in those techniques to cooperate with each other in a hybrid model.

3. Fuzzy logic has been rarely used in cancer prognosis. Being non-crisp, it can act as a natural ally of a physician in prognostic decision making process.
4. Lack of attention paid to data size. Data sets considered are not sufficiently large that can be reasonably partitioned into disjoint training and test sets.

Before going into the details of these observations, let us first analyze their conceptual importance to intelligent cancer prognosis at the grass root level.

The design of any decision support system always faces a critical trade-off known as accuracy-interpretability trade-off. This trade-off becomes very sensitive and important in case of prognostic decision making for cancer treatment. Such data analysis systems, intended to assist a physician, are highly desirable to be accurate, human interpretable and balanced, with a degree of confidence associated with final decision. Accuracy and interpretability are highly conflicting requirements; since complexity of system usually increases as a result of accuracy maximization, resulting in reduced comprehensibility of system's overall behavior. "Improving accuracy while preserving interpretability" is a challenging research issue being actively pursued by designers of decision support systems (Gonzalez et al. 2007, Rafael et al. 2006, Ralf et al. 2004, Cristina and Louis 2004). This trade-off is one of the basic motivational factors behind the model presented in this paper. As mentioned earlier, majority of research efforts in breast cancer diagnosis and prognosis used neural networks (Joseph et al. 2006). This is because relative ease in their use, abilities to provide gradual responses and good classification performance. But in prognostic decision making systems where physicians want to understand and justify the decisions, they act totally as "Black Boxes" with poor interpretability because it is difficult for humans to interpret the symbolic meaning behind the learned weights. Moreover, neural network learning with too many attributes, as in case of breast cancer data (SEER 1973-2003), can result in over-fitting (Joseph et al. 2006).

Unlike neural network, decision trees have always been praised for comprehensibility of their knowledge representation and inference procedures. They have been shown to be problem independent and able to treat large scale industrial applications (Cristina and Louis 2003). Pruned decision tree effectively overcomes over-fitting problem when dealing with large number of attributes (Joseph et al. 2006). The fundamental weakness of decision trees is that the decision boundaries are sharp at each node (for continuous valued attributes), due to which even small changes in attribute values may result in misclassifications (Crockett et al 2006, Cristina and Louis 2003). That is why; they are recognized to be unstable, with high variance. Therefore, decision boundaries need to be softened and there should be a gradual transition between attribute values. Here comes the role of fuzzy

logic, as explained next.

Based upon above mentioned observations, we propose to investigate a hybrid scheme based on weighted fuzzy decision trees (wFDT), as an efficient alternative to crisp classifiers that are applied independently. Fuzzy sets, along with fuzzy logic and approximate reasoning methods, provide the ability to model fine knowledge details (Lotfi A. Zadeh 1983). Accordingly, fuzzy representation is becoming increasingly popular in dealing with problems of uncertainty, noise, and inexact data (Cezary Z. Janikow 1998). That is why we believe, it can act as natural ally of physicians. To help decision trees, the role of fuzzy logic becomes very crucial in softening the sharp decision boundaries because of the elasticity of fuzzy sets formalism. An important aspect of this model is an interesting simultaneous cooperation between Fuzzy Logic and Decision Trees. This bidirectional cooperation tries to soften the accuracy/interpretability trade-off, and can be realized as follows:

1. Fuzzy representation, with its approximate reasoning handles uncertainty and gradual processing to help soften the crisp decision tree boundaries. This results in reduced misclassifications and increased accuracy.
2. There are two approaches of fuzzy modelling (FM) depending on problem domain (Rafael et al 2006):
 - Linguistic FM: based on Linguistic (Mamdani 1974) fuzzy rule based systems. These systems have high interpretability but strive to achieve improved accuracy.
 - Precise FM: based on Takagi-Sugeno 1985 fuzzy rule based systems. Such systems focus on accuracy but lack in interpretability.

Figure-1 describes how linguistic and precise fuzzy modelling tend to achieve required optimal. For breast cancer prognosis problem, precise fuzzy modelling is used i.e. rules are in the form of Takagi-Sugeno 1985. Therefore, such systems lack in interpretability and need to achieve an optimal level of comprehensibility. In this case, decision trees will help such fuzzy representation.

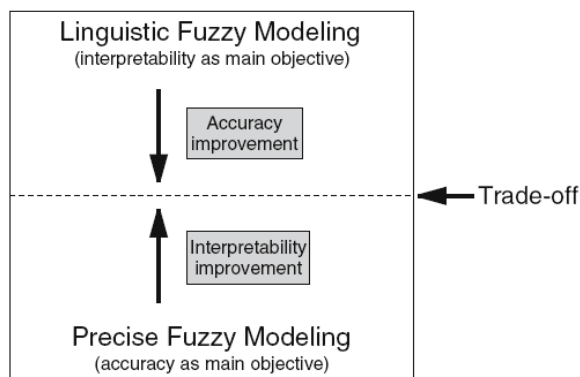


Figure 1: Precise fuzzy modelling tends to achieve optimal interpretability (Rafael et al. 2006)

Fuzzy decision tree (FDT) IF-THEN rules for an m-class pattern classification problem with 'n' attributes

can be written as:

Rule R_i : If x_1 is A_{i1} AND x_2 is A_{i2} AND.....AND x_n is A_{in}
THEN **Class** C_i $i = 1, 2, \dots, N$

where $x = (x_1, x_2, \dots, x_n)$ n-dimensional pattern vector,
 A_{ij} is antecedent Precise fuzzy set (like <30 , ≥ 30),
 C_i Consequent class (one of the given m-classes in labelled data),
 N is the number rules used in a particular model.

Effect of rule weights on fuzzy decision trees is analysed using certainty grade concept. It determines the degree of confidence in the decision of a particular rule. A FDT rule with an associated certainty grade can be written as:

Rule R_i : If x_1 is A_{i1} AND x_2 is A_{i2} AND.....AND
 x_n is A_{in} THEN **Class** C_i with CF_i $i = 1, 2, \dots, N$

Usually, CF_i is a real number in unit interval ($0 \leq CF_i \leq 1$). A special rule weighting technique is used which learns certainty grades from training data. Using certainty grade, compatibility of each rule is calculated for each incoming input record to be classified. The most compatible rule for a particular input record decides its final class. Effect of this weighting is investigated to be an alternative to membership function optimization. A significant performance enhancement is achieved by weighting rules, which also helps oncologists to have certain degree of confidence in the final decision.

The overall aim of this research is to determine the potential of wFDTs for prediction of breast cancer survivability in particular, and breast cancer prognosis in general. wFDT is studied in detail and compared with FDT and crisp decision tree. Experiments were performed rigorously using different combinations of: number of rules in a model, types of fuzzy membership functions and inference techniques. Results show that wFDT achieved much improved prediction accuracy and much reduced variance, as compared with crisp decision tree.

Rest of the paper is organized as follows: section 2 presents the related work, section 3 describes materials and methods used in this research, section 4 presents the experimental evaluations and finally section 5 concludes this manuscript.

2 Related Work

In (Joseph et al. 2006), authors conducted a broad survey of the different types of machine learning methods being used, the types of data being integrated and the performance of these methods in breast cancer prediction and prognosis. To get possible research directions in application of machine learning techniques for cancer prognosis, this survey is the only detailed manuscript (by date) especially for researchers new to this application area. In (Dursun et al. 2004), a comparison between two data mining techniques namely decision trees and neural networks and a statistical method namely logistic regression, is presented. These techniques were applied independently on SEER breast cancer data (SEER

1973-2003) to predict survivability. This research effort concluded that decision trees proved to be the best classifier in that experimental environment. We propose that this performance can be extensively increased using weighted and fuzzified decision tree i.e. wFDT. This was another reason (besides others mentioned earlier) to select decision trees for constructing crisp rule base.

In (Carlos and Moshe 1999), fuzzy rules for cancer diagnosis are generated by randomly selecting data instances from training data, and performing rigorous genetic search evolving different models and then selecting the best ones. According to our approach, an efficient and well tested classifier can be used to build initial rule base, avoiding complexities and optimization errors due to random selection of training records. Moreover, rigorous and repetitive genetic search through a realistically huge cancer patient data (like one used in this research) would result in tedious time and memory complexities.

In recent research efforts for cancer prediction (Ilias et al. 2007 and Leonardo et al 2007), support vector machine (SVM) and neural network (ANN) modelling were performed. In both the cases, main focus was accuracy and no doubt, they would have achieved "high peaks" of accuracy. But a clinician, involved in sensitive decision making about a patient's treatment, demand more than that. Factors including interpretability, system's ability to adopt human reasoning behaviour to deal with uncertainties and performance consistency were ignored.

Let us review research efforts specifically focused on hybridization of fuzzy logic and decision trees, other than cancer prediction domain. To cope with sharp decision boundaries problem, a number of approaches (Cristina and Louis 2003, C.Z Janikow 1999 and 2004, C.W. Olaru 2003 and Yuan et al. 1995) have made use of fuzzy theory to create fuzzy trees. In (Webb and Ting 2005, Umamo et al. 1994) fuzzy tree is induced directly from pre-fuzzified data. The difference between these approaches and the one used in this manuscript is that, they focus on modification of decision tree pruning algorithm and require fuzzy parameters to be set by domain experts. Here fuzzy sets produced can be the outcome of subjective perception. This way an additional aspect of uncertainty is introduced in the system, when there are conflicting opinions between domain experts. In (Crockett et al. 2006), a similar architecture is proposed in which pre-constructed tree is fuzzified without modifying ID3 algorithm. But the rule weighing technique in their inference procedure is very trivial and they did not focus on analysing the strength of certainty grades in system's performance and comprehensibility.

Effect of rule weights in fuzzy rule-based classification systems is examined in (Hisao and Tomoharu 2001). They presented an effective analysis of applying rule weights as an efficient alternative to membership function learning and optimization. Effect of certainty grades on the decision areas of fuzzy rules is illustrated. The larger the certainty grade of a rule, the larger will be its decision area. Their experiments are focused on linguistic values of fixed membership functions. In this work, we have shown the same effect on precise fuzzy modelling, in section 4.

3 Materials and Methods

3.1 Prognostic and Predictive Factors in Breast Cancer

Survival of patients with breast cancer depends on two different types of prognostic factors: 1) Chronological [indicators of how long the cancer has been present (e.g. tumor size)], 2) Biological [indicators of metastatic aggressive behaviour of a tumour (e.g. tumor grade)] as described in (Bundred N.J. 2001). They determine, either or not a particular tumor might respond to a specific therapy. Definitions and effects of some of the most important prognostic factors in breast cancer are given below:

Lymph node status: Lymph nodes, where cancer cells get accumulated (usually under the arm pits). Both number of nodes and level of involvement worsen the prognosis. If the lymph nodes accumulate cancerous cells, they are called positive nodes, otherwise negative.

Stage: Defined by the size of tumor and its spread. Survival is inversely proportion to size of tumor. The probability of long-term survival is better with smaller tumors than with larger tumors (Bundred N.J. 2001). Let us see some examples of breast cancer stage from (National Cancer Institute 2008).

1. Stage-1 is an early stage of invasive cancer. Tumor is no more than 2 cm. Cancer cells remained inside breast.
2. In Stage-2, tumor size can be from 2-5 cm. Cancer cells may or may not spread out of breast.

Similarly, there are in total four types of stages (further divided into many sub-types), in which tumor size keeps on increasing along with spreading of cancer cells. Higher the stage, difficult is survival.

Grade: How does the tumor look like and its resemblance to more or less aggressive tumors. Histological grade is a combination of mitotic rate, nuclear grade and architectural morphological appearance (Rampaul 2001). Here also, patients with grade-1 tumor have higher chances of survival than patients of grade-3.

Figure-1 shows ranking of survivability attributes in terms of their decisive strength, calculated using information gain (IG) applied to breast cancer data, as described in subsection 3.3.

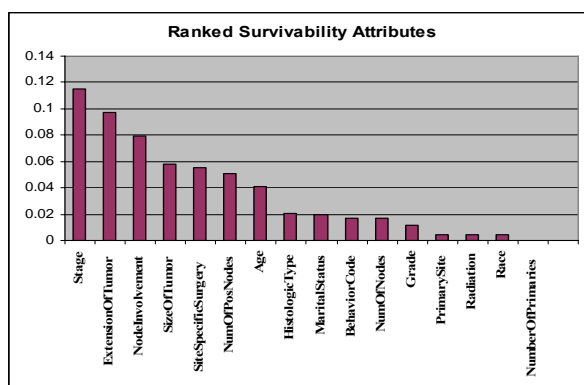


Figure 2: Ranked Survivability Attributes

3.2 Data

In this research work, SEER (Surveillance, Epidemiology, and End Results) data (1973-2003) is used for breast cancer prognosis. Files were requested through website (www.seer.cancer.gov) of SEER program which is a part of Surveillance Research Program at National Cancer Institute. The data set is considered to be the most comprehensive source of information on cancer incidence in USA and SEER program claims quality and completeness of data. A search for term 'SEER' on PUBMED (National Library of Medicine's database), gives a list of more than 1500 publications for the time period of 1978-2008.

Initially, there were 505,367 records each with 86 variables. These variables describe socio-demographic and cancer-specific information of an incidence of cancer. We used Clementine data analysis tool for all preprocessing mentioned below. Considering the aim of survival prediction, a binary target variable is created with values 0 (did not survive) and 1 (survived). To calculate this variable, 'SurvivalTimeRecode' field is used which provides number of years and months of survival after diagnosis. Although much of the time in this research work was spent on data cleansing and preprocessing, only a brief description is given here. To adjust the survival rate, those records were removed in which patient died within 5 years after diagnosis and the cause of death was not breast cancer.

SEER used the same database schema for the data of all anatomical sites (like breast, throat, urinary etc.). So there were many attributes which are common to all cancer types and not specific to breast cancer. Moreover, some redundant variables like recodes and overrides were also removed. For instance, Extent_of_Disease variable aggregates tumor size, # of nodes examined, # of positive nodes examined, lymph node involvement and clinical extension of tumor.

Other than this, variables that had more than 70.0% missing values, categorical variables that had a single category accounting for more than 90.0% of cases, continuous variables that had standard deviation less than 0.1%, and continuous variables that had a coefficient of variation (SD/mean) less than 0, were also removed. For input variable selection, we tried to limit the number of variables and selected only the clinically relevant variables. But for some variables like Stage, 40% of records contained missing values. Since this variable is an important predictor of survivability, instead of deleting whole column, only records containing missing values for this variable were removed.

After an exhaustive pre-processing, final data set with 162500 records, 16 predictor variables and 1 target variable, was constructed. The target variable 'IsSurvival' is a binary categorical variable with possible values '0' (did not survive) and '1' (survived). Table-1 shows the predictive variables and their descriptions, used in our work:

Field	Description
Stage	Defined by the size of cancer tumor and its spread
Grade	How does the tumor looks like and its resemblance to more or less aggressive tumors.
Lymph Node Involvement	None, (1-3) Minimal, (4-9) Significant etc
Race	Ethnicity like White, Black, Chinese etc.
Age at Diagnosis	Actual age of patient in years
Marital Status	Married, Single, Divorced, Widowed, Separated
Primary Site	Presence of tumor at a particular location in body. Topographical classification of cancer
Tumor Size	2-5 cm, at 5cm prognosis worsens
Site Specific Surgery	Information on surgery during first course of therapy whether it was cancer directed or not.
Radiation	None, Beam Radiation, Radioisotopes, Refused, Recommended etc.
Histological Type	The form and structure of tumor
Behavior Code	Normal or aggressive behaviour of tumor have been defined in codes.
# of Positive Nodes Examined	When the lymph nodes are involved in the cancer, they are called "positive."
# of Nodes Examined	Total nodes (positive/negative) examined
# of Primaries	Number of primary tumors (1-6)
Clinical Extension of Tumor	Defines the spread of tumor relative to breast
IsSurvival	Target binary variable defines the class of survival of patient.

Table 1: Breast Cancer Predictive Factors used for Survivability Prediction

Following table shows the important statistics related to above mentioned prognostic factors in training data. Here symbol is assigned to recognize each feature, in the order it appears in training record i.e. Age (A) comes first and number of primaries (P) appears last in training record.

Symbol	Nominal Variable Name	Num of Distinct Values
B	Race	28
C	Marital Status	9
D	Primary Site	9
E	HistologicType1CD	44
F	Behaviour Code	2
G	Grade	5
I	Extension of Tumor	12
J	Node Involvement	10
M	Site Specific Surgery	22
N	Radiation	9
O	Stage	9

Symbol	Numeric Variable Name	Mean	Std.Dev	Range
A	Age at Diagnosis	61.105	14.165	20-106
K	Num of Pos Nodes	24.376	41.238	0-99
H	Tumor Size	103.168	273.144	0-999
L	Num of Nodes	14.033	16.89	0-99
P	Num of Primaries	1.302	0.565	1-6

Table 2: Statistical Description of Predictor Variables

3.3 Decision Trees

Decision tree techniques have always been popular for extracting rules from domain knowledge to classify objects. As mentioned above, to generate fuzzy rules, we opted to use decision trees in the first step of modeling. We used binary C4.5 for all the experiments mentioned in this manuscript. A brief description of its working is given here. To partition the data at each stage of tree, a test is performed to select an attribute with lowest entropy. Information gain (IG) is used as a measure of entropy (H) difference when an attribute contributes the additional information about class C (Witten and Frank 2005).

$$H(C) = -\sum p(c) \log p(c) \quad , c \in C \quad (1)$$

$$H(C|X_i) = -\sum p(x) [\sum p(c|x) \log p(c|x)] \quad , x \in X_i, c \in C \quad (2)$$

$$IG_i = H(C) - H(C|X_i) \quad (3)$$

In equation (1), $p(c)$ is the probability that an arbitrary sample belongs to class 'c'. Equation (2) shows the entropy after observing the attribute X_i for the class 'c' and ' $p(c|x)$ ' is the probability that a sample in attribute branch X_i belongs to class 'c'. We used binary decision tree because it has been proved in previous research (H. Al-Attar 1996, J.R. Quinlan 1990) that they usually outperform the multi-branch trees generated by the original ID3 algorithm. To cater for over fitting problem, the constructed tree is optimized in size using pruning. During experiments, we generated different decision tree models, and analysed the following trend.

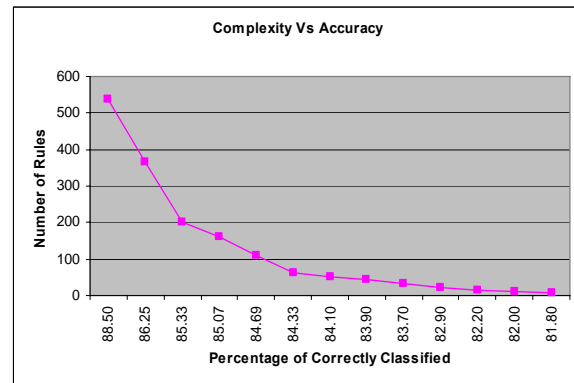


Figure 3: Rules per Model and Accuracy

As shown in figure-3, with a maximum complexity we got maximum accuracy. But for a model with 20 rules to a model with 8 rules, we got the almost similar value of accuracy i.e. around 82%. This trade-off is explained below.

3.3.1 C4.5 Limitations, Interpretability and Model Selection

As described earlier, the fundamental weakness of crisp C4.5 decision tree is that the induced tree will have sharp decision boundaries at each node. In case of continuous attributes, even small changes in attribute values may result in misclassifications. In (Quinlan 1990

R1: IF ClinicalExtensionOfTumor ≤ 40.0 AND AgeAtDiagnosis ≤ 73.0 THEN IsSurvival = YES

R2: IF ClinicalExtensionOfTumor ≤ 40.0 AND AgeAtDiagnosis > 73.0 AND AgeAtDiagnosis ≤ 83.0 And LymphNodeInvolvement ≤ 1.0 THEN IsSurvival = YES

R3: IF ClinicalExtensionOfTumor ≤ 40.0 AND AgeAtDiagnosis > 73.0 AND AgeAtDiagnosis ≤ 83.0 And LymphNodeInvolvement > 1.0 AND NumOfPositiveNodesExamined ≤ 5.0 THEN IsSurvival = YES

R4: IF ClinicalExtensionOfTumor ≤ 40.0 AND AgeAtDiagnosis > 73.0 AND AgeAtDiagnosis ≤ 83.0 And LymphNodeInvolvement > 1.0 AND NumOfPositiveNodesExamined > 5.0 THEN IsSurvival = NO

R5: IF ClinicalExtensionOfTumor ≤ 40.0 AND AgeAtDiagnosis > 73.0 AND AgeAtDiagnosis > 83.0 AND AJCCStage3ed ≤ 10.0 AND AgeAtDiagnosis ≤ 85.0 THEN IsSurvival = YES

R6: IF ClinicalExtensionOfTumor ≤ 40.0 AND AgeAtDiagnosis > 73.0 AND AgeAtDiagnosis > 83.0 AND AJCCStage3ed ≤ 10.0 AND AgeAtDiagnosis > 85.0 THEN IsSurvival = NO

R7: IF ClinicalExtensionOfTumor ≤ 40.0 AND AgeAtDiagnosis > 73.0 AND AgeAtDiagnosis > 83.0 AND AJCCStage3ed > 10.0 THEN IsSurvival = NO

R8: IF ClinicalExtensionOfTumor > 40.0 THEN IsSurvival = NO

Figure 4: Least Complex Model for Interpretability

and Carter et al. 1987), some threshold softening approaches are considered for categorical and continuous attributes. But the results related to splitting of continuous attributes do not show significant improvement (Quinlan 1996). The trade-off presented in figure-3, gets harder due to sharp decision boundaries. For a physician involved in prognostic decision making, both accuracy and interpretability are a must. So we decided to choose interpretability first and leave the accuracy and decision confidence for the second stage.

Although it is difficult to give a precise definition of interpretability, many researchers like (Ralf et al. 2005, Bodenhofer and Bauer 2002, Cordon and Herrera 2000, Jin et al. 1998) have agreed on interpretability involving following aspects:

1. Number of rules should be small enough to be comprehensible.
2. Rule antecedents and consequents should be in easy in structure and it should contain only few features.
3. Rule base should be consistent i.e. similar antecedents should produce similar consequents.
4. Fuzzy system should use features familiar to users.
5. The inference mechanism should produce technically and intuitively correct results.

Based on above recommendations, out of different models generated during experiments, we have chosen the 8-rules least complex model shown in figure-4. This is because from models with 20 rules to model with 8 rules, accuracy remained almost same, as shown in figure-3. Hence, we decided to choose simplest model with 8 rules and an acceptable accuracy 81.5%, to act as base model for FDT and wFDT in next section. Note that, this model contains most decisive factors ranked using Information Gain i.e. extension of tumor, stage, lymph node involvement and positive nodes examined as shown in figure-2. Age also gives a strong idea about survival since it is easier to recover in young age.

3.4 Weighted Fuzzy Decision Trees (wFDT)

In continuation of previous section, we already have an induced crisp decision tree, which partitions the input and output space into n-dimensional space where 'n' is total number of attributes. To convert a sharp transition at decision node into a gradual one, fuzzification is applied to both branches of a decision node (since we are using a binary decision tree). Figure-5 gives a simple and clear idea of crisp and fuzzy classes visually.

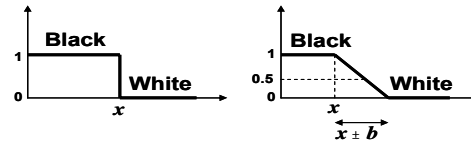


Figure 5: Difference between Crisp & Fuzzy Sets

In figure-5, ' $x \pm b$ ' is a relaxation applied to crisp threshold. To do this, an attribute or decision node is represented by a fuzzy set using a pair of complimentary membership functions M_1 and M_2 , as elaborated in figure-6.

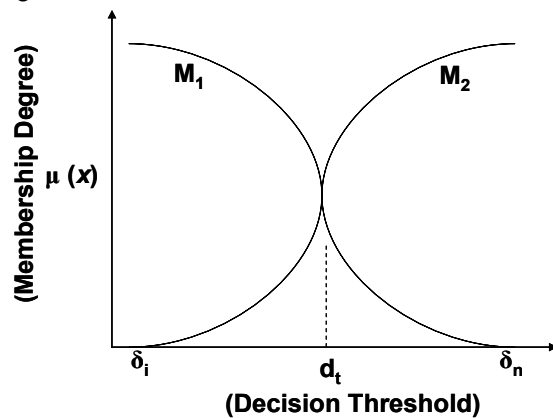


Figure 6: Complimentary Membership Functions over domain δ_i and δ_n

Fuzzy region is defined around a crisp threshold 'dt', already defined at each decision node or attribute by C4.5

algorithm. A membership function defines degree of membership $\mu(x)$, of a particular input value 'x', into a fuzzy set. This degree lies in the range 0 to 1, with $\mu(x)=0$ means 'no membership' and $\mu(x)=1$ means 'full membership'. Membership degree or value is a key concept which ensures that sharp transition concept ceases to exist. Some examples of membership functions will be explained later in this section. Although there can be many smart ways to initially specify the domain, lower bound ' δ_i ' and upper bound ' δ_n ' of a membership function, we stick to a common and simplified domain specification. Since decision threshold ' d_i ' is already generated at each node of DT and remains fixed, domain delimiters can be calculated as:

$$\delta_i = d_i - f * \sigma \quad \text{and} \quad \delta_n = d_i + f * \sigma \quad (4)$$

here δ_i and δ_n are lower and upper bounds of membership function, respectively. ' σ ' is the standard deviation of the domain attribute. It determines how tightly an attribute's values are bound around its mean value. It helps in guessing what proportion of an attribute would be assigned partial membership degree. ' f ' is the fuzzification applied around decision threshold ' d_i '. Studies have shown empirically that ' f ' is usually chosen in the domain '0-5'. This is because larger values of ' f ' would introduce too much fuzzification and decision making process would become unclear. For our experiments, ' $f=2$ ' gave the optimal results.

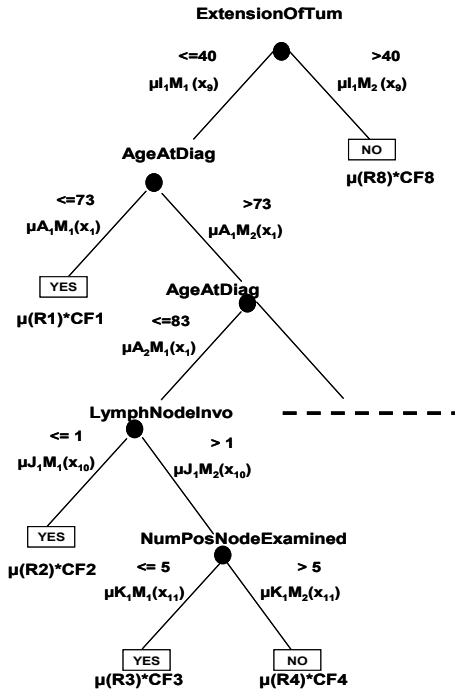


Figure 7: Fuzzified Decision Tree

A portion of fuzzified decision tree is shown in Figure-7 It describes how an attribute at each node is fuzzified using left and right complimentary membership functions (M1, M2). Crisp rules R_1 and R_8 , in figure-4, can be obtained in fuzzified form by traversing the left most and right most paths of the tree in figure-7. An important feature of this

approach is that, it preserves the decision thresholds and symbolic structure obtained from induced tree.

3.4.1 Fuzzy Inference

The approach used is very simple and interesting in a way that, for classifying an example, all the rules contribute their knowledge to some degree. A brief description is given below: (for classification of an incoming record)

1. For each path (or rule) of the fuzzy decision tree, a cumulative membership grade is calculated by applying an intersection operator (like Yager or Zadeh operator) to the set of individual membership function values at each branch on that path. For example, cumulative membership grade of first rule R_1 (left most path from root to leaf, in figure-7) is computed as:

$$\mu(R_1) = \cap \{ \mu_{I_1} M_1(x_9), \mu_{A_1} M_1(x_1) \} \quad (5)$$

Here ' $\mu_{I_1} M_1(x_9)$ ' is the membership function of fuzzy set "ExtensionOfTumor ≤ 40 ". ' I_1 ' means first occurrence of 'ExtensionOfTumor' node in tree. ' M_1 ' means this function is left complimentary function. ' x_9 ' is 'ExtensionOfTumor' value in input record. This membership function will compute membership value of the input record, for a particular branch in rule path. Now we have 8 cumulative membership grades. Each of such grades is multiplied by the rule weight or certainty factor CF_i . Section 3.4.3 explains in detail about the computation of CF and its effect on fuzzy rule based classification system. This weight is calculated for each rule using training data. Rule weight has a great significance in fuzzy inference and here it is used as an alternative of Genetic Algorithms for parameter optimization.

2. Finally, all the products ($\mu(R_i) * CF_i$) are combined using *union operator*, and a rule (e.g. Zadeh) or a class (e.g. Yager) with maximum membership grade, will decide the class of incoming record.

$$\text{Decision} = \cup \{ \mu(R_1) * CF_1, \mu(R_2) * CF_2, \dots, \mu(R_8) * CF_8 \} \quad (6)$$

We used Yager and Zadeh (Witten and Frank 2005) inference operators for intersection and union.

3.4.2 Fuzzy Membership Functions

We have used different types of fuzzy membership functions like Linear, sigmoid, convex and concave membership functions (Earl Cox 1994) to evaluate wFDTs. A brief description of these membership functions is given below:

$$\text{Linear}(\delta_i, \delta_n, x) = \begin{cases} 0 \rightarrow x \leq \delta_i \\ x - \delta_i / \delta_n - \delta_i \rightarrow \delta_i \leq x \leq \delta_n \\ 1 \rightarrow x \geq \delta_n \end{cases}$$

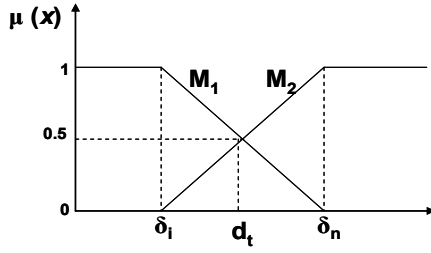


Figure 8: Linear Membership Function

Here 'x' is the input value of an attribute. 'δi' generates zero membership while 'δn' generates maximum membership i.e. '1'. Both 'δi' and 'δn' are computed from equation-4.

$$\text{Sigmoid } (x; \delta_i, \delta_n, \beta) = \begin{cases} 0 \rightarrow x \leq \delta_i \\ 2 \left(\frac{(x - \delta_i)}{(\delta_n - \delta_i)} \right)^2 \rightarrow \delta_i \leq x \leq \beta \\ 1 - 2 \left(\frac{(x - \delta_n)}{(\delta_n - \delta_i)} \right)^2 \rightarrow \beta \leq x \leq \delta_n \\ 1 \rightarrow x \geq \delta_n \end{cases}$$

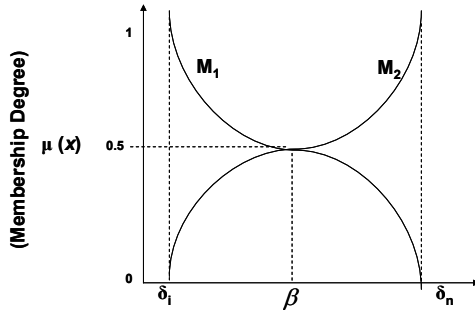


Figure 9: Sigmoid Membership Function

Here β is defined as half membership point $(\delta_i + \delta_n)/2$. It represents known distribution of sample space, and assumed to be $\beta = dt$. Fuzzification gets to its maximum as 'x' gets closer to 'δn'.

$$\text{Convex } (\delta_i, \delta_n, x) = \begin{cases} 0 \rightarrow x < \delta_i \\ 1 - \left[2 * \frac{(x - \delta_n)}{\delta_n} \right] \rightarrow \delta_i \leq x \leq \delta_n \\ 1 \rightarrow x > \delta_n \end{cases}$$

$$\text{Concave } (\delta_i, \delta_n, x) = \begin{cases} 0 \rightarrow x < \delta_i \\ \frac{(x - \delta_i)}{\delta_n} - \delta_i \rightarrow \delta_i \leq x \leq \delta_n \\ 1 \rightarrow x > \delta_n \end{cases}$$

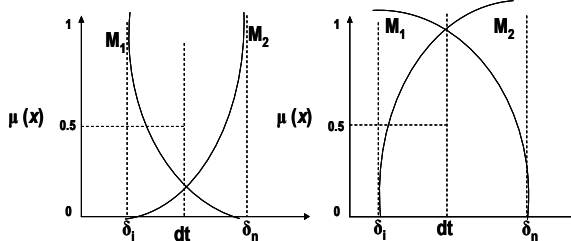


Figure 10: Convex (Left) and Concave (Right) Membership Function

In convex membership function, the membership grade at point of intersection is significantly less than 0.5. As the value of 'x' equals 'dt', membership would be low in both M1 and M2 representing low confidence in both child branches of decision node in binary tree.

Concave membership function assigns higher membership grade to the branching threshold, which implies strong confidence in both child branches of decision node.

3.4.3 Effect of Weights on Fuzzy Rules

In this section, we have discussed a very important and significant aspect of wFDT modelling. Effect of weights, learnt from data, on fuzzy rule base is analysed. For the performance enhancement of fuzzy rule based systems, there has always been a room for membership function optimization through learning or other adjustment techniques. This analysis is based on an argument that learning of certainty grades (rule weights) can partially replace the adjustment of membership function. A few aspects of this analysis are referenced from (Hisao and Tomoharu 2001, Nauck and Kruse 1998) in which they discussed rule weighing for linguistic fuzzy modelling.

This concept is based on an assumption that modifying a membership function can deteriorate the comprehensibility of fuzzy classification system. It can also introduce a gap between modified membership function and expert's knowledge about that function. On the other hand, learning single real number is a relatively easier task, and it improves the classification accuracy of fuzzy rule based system. Another significant importance is that it represents the strength of each rule, in other words the confidence in rule's decision. This would help physician in establishing his confidence in a particular rule.

In fuzzy rule based systems where inference is based on one winner rule classification, if certainty grades are not used, the rule with maximum compatibility grade (membership value) for a record to be classified, decides the class (detailed inference mechanism is described in next section). Following expression formalize this concept.

$$\mu_j^*(X) = \max (\mu_j(X) | j=1,2,...N) \quad (7)$$

This expression simply shows that the rule with maximum membership value for an input record 'X', will decide its class. 'N' is total number of rules. Based on this, each rule has a particular decision area as shown in figure-11.

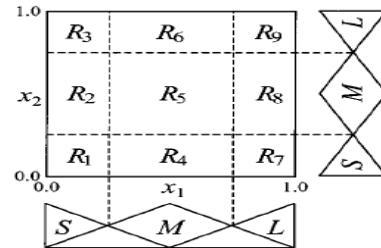


Figure 11: Decision area of fuzzy rules (Hisao et al. 2001)

Decision areas of rules without certainty grades are proved to be rectangular (Kenchuva 2000). These decision areas can be modified and adapted, by learning certainty grades (rule weights) from data, to alternatively affect the membership functions without explicitly modifying them. Modified decision areas will automatically result in modified class boundaries. Figure-12 shows the effect of certainty grade on fuzzy sets.

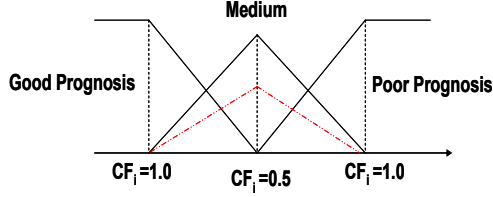


Figure 12: Effect of Certainty Grade on Fuzzy Set

The red dashed line shows the product of compatibility grade and certainty factor (CF) of a rule which modified class boundary. Now the rule with maximum of this product will be the winner as expressed by the following equation:

$$\mu_{j*}(X)CF_{j*} = \max (\mu_j(X)CF_j \mid j=1,2,\dots,N) \quad (8)$$

Certainty grades or rule weights are calculated for each rule as:

1. When consequent class of Rule is YES (or 1)

$$CF_i = \frac{\beta_{ClassYES}(R_i) - \beta_{ClassNO}(R_i)}{\beta_{ClassYES}(R_i) + \beta_{ClassNO}(R_i)} \quad (9)$$

2. When consequent class of Rule is NO (or 0)

$$CF_i = \frac{\beta_{ClassYES}(R_i) - \beta_{ClassNO}(R_i)}{\beta_{ClassYES}(R_i) + \beta_{ClassNO}(R_i)} \quad (10)$$

$$\text{Where } \beta_{ClassK}(R_i) = \sum_{x \in ClassK} \mu_i(x), k = YES, NO$$

In simple words, for each rule 'R_i' its combined membership value for all the training patterns of class 'YES' ($\beta_{ClassYES}(R_i)$), and its combined membership value for all the training patterns of class 'NO' ($\beta_{ClassNO}(R_i)$) is computed to get its certainty grade using above equations. Certainty grade values lies in the range $0 \leq CF_i \leq 1$, which means when all compatible patterns with rule R_i (those with $\mu_i(x) > 0$ for R_i) belong to the same class, CF_i equals one.

This analysis, resulted in significantly increased performance, as mentioned in next section.

4 Performance Evaluation

Experiments were performed using WEKA, Matlab and Java on a Pentium PC at 1.7GHz with 1.5GB RAM. Execution time for calculating decision tree with different kernel functions varied for 6 to 12 seconds. Out of 162500 records, 30000 records as training and 10600 as test data were obtained using uniform random selection, taking into account the overlapping factor (stratified sampling).

4.1 Accuracy, Sensitivity and Specificity

In this study, we have used following three performance measures:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (11)$$

$$\text{Sensitivity} = \frac{TP}{TP + FN} \quad (12)$$

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (13)$$

where TP, TN, FP and FN denotes true positives, true negatives, false positives and false negatives, respectively.

4.2 10-Fold Cross Validation

k-Fold cross validation is used to minimize the bias associated with random sampling of training and test data samples in comparing predictive accuracy of two or more methods (Dursun et al. 2004). Here the whole data set is randomly split into 'k' mutually exclusive subsets of approximately equal size. Classification model is trained and tested k times. Each time it is trained on all but one fold. For example, we used 10-fold cross validation because empirical studies (Kohavi 1995, Breiman et al. 1984) have shown that 10 folds are appropriate to optimize the testing time and minimize the bias and variance associated with validation process. In this case, data is split into 10 mutually exclusive subsets (using stratified sampling). Each of these 10 folds is used once to test performance of classifier, while other 9 are used for training.

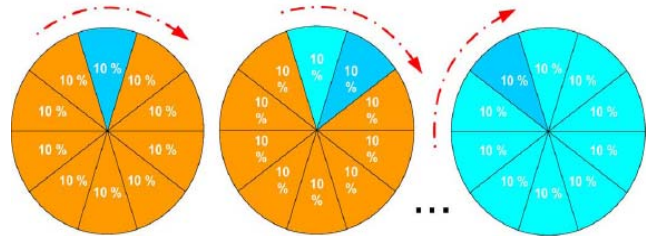


Figure 13: 10-Fold Cross Validation (Dursun et al.)

Cross validation estimate of classifier's overall accuracy is calculated by simply taking the mean of 'k' individual accuracy measures. Table-3 shows the 10-fold cross validation estimates of crisp decision trees, fuzzy decision trees (FDT) and weighted fuzzy decision trees.

Fold #	Crisp Decision Tree					EDT					wFDT				
	Confusion Matrix		Accuracy	Sensitivity	Specificity	Confusion Matrix		Accuracy	Sensitivity	Specificity	Confusion Matrix		Accuracy	Sensitivity	Specificity
1	7810	1451	0.812	0.85	0.607	7943	1318	0.8313	0.8576	0.666	8356	1090	0.8914	0.885	0.944
	551	850				481	920				68	1148			
2	7755	1496	0.8028	0.838	0.571	7923	1328	0.8279	0.8564	0.641	8346	1100	0.8895	0.8835	0.94
	606	805				506	905				78	1138			
3	7705	1526	0.7962	0.8346	0.5485	7938	1314	0.8299	0.8579	0.646	8359	1082	0.8913	0.8854	0.9369
	646	785				499	911				77	1144			
4	7825	1421	0.8111	0.846	0.581	7954	1290	0.8324	0.86	0.6495	8347	1094	0.8890	0.8841	0.926
	593	823				497	921				90	1131			
5	7741	1471	0.7988	0.84	0.535	7950	1293	0.8329	0.86	0.655	8367	1079	0.8923	0.8858	0.9433
	674	776				489	930				69	1147			
6	7680	1502	0.79	0.836	0.51	7940	1321	0.831	0.8574	0.654	8380	1066	0.8947	0.8871	0.9539
	724	756				485	916				56	1160			
7	7845	1421	0.8169	0.8466	0.6196	7905	1351	0.8295	0.8540	0.6685	8320	1116	0.8914	0.8850	0.9666
	531	865				466	940				41	1185			
8	7621	1645	0.7837	0.8224	0.5265	7976	1280	0.8397	0.8617	0.6948	8335	1101	0.8877	0.8833	0.9217
	661	735				429	977				96	1130			
9	7820	1446	0.8157	0.8439	0.6282	7923	1328	0.8279	0.8564	0.641	8390	1046	0.8972	0.8891	0.9592
	519	877				506	905				50	1176			
10	7700	1566	0.7831	0.8310	0.4656	7966	1285	0.835	0.8611	0.664	8359	1082	0.8913	0.8854	0.9369
	746	650				474	937				77	1144			
Mean			0.8010	0.8388	0.5592			0.8318	0.8583	0.6579			0.8916	0.88537	0.94285
St Dev			0.0121	0.0078	0.0492			0.00338	0.00229	0.0154			0.00261	0.00164	0.01329

Table 3: 10-Fold Cross Validation Estimation of Three Models

Results in table-3, describe significant and consistent performance enhancement using wFDT as compared with crisp decision trees. Figure-13, describe the Receiver Operating Characteristics (ROC) for FDT (blue) and wFDT (red).

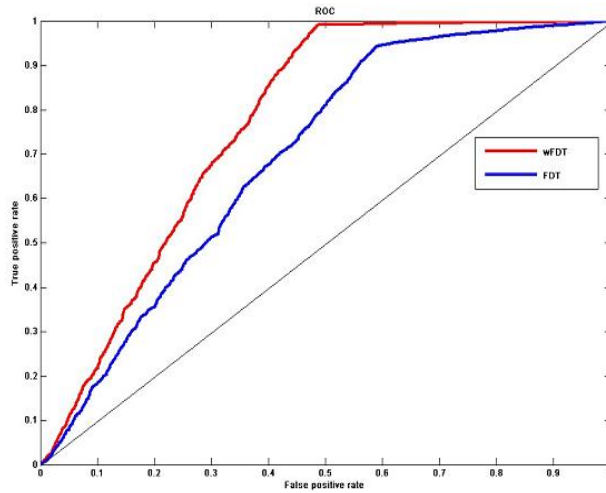


Figure 14: ROC Curve Analysis of FDT and wFDT

	AUC
FDT	0.69
wFDT	0.77

Table 4: AUC Measures

In the results presented in table-3, the variance of crisp decision tree and weighted fuzzy decision trees needs to be analysed. There is an obvious uncertainty in crisp decision tree performance. On the other hand, the estimations of FDT and wFDT are consistent through out the 10 folds. This high variance of decision tree estimations is due to sharp, inflexible decision boundaries.

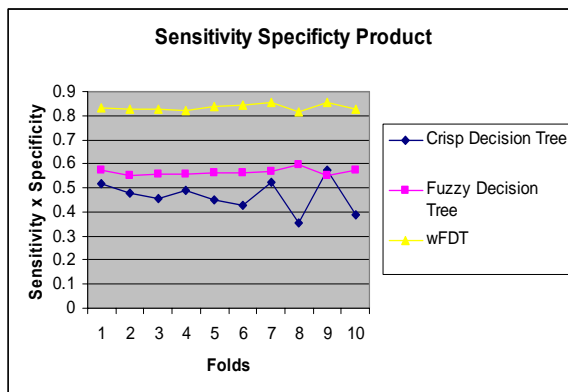


Figure 15: Sensitivity Specificity Product

Figure-14 graphically depicts the value of wFDT to a clinician who is now confident about the survival chances of patient, with such a high sensitivity specificity product. Again consistency (variance) of three curves should also be noticed.

The effect of weights on the performance of fuzzy

decision trees has become obvious. There is a significant increase in accuracy of wFDT as compared to FDT. High performance of wFDT in terms of specificity proves its robustness specially when there is some bias like 'class imbalance problem' or variance due to sampling bias. Table-5 shows that wFDT performed best for Yager inference and sigmoid membership function.

Inference Tech.	Linear	Sigmoid	Convex	Concave
ZADEH	11.5	10.40	12.56	13.01
YAGER	10.45	10.02	12.07	12.75

Table 3: Average Error Rate on Unseen Date for wFDT

Performance comparisons suggests that weighted fuzzy decision trees have good compatibility with all the requirements and features of an accurate and comprehensible prognostic decision making system, mentioned in introduction and related work sections.

5 Conclusion

In this paper, we have shared our experiences of investigating intelligent machine learning techniques for breast cancer prognosis analysis. We analyzed the possible potential of fuzzy logic based classifiers, and came up with a conclusion that they are fit to act as natural allies of a physician involved in predictive medicine. Moreover, they can proficiently manage contrasting requirements of accuracy, interpretability and balance in decision. When we say balance, obviously it is not crisp. Interesting cooperation between DTs and Fuzzy theory helps to realize this aim.

After these experiments, we outlined some future dimensions which can help wFDTs to prove their potential as a strong classifier and predictor in cancer prognosis. Optimization through rule weights or genetic algorithms; an analysis is required, since rule weights, domain delimiters and inference parameters are the key players affecting accuracy. Cooperation among rules in decision making process can also be a good area research in this perspective.

We are committed to explore the strengths of wFDTs for personalized predictive medicine, which is indeed a growing trend in personalized healthcare.

Acknowledgment

This research is supported by Foundation of ubiquitous computing and networking (UCN) project, the Ministry of Knowledge Economy (MKE) 21st Century Frontier R&D Program in Korea and a result of subproject UCN 08B3-S2-10M. Moreover, Hyunjung Shin would like to gratefully acknowledge support from Post Brain Korea 21 and the research Grant from Ajou University.

6 References

- Amir A, Evans DG, Shenton A, Lalloo F, Moran A, Boggis C, Wilson M, Howell A (2003): Evaluation of breast cancer risk assessment packages in the family history evaluation and screening programme. *J Med Genet* 40: 807–814
- Brenner H, Gefeller O, Hakulinen T. A computer program for period analysis of cancer patient survival. *Eur J Cancer* 2002;38(5):690–5.
- Bundred NJ. Prognostic and predictive factors in breast cancer. *Cancer Treatment Rev* 2001;27:137–42
- Breiman L, Friedman JH, Olshen RA, Stone CJ. Classification and regression trees. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software; 1984.
- Cox DR. Analysis of survival data. London: Chapman & Hall; 1984.
- Carlos Andres, Pen a-Reyes, Moshe Sipper: A fuzzy-genetic approach to breast cancer diagnosis. *Artificial Intelligence in Medicine* 17 (1999) 131–155
- C. Carter, J. Cartlett, Assessing credit card applications using machine learning, *IEEE Expert*, Fall Issue (1987) 71–79.
- Dursun Delen*, Glenn Walker, Amit Kadam (2004): Predicting breast cancer survivability: a comparison of three data mining methods. *Artificial Intelligence in Medicine* (2005) 34, 113–127
- D. Nauck and R. Kruse, “How the learning of rule weights affects the interpretability of fuzzy systems,” in Proc. 7th IEEE Int. Conf. Fuzzy Systems, Anchorage, AK, May 4–9, 1998, pp. 1235–1240.
- Earl Cox (1993) The Fuzzy Systems Handbook: A Practitioner's Guide to Building, Using, and Maintaining Fuzzy Systems
- H. Brenner, O. Gefeller (2002): A computer program for period analysis of cancer patient survival. *Eur J Cancer* 2002;38(5):690–5.
- H. Al-Attar, Improving the performance of decision tree induction in non-deterministic classification domains, M.Phil. Thesis, Manchester Metropolitan University, Manchester, England, 1996.
- Ilias Maglogiannis, Elias Zafropoulos and Ioannis Anagnostopoulos (2007): An intelligent system for automated breast cancer diagnosis and prognosis using SVM based classifiers. *Journal of Applied Intelligence*.
- Ian H. Witten and Eibe Frank. Data Mining: Practical Machine Learning Tools and Techniques, 2nd Edition. San Francisco: Morgan Kaufman; 2005
- J. Quinlan, Induction of Decision Trees, Machine Learning, vol. 1, Kluwer Academic Press, Dordrecht, 1986 pp. 81–106
- Joseph A. Cruz, David S. Wishart (2006): Applications of Machine Learning in Cancer Prediction and Prognosis. *Cancer Informatics*
- J.R. Quinlan, Probabilistic decision trees, in: Y. Kockatoft, R. Michalshi (Eds.), Machine Learning, vol. 3: An AI Approach, 1990, pp. 140–152.
- J.R. Quinlan, Improved use of continuous attributes in C4.5, *J. Artificial Intelligence Res.* 4 (1996) 77–90.
- Keeley Crockett, Zuhair Bandar, David Mclean, James O'Shea: On constructing a fuzzy inference framework using crisp decision trees, *Fuzzy Sets and Systems* 157 (2006) 2809 – 2832
- Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Wermter S, Riloff E, Scheler G, editors. The Fourteenth International Joint Conference on Artificial Intelligence (IJCAI) San Francisco, CA: Morgan Kaufman; 1995. p. 1137–45.
- L. Sison, E. Chong, Fuzzy modeling by induction and pruning of decision trees, *IEEE Symposium on Intelligent Control U.S.A.*, 1994, pp. 166–171
- Lotfi A. Zadeh The role of fuzzy logic in the management of uncertainty in expert systems,” *Fuzzy Sets Syst.*, vol. 11, pp. 199–227, 1983.
- L. I. Kuncheva (2000): How good are fuzzy if-then classifiers, *IEEE Trans. Syst., Man, Cybern. B*, vol. 30, pp. 501–509, Aug. 2000.
- M. Umano, H. Okamoto, I. Hatono, H. Tamura, Generation of fuzzy decision trees by fuzzy ID3 algorithm and its application to diagnosis by gas in oil, Japan–U.S.A. Symposium, 1994, pp. 1445–1450.
- Mamdani EH (1974) Applications of fuzzy algorithms for control a simple dynamic plant. In: Proceedings of the IEEE 121(12):1585–1588
- National Cancer Institute USA (2008): Breast Cancer Statistics <http://www.cancer.gov>
- O. Cordon, F. Herrera, L. Magdalena (Eds.), Trade-off between Accuracy and Interpretability in Fuzzy Rule-Based Modelling, Studies in Fuzziness and Soft Computing, Physica, Heidelberg, 2002.
- O. Cordon, F. Herrera, A proposal for improving the accuracy of linguistic modeling, *IEEE Trans. Fuzzy Systems* 8 (3)(2000) 335–344.
- U. Bodenhofer, P. Bauer, A formal model of interpretability of linguistic variables
- Yijun Sun, Steve Goodison, Jian Li, Li Liu and William Farmerie (2007): Improved breast cancer prognosis through the combination of clinical and genetic markers. *Bioinformatics*, Vol. 23 no. 1 2007, pages 30–37
- Y. Jin, W. von Seelen, B. Sendhoff, An approach to rule-based knowledge extraction, in: Proc. IEEE Conf. on Fuzzy Systems, Anchorage, Alaska, 1998, pp. 1188–1193.

Identifying Stock Similarity Based on Multi-event Episodes

Abhi Dattasharma¹

Praveen Kumar Tripathi²

Sridhar G³

¹ Intermedia Softech Pvt. Ltd.

Bangalore, India

Email: abhi@intermediasoftech.com

² ARG, SCSL Bangalore, India

Email: praveen_tripathi@satyam.com

³ ARG, SCSL Bangalore, India

Email: sridharg@satyam.com

Abstract

Predicting stock market movements is always difficult. Investors try to guess a stock's behavior, but it often backfires. Thumb rules and intuition seems to be the major indicator. One approach suggested that instead of trying to predict one particular stock's movement with respect to the whole market, it may be easier to predict a stock A 's movement based on another stock B 's movement; the reason being that A may get affected by B after B 's movement, giving the investor invaluable time advantage. Evidently, it would be very useful if a general framework can be introduced that can predict such dependence based on any user defined criterion. A previous paper laid a basic framework for a single event based criterion, but that was not enough where multiple criteria were involved. This paper gives a general framework for multiple events. We show that it is possible to encode a time series as a string, where the final representation depends on the user defined criterion. Then finding string distances between two such encoded time series can effectively measure dependence. We show that this technique is more powerful than the '*Pairs Trading strategy*' as varied user defined criterion can be handled while detecting similarity. We apply our technique with one practical user defined criterion. To the best of our knowledge, this is the first attempt to find similarity between stock trends based on user defined multiple event criteria.

Keywords: Time series, stocks, similarity

1 Introduction

Efficient market hypothesis (EMH) [22] says that the price of a stock completely encapsulates all the information available, making it impossible to predict the

movement of prices. Despite that, algorithmic trading [1] tries to identify possible "*windows of opportunities*" automatically, using algorithms, for performing stock trading. These windows of opportunities are determined by predictions from the historical data. Thus, algorithmic trading basically involves predicting stock trends and making financial decisions based on historical stock data. Stock data is a time series data [7], describes how the stock values behave over time. The most frequently used values for visualizing and predicting stock movement over a time window (Week/Day/Time) are, price at trading opening time, maximum value on that window, lowest value on that window, price at closing time and total volume of transaction within that window.

One interesting related problem is to find out how one stock affects another. We are trying to detect any kind of dependency between two stocks. It is quite important from the viewpoint of investors; as it gives the investor an idea about how the markets are moving and also some unforeseen insights. Suppose an investor knows that a stock B closely follows stock A within a three-day period. In current market, the investor finds that A is increasing steadily for the last two days. This now tells the investor that B is about to go up, and investing in B is a good idea. Thus the investor found an opportunity using " $A - B$ " relationship data instead of B 's pattern alone. This also gives the investor an invaluable time advantage.

In the well known *Pairs Trading strategy* [4], [21] two similar stocks are chosen so that, if the behavior of one stock outperforms the average performance it is sold short, where as, if the behavior of one stock falls below average, it is bought in the hope that it will reach average behavior sometime. The key concept behind pairs trading strategy is that a well chosen pair tend to cancel each other's deviations and thus, a loss in one will be gained in the other.

Evidently, it would be very useful if a general framework can be introduced that can predict such dependence between stocks, that could be used for pairs trading strategy. For pairs trading, the most commonly used index is a correlation coefficient [19], and investors often also check other metrics like relative strength index [23, 3] and Bollinger band [5, 3].

There exists some attempt to compare two differ-

Copyright ©2008, Australian Computer Society, Inc. This paper appeared at the Seventh Australasian Data Mining Conference (AusDM 2008), Glenelg, Australia. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 87, John F. Roddick, Jiuyong Li, Peter Christen and Paul Kennedy, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

ent stocks' similarity. The indicators, Price Ratio and Price Comparison (see [3] for a nice list) do compare two stocks' prices, but this is for the purpose of finding strong stocks rather than dependences between stocks. People have used clustering techniques [12], mining association rules from database of transactions [20, 18, 16], and geometric properties [9, 6] to find stock similarity. The metric distances and methods based on topological properties and random matrix theory [8, 13] have also been used for the purpose. These methods were complicated and could not provide simple yet practical indicators useful for investors which are applicable both for short term and long term dependencies. Also, all these metrics invariably try to find similarity between stocks in terms of one fixed distance measure. In [11], this problem was investigated and three simple metrics were introduced which could effectively predict two stocks' dependence on each other. However, those metrics were applicable only for a certain class of dependence (for example, if the opening price pattern of one stock follows the opening price pattern of another). Naturally, it would be far more useful if some generic framework could be devised that would enable an investor check for dependence between stocks with respect to a criterion defined by the investor.

In article [10], a framework based on events and episodes for doing the above has been described. It has been shown that if the user criterion is simple enough, then using the concepts of events and episodes, first introduced by Manilla et. al. [17], it is possible to encode a stock's time series data using a binary alphabet, and then similarity between two stocks can be measured using a string distance metric. However, that framework was not applicable to complex criteria that used more than one event in the time series. In this paper, we give a general framework that can take multiple events and combine them effectively. We show that, given a user defined criterion involving multiple events, it is possible to encode any stock's time series data using a finite alphabet, and then the distance between these two encoded series efficiently measures dependence. We give some practical results for one particular user defined criterion. We also show that the proposed technique can be used effectively for pairs trading; in fact, this technique is more powerful as it can model different user requirements while finding similarity.

Our data set is taken from Standard and Poor 500 [2]. This data shows daily price and volume fluctuations (opening price, closing price, maximum price, minimum price and volume) over an year for some well-known stocks. Each stock is a series of 252 numbers; thus we have a time series of length 252.

This paper is organized as follows. In Section 2, we define events and episodes. Section 3 describes the framework. Section 4 gives results, and Section 5 concludes the paper.

2 Events and Episodes

Manilla et. al. introduced the idea of an event for time series in [17]. An event is an occurrence of a particular type which has a time stamp attached to

it. Given a set \bar{E} of event types, an event α is a pair $(A, t) \in \bar{E} \times \mathcal{N}$, \mathcal{N} set of natural numbers, where $A \in \bar{E}$ is an event type and t is an integer, the occurrence time of the event. Mathematically, a time series can be seen as an ordered sequence of events $\{\alpha_i\} = \{(e_i, t_i)\}$ where e_i is an occurrence of a particular type and t_i is the time stamp. For example, for a stock time series, the set of event types can be (daily opening price, daily closing price, daily trading volume, daily maximum price and daily minimum price) and the time series can be described by any of these four event types by giving the value associated with the day.

An episode is a sequence of events. Manilla et. al. [17] defines an episode in the following way. An episode is a triple $(V, <, g)$ where V is a set of nodes, $<$ is a partial order on V , and $g : V \rightarrow \bar{E}$ is a mapping associating each node with an event type. Manilla et. al. [17] gave algorithms for identifying frequently occurring episodes in a sequence of events.

Therefore, each stock data time series can be viewed as a series of events, where each event is a value (the opening price, or closing price, or volume traded and so on) associated with a time. Now, a user can define an episode as a predefined sequence of events. For example, the user can define "three successive days of trading volume increase" as an episode ($T_i < T_{i+1} < T_{i+2}$, $T_i = \text{Trading Volume value on } i^{\text{th}} \text{ day}$).

Note that the definition implicitly assumes that an episode is defined completely only by the event type and their partial ordering. This is not enough for defining somewhat complicated episodes where conditionals or temporal behavior comes into play, for example, the episode "closing price lower than opening price, occurring at least twice a week" is difficult to define using the above definition. The problem here lies in the fact that the episode does not only need a sequence of events, but also a quantifier on the time.

As we are trying to determine similarities between stock prices based on their time series data, we need to find episodes in that data in a time sequential order. We define an episode as a sequence of events together with a restriction on the values as well as the time stamps of the events. Thus, an episode dependent on a single event is defined as:

$$E = \{(e_i, t_i) \in \bar{E} \times \mathcal{N} : f(e_i), g(t_i)\},$$

where $f(e_i)$ and $g(t_i)$ are functions of e_i and t_i respectively, and events are always found from left to right. Suppose O_i is the opening price on i^{th} day for the stock data. An example f and g can be seen from the episode "Opening price of stock is more than 10 USD for two days in succession", which is equivalent to $O_i > 10, O_j > 10, j = i + 1$. Here the event type is opening price, f defines the relationship between opening price and the value 10, and g defines the relationship between i and j . Note that, as we are defining an episode in terms of a set of events detected from left to right, each event's position in time gets fixed and the ordering is automatic. We will call a sequence of events which satisfy

the complementary conditions as a complementary episode. Thus, for the above definition of episode, a complementary episode would be “Opening price of stock is more than 10 USD but not for more than one day in succession”, which can happen by the opening price falling below 10 USD on the second day. Note that “Opening price of stock is not more than 10 USD for two days in succession” is not the right definition, as it can also include two successive days on which the opening prices are lesser than 10 USD.

Note that the above discussion is based on episodes defined by one event. This may not always be the case. For example, consider the following criterion:

“Stock opening price increases on the next day of a financial press release”. The complementary episode will be “Stock opening price does not increase on the next day of a financial press release”. This can happen either by the opening price falling on the next day of a press release or the opening price staying at the same value on the next day of a press release.

Here, there are two events, opening price increase, and a financial press release. If we use each event independently, and try to form the episodes, we will get something like the following:

$$E_1 = \{(O_k, t_k) \text{ such that } O_k > O_{k-1}, k = 2, \dots, n\}, \\ E_2 = \{(P_i, t_i), i = 1, \dots, n-1\}$$

However, the true episode will be given by:

$$E = \{(P_i, t_i), (O_k, t_k), k = i+1, i = 1, \dots, n-1, O_k > O_i\}$$

where O_k is the opening price on the k -th day, and P_i is a press release that has happened on the i th day. Thus, the relationship between the two events must also be brought in as a set of constraints. This implies that an episode using multiple events must modify its constraint set to take into account any relationship between the events themselves.

We can now define an episode, defined for multiple events $\{e_1, e_2, \dots, e_m\}$ as:

$$E = \{(e_1, t_1), \dots, (e_m, t_m), (e_i, t_i) \in \bar{E} \times \mathcal{N}, \forall i = 1, \dots, m : F_j(e_1, \dots, e_m, t_1, \dots, t_m), \forall j = 1, \dots, M\}$$

for some M , M is a non-negative integer, and F_j denotes a constraint and \bar{E} is the set of all event types. Note that, there can be M constraints and that the events are always found from left to right.

We introduce the following definition that we will need later.

Definition The events are always found from left to right, and the F_j functions define constraints on the events and times. The event in an episode definition that comes leftmost (that is, by the F definitions occurs first) is called a *starter event*. All other events are called *follower events*. If the episode definition is such that multiple events occur simultaneously at the leftmost point, then every such event can be seen as a starter event.

The definition of an episode effectively tells us that E is a set of m two dimensional points, subject to a set of functional constraints. We can therefore define a complementary episode, \bar{E} as the complement of E .

Given this definition of a multi-event episode and its complement, we can find all occurrences of the multi-event episode from a time series. Below, we give an outline of the algorithm:

```

ALGORITHM FIND_EPISODES_FROM_STOCK_TIME
_SERIES
Read time series  $S(t)$ ,  $t = 1, \dots, N$ 
Read episode definition
 $E = \{(e_1, t_1), \dots, (e_m, t_m), (e_i, t_i) \in \bar{E} \times \mathcal{N}$ 
 $\forall i = 1, \dots, m : F_j(e_1, \dots, e_m, t_1, \dots, t_m),$ 
 $\forall j = 1, \dots, M\}$  for some  $M, M$  non-negative int
episode_index = 1
complementary_episode_index = 1
for every event  $e_j, j = 1, \dots, m$  do
for every  $i = 1, \dots, N$  do
if  $S(t_i)$  is an event  $e_j$ 
push  $(e_j, t_j)$  to valid_times[j]
endif
endfor
endfor
for  $j = 1, \dots, m$  do
read valid_times[j]
endfor
find combinations of  $(e_i, t_i), i = 1, \dots, m$  such
that either all
 $F_j, j = 1, \dots, M$  or complements of  $F_j, j =$ 
 $1, \dots, M$  are satisfied
If  $F_j$  are satisfied, push that to
episode[episode_index] episode_index++;
endif
If complements of  $F_j$  are satisfied, push
that to
complementary_episode[complementary_episode
_index]
complementary_episode_index++;
endif

```

With this definition, the episode “closing price is lower than opening price, occurring at least twice in the same week” will be:

$$\{(C_i, t_i), (O_i, t_i), (C_j, t_j), (O_j, t_j) \text{ such that }, \\ C_i < O_i, C_j < O_j, \\ |i - j| \leq 7, i, j \text{ not part of a member already}\}. \quad (1)$$

Where C_i and O_i are the closing and opening prices, respectively on i^{th} day. Recall that events are always found from left to right. Thus, if a time series has one closing price less than opening price on day 1, another on day 4 and a third one on day 8 and never again, the first and the only episode present in the time series is the prices of day 1 and day 4. The pair of prices on day 4 and day 8 does not become an episode, as day 4's price is already a part of the first member.

Similarly, an episode “the stock opening price increases by X and does not decrease by Y or more anywhere in the same period” will be described as

$$\{(O_k, t_k), \quad k = i, i+1, \dots, j \text{ such that }, \\ O_j - O_i \geq X, \quad O_m - O_l < X, \forall m \neq j, \forall l \neq i, m > l \\ O_m - O_l < Y, \forall \{m, l\} \in (i, i+1, \dots, j), m < l, \\ \text{no } k \in \{i, i+1, \dots, j-1\} \\ \text{a part of a member already}\}. \quad (2)$$

Note the definition of k ; j can be a part of two members and exactly two members which share the boundary.

3 Framework

We showed in the earlier paper that given an episode definition E with just a single event, it is possible to encode a stock time series $S(t), t = 1, \dots, N$ as a binary string. Suppose the user defined episode is E , and it is the set of events and their time stamps, subject to $f(\cdot)$ and $g(\cdot)$ as defined by the user. Now, consider every episode E present in the time series, which is a set $\{(e_i, t_i)\}$. We encode the stock's time series the following way. We replace the t_i -th value of the time series by the letter '1', that is, $S(t_i)$ is assigned the value '1'. Thus, for every event member of an episode, the corresponding value of the time series gets encoded to '1'. Then, every member of the complementary set can be similarly coded by a letter '0'. Therefore, every stock data time series can now be defined as a string of 1's and 0's, the positions of the 1's and 0's being determined by the events which are member of the episode as defined by the user. Mathematically, we define the following mapping:

Suppose $S(t), t = 1, \dots, N$ is the stock time series, and E' is the set of all episodes in $S(t)$, each episode E defined as $\{(e_i, t_i) : f(e_i), g(t_i)\}$ found from $S(t)$. Define

$$\begin{aligned} h : \{1, \dots, N\} &\rightarrow \{0, 1\} \text{ such that,} \\ \forall \text{ event } e = (e_i, t_i) &\in E, \\ \forall E \in E', h(t_i) &= 1, \\ h(t) &= 0 \text{ otherwise.} \end{aligned} \quad (3)$$

Note that this encoding removes all dependence on amplitude and therefore, no normalization is needed.

This technique will not hold when episodes based on multiple events are being considered, as the episodes now incorporate multiple events. As the episode definition uses multiple events, we must encode every event.

Consider the episodes found using the algorithm above. As explained before, every episode definition can be seen as a set of points involving the events subject to the constraints.

We encode the time series as follows. Recall that an episode happens if all its constituent events happen satisfying the constraints. Since events are found from left to right, all starter events happen first. Suppose all starter events happen at a point t_1 , and then follower events occur at time instances t_2, t_3, \dots, t_m , satisfying all the constraints with no violation of constraints in between, then the episode is defined by the entire time series block from t_1 to t_m . We encode every event value of this block of the time series with a character a . If all starter events happen at a point, and then follower events happen with the complement of the constraints, we get a complementary episode, and we encode every event value of that block with a character b . If a time instant is such that it is not a part of an episode, neither of a complementary episode, and at least one starter event is not happening at that instant, we encode that point

with a character c . Then the entire time series can be encoded using only these three characters a , b and c , with every a defining one point of an episode as described by user, every b defining one point of a complementary episode, and one c defining a time instant when neither of this happens.

Mathematically, we define the following mapping:

Suppose $S(t), t = 1, \dots, N$ is the stock time series, E' is the set of all episodes in $S(t)$, E'' is the set of all complementary episodes in $S(t)$. Each episode E found from $S(t)$ is defined as $E = \{(e_1, t_1), \dots, (e_m, t_m), (e_i, t_i) \in \bar{E} \times \mathcal{N}, \forall i = 1, \dots, m : F_j(e_1, \dots, e_m, t_1, \dots, t_m), \forall j = 1, \dots, M\}$ for some M , M is a non-negative integer, F_j defines a constraint and \bar{E} is the set of all event types.

$$\begin{aligned} \text{Define } h : \{1, \dots, N\} &\rightarrow \{a, b, c\} \text{ such that,} \\ \forall \text{ event } e = (e_i, t_i) &\in E, \forall E \in E' \\ h(t_i) &= a, \\ \forall \text{ event } e = (e_i, t_i) &\in \bar{E}, \forall \bar{E} \in E'' \\ h(t_i) &= b, \\ h(t) &= c \text{ otherwise.} \end{aligned} \quad (4)$$

Note that this encoding removes all dependence on amplitude and therefore, no normalization is needed.

Once the encoding is done, we further shorten the string size by replacing any contiguous block of a 's in a single episode with a single a , and any contiguous block of b in a single complementary episode set with a single b . This is meaningful in the following way. Consider the episode "the stock opening price increases by X and does not decrease by Y or more anywhere in the same period". Thus, intuitively, the entire period of movement over which the price increased is one single episode instance, and the whole set should be seen as 'a'. The function h as defined above replaces each time stamp of that period by an a , and so replacing the entire block by a single a gives us the intuitive picture. Also note that, in case of two successive instances of episodes, we get two successive a 's. Note that we do not replace the contiguous block of c 's by one c , because to find similarity between two stocks, we need to know the length of sequences where no episodes took place.

Note that by the definition of episode "the stock opening price increases by X and does not decrease by Y or more anywhere in the same period", h may introduce more alphabet in the encoding string as some points belong to more than one member of the episode. However, because of the collapsing of successive a 's or b 's from one member, this does not matter. Below, we give an outline of the algorithm for the encoding.

ALGORITHM FIND_ENCODING_OF_STOCK_TIME_SERIES

Read episode description

$E = \{(e_1, t_1), \dots, (e_m, t_m),$
 $(e_i, t_i) \in \bar{E} \times \mathcal{N},$
 $\forall i = 1, \dots, m : F_j(e_1, \dots, e_m, t_1, \dots, t_m),$
 $\forall j = 1, \dots, M\}$
for some M, M non-negative integer

```

B =  $\phi$ 
for each  $i = 1, \dots, N$ ,
N = length of time series
do
  if  $t_i$  belongs to a member of episode
  description
    B = B.'a'
  else if  $t_i$  belongs to a member of
  complementary
  episode description
    B = B.'b'
  else B = B.'c'
enddo
for each episode do
  replace contiguous block of 'a's by a
  single 'a'
enddo
for each complementary episode do
  replace contiguous block of 'b's by a
  single 'b'
enddo

```

Therefore, for any two stocks, their time series representations can now be seen as two equivalent ternary strings $S1$ and $S2$. Now, similarity between these two stocks with respect to the episode definition can simply be seen as $d(S1, S2)$ where $d(.,.)$ is a string distance measure, like Hamming distance.

We use Levenshtein distance [15] and Jaro string distance [14] metric for measuring the distance between two strings. Both of these string distances are well known and extensively used for checking edit distances between two strings, or duplicate strings. If the distances are small enough, we conclude that the two strings $S1$ and $S2$ are close enough in their behavior, and thus, the stocks are dependent.

Note that a binary difference will be weaker. The strength of our method lies in the fact that Jaro/Levenshtein distances are edit distances and thus can measure similarity between two strings even if they are slightly time shifted. For example, if the first string is "abcabc", and the second string is "bcabca", then the binary distance between the two strings, which is the number of places where the two strings do not match exactly is 6. On the other hand, the Jaro distance between the two strings is 0.167 and the Levenshtein distance between the two strings is 2. Thus, both Levenshtein and Jaro tells us that the strings are indeed close enough, whereas binary distance will say that they are completely mismatched.

Here is one important observation. As the episodes and complementary episodes themselves are being encoded by a 's and b 's, these are the two characters which truly matter when it comes to finding the distances between the strings. The character c shows the presence of non-episodes, and thus qualifies primarily as a marker between occurrences of episodes and complementary episodes. Therefore, if two stocks encode as a string of c 's only, we define the distance between them to be ϕ , which implies that the stock similarity is undefined. The overall similarity function is defined below.

Let $S1$ and $S2$ be the two encoded strings obtained from two stocks. We define the distance

between $S1$ and $S2$ as follows.

$d(S1, S2) = \text{undefined}$ if both $S1$ and $S2$ are encoded entirely by c 's

$d(S1, S2) = D(S1, S2)$ otherwise, where $D(.,.)$ is one of Levenshtein string distance or Jaro string distance.

We take the following thresholds. For the Levenshtein metric, we use a cutoff value of 20. For Jaro distance, we use a cutoff value of 0.20. Values less than these are considered close. These thresholds were chosen by trial and error. Levenshtein distance is a measure of edit distance between two strings, that is, how many edits are needed to transform one string into another. The time series that we used has 252 values for each stock data. Thus, when encoded, we can expect a binary string whose length is of the same order. Therefore, a distance of 20 or less will imply that the two strings are less than 10% different. Jaro metric is somewhat stronger as it considers transpositions, so we relax the condition and take a threshold of 0.2, about 20. Note that classically Jaro metric implementation returns a value between $[0,1]$ for dissimilar to similar strings. We have used an implementation where 0 implies same strings.

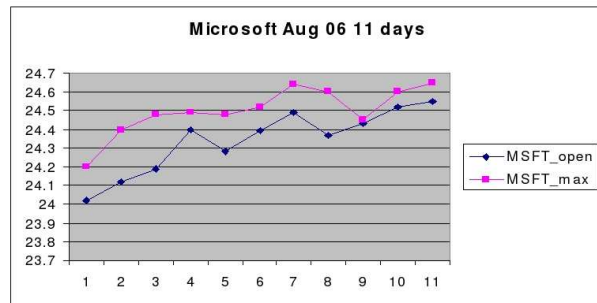


Figure 1: Stock time series snapshot for MSFT

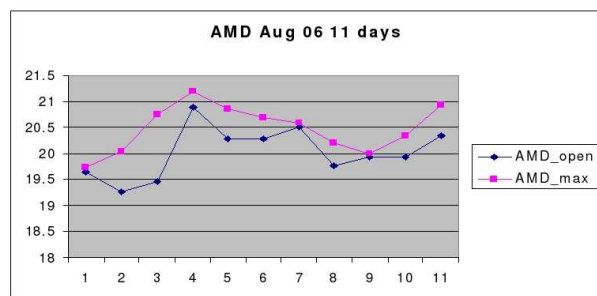


Figure 2: Stock time series snapshot for AMD

Figure 1, Figure 2 and Figure 3 show a snapshot of Microsoft, AMD, and IBM's stock prices respectively, for a period of 11 days in August 2006. Let us try to encode these stocks with respect to the episode definition "max price increases when opening price increases and falls when opening price falls."

From Figure 1, we see that MSFT max price increase is almost always proportional to opening price

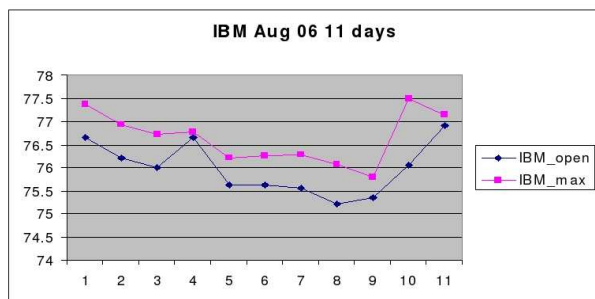


Figure 3: Stock time series snapshot for IBM

increase. AMD max price in Figure 2, does NOT follow it strictly. These two should be “far”. IBM max price in Figure 3, follows the opening price, but there are aberrations. So we do not know which way it would go.

Using our proposed scheme, the encoding would be:

MSFT: aaaaaaabaa (Note that we do not compress the a’s because each day gives an episode. Also the last a is dropped as it is the 12th day’s data).
 AMD: baaabbabaa
 IBM: aaaaababab

Suppose the episode definition is changed slightly to the following: “max price change follows the direction of opening price change for three days in succession”. Then, the encoding becomes:

MSFT: aaa/aaa/bbb/a (Note that bbb. Because we are now looking at three day series, those three days actually violated the constraints and thus the entire three day set is seen as a complementary episode).

When series of a-s and b-s within the same episode is compressed, we get aabc (the last a goes to c as we do not have three days’ data there, making it a void set).

AMD: bbb/bbb/bbb/a (Note those bbb’s. It compresses to: bbbc. Quite far from MSFT, in fact, the first two characters are different.

IBM: aaa/bbb/bbb/b (Note those bbb’s which compresses to: abbc. From MSFT, the distance is one character. From AMD, the distance is 1 character too, which says IBM, with this criterion, is about as far from MSFT as from AMD.

Of course, this is just 11 days’ data which makes no practical sense. This is only for demonstration of the idea.

4 Results

We can have different episodes comprising different combinations of the multiple events. Most interesting combinations would be the ones using numerical and non-numerical external attributes, like company policies, or a press release. It is easy to incorporate such data in the framework defined above. However, such special non-numerical data are usually found in commercial databases. Therefore, we will show the

usefulness of the strategy using two numerical values taken from Standard and Poor 500 data set.

We have tested our technique for all pairs of companies in Standard and Poor 500. However, because of space constraint, we present a representative set of results.

We use the following criterion: “maximum price follows the opening price movement for three days in succession”. That is, if the opening price moves up, maximum price also moves up, and if the opening price moves down, maximum price also moves down. This is a practical criterion used by investors for short term trading. Intuitively, this means that a stock is behaving steady for a significant period of time at a stretch, and it can open up trading opportunity. Three day steady match implies in terms of short trading, a trader can buy one day, and if it opens higher, hold and sell at a high maximum.

With this criterion, similarity can be intuitively explained as follows. A company *A* and another company *B* are similar implies their three day following patterns are similar. So if short trading on *A* is profitable on a day *d*, short trading on *B* may be profitable sometime soon.

We have computed the results for all companies using the entire Standard and Poor 500 data. However, because of space constraints, we present results for a few pairs of companies. To illustrate the strength of the proposed method, we give the distances found by the method, and also the correlation coefficient (which is the standard indicator used in pairs trading strategy), and show how the proposed method can indeed predict a better similarity. Together with that, for an intuitive visual justification, we also show the trend of the companies’ opening and maximum prices, but as it is difficult to follow the complete plot for all 252 days in a compressed scale, we explain the visual similarities using smaller snap shots.

First, we take the example of Alcoa, Apple and Ambac Financial group in Figure 4, Figure 5 and Figure 6 respectively. The distances for these three are (given in X/Y/Z format, where X is Levenshtein, Y is Jaro and Z is correlation coefficient) shown in Table 1:

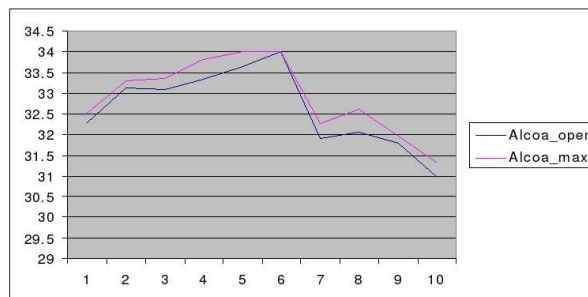


Figure 4: Stock time series snapshot for Alcoa multiple events

Alcoa and Apple form an interesting pair. Looking at the graphs, we see that for Alcoa, the maximum price follows the opening price quite faithfully, while Apple’s maximum price does not. Thus, for

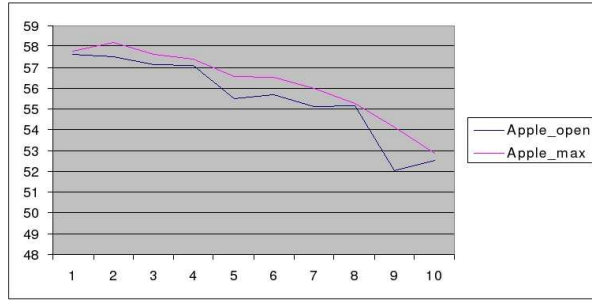


Figure 5: Stock time series snapshot for Apple multiple events

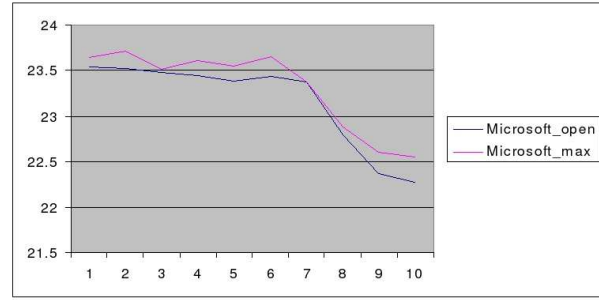


Figure 7: Stock time series snapshot for Microsoft multiple events

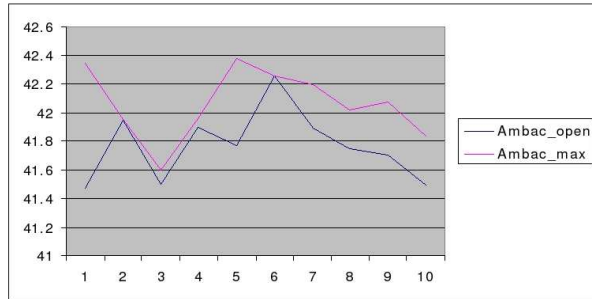


Figure 6: Stock time series snapshot for Ambac multiple events

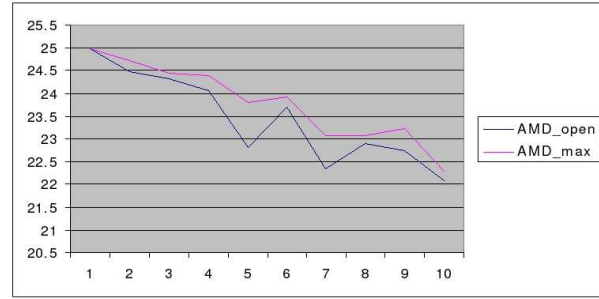


Figure 8: Stock time series snapshot for AMD multiple events

Table 1: Values of Levenshtein, Jaro distance and correlation coefficient in an $X/Y/Z$

	Alcoa	Apple	Ambac
Alcoa		27/0.12/0.80	19/0.09/0.63
Apple			26/0.15/0.65

an investor using our criterion, these two are indeed far apart. Our method detects its separation, while the correlation coefficient shows a very high degree of correlation. Thus, for pairs trading, these two will form a good pair. However, an investor who wants to use our criterion as a safeguard will not get the right cue from standard pairs trading, while the proposed method can indeed provide that.

Next we take Microsoft, AMD and Brunswick Corporation. Brunswick is a consumer discretionary group. Their graphs are given in Figure 7, Figure 8 and Figure 9, respectively, and the distances are shown in Table 2.

AMD's three day pattern shows random behavior. See AMD's graph in Figure 8. It mostly follows the opening price pattern, but at times deviates from it strongly, which Microsoft never does. Thus, they are far apart. The correlation coefficient in Table 2 also seconds that observation. Brunswick and Microsoft are closer and this observation is also strengthened by the correlation coefficient. Here, pairs trading obser-

vations and the proposed method's observations are similar.

Next we take Microsoft vs Biomet Inc, which is a biomedical company. The distance between these two companies are 16/0.07/0.8053. Note the movements of the two companies, Biomet (shown in Figure 10) and Microsoft (shown in Figure 7). The maximum price follows the opening price and then the rate of fall changes, it may also start rising. Thus, a three day steady window with the criterion can tell the investor that maximum price is about to rise. Thus, these two companies suggest a good similarity. This pair is a good candidate for pairs trading strategy too, with a high correlation.

Next, we compare Starbucks, a consumer discretionary and Perkin Elmer, which is in health care, in Figure 11 and Figure 12 respectively. The distance are 18/0.10/ - 0.43. Here, the companies are not similar if one looks at their general behavior. The correlation coefficient is negative, but the value is not too strong for ensuring a definite negative correlation. However, with our criterion, let us look at the graphs. It shows that for Starbucks, the maximum value moves exactly with the opening price, and Perkin Elmer does the same. Thus, the three day pattern for both these companies is exactly similar, making them similar with respect to our criterion. Interestingly, the companies are from different areas and thus not very easy to find as a prospective pair.

Next, we compare Starbucks and Sanmina-SCI Corp., which is in electronic manufacturing services.

Table 2: Values of Levenshtein, Jaro distance and correlation coefficient in an $X/Y/Z$ form

	AMD	Burnswick Corporation	Microsoft
AMD		25/0.14/ - 0.33	29/0.19/ - 0.48
Burnswick Corporation			19/0.09/0.53

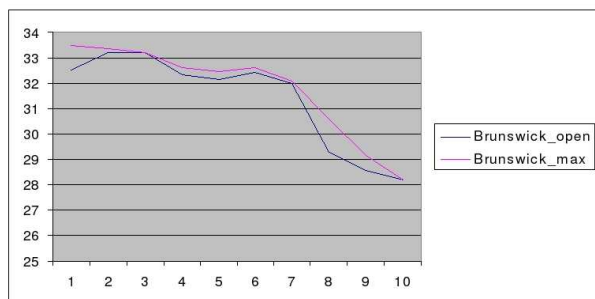


Figure 9: Stock time series snapshot for Burnswick Multiple events

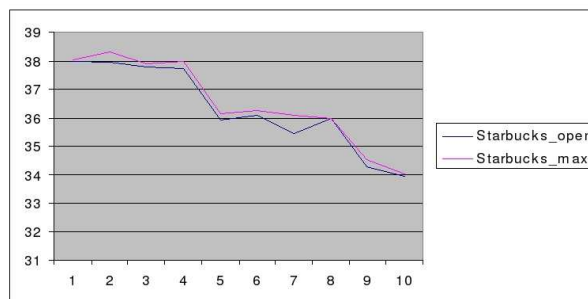


Figure 11: Stock time series snapshot for Starbucks multiple events

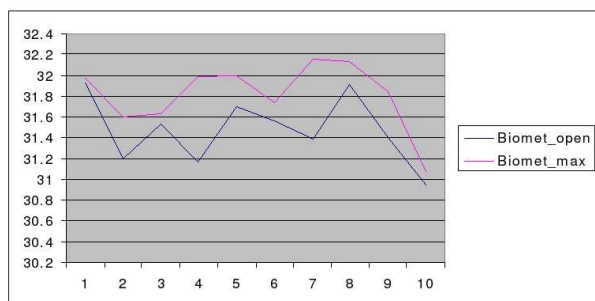


Figure 10: Stock time series snapshot for Biomet multiple events

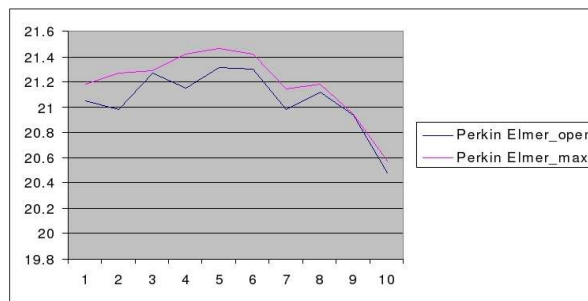


Figure 12: Stock time series snapshot for Perkin Elmer Multiple events

The distance here are 24/0.18/0.66. Now observe the following from the graph for Sanmina-SCI in Figure 13. Here, the maximum value does not follow the opening price exactly. Right at the beginning, the maximum has increased even though the opening has fallen and later the maximum has fallen even though the opening has increased. Thus, Sanmina and Starbucks (Figure 11), are indeed quite far as far as their three day maximum vs opening price movement patterns are concerned. The distances reflect this, though the correlation coefficient is not very low and thus can not point to this dissimilarity.

Finally, we compare NI (Nisource Inc, utilities) and KG (King Pharmaceuticals, Health care), the distance are: 18/0.12/0.26.

The opening price pattern themselves are quite different between these two companies (see Figure 15 and Figure 14 respectively), and therefore their correlation coefficient is low. However, the relative pattern between the maximum and opening price (which is what we are looking at) are similar. Except at

the beginning, the maximum price follows the opening price steadily. This will tell an investor speculating in King pharmaceutical to consider Nisource Inc. also at about the same time. Note the fact that these companies are from different sectors and thus such dependences may not be obvious to an investor.

5 Conclusions

Finding similarities between stocks can be very useful for investors. Instead of trying to predict a stock's movement from overall market trend, dependencies between two stocks can give the investors unforeseen insights, especially if the similarities occur between stocks across very different areas which may not be obvious and easy to find. This is especially useful if the similarities can be detected for any user defined criterion, giving the investor a framework to find such inter-relationships for any relevant criterion. We showed that many such user defined criterion can be described mathematically as a set of events with

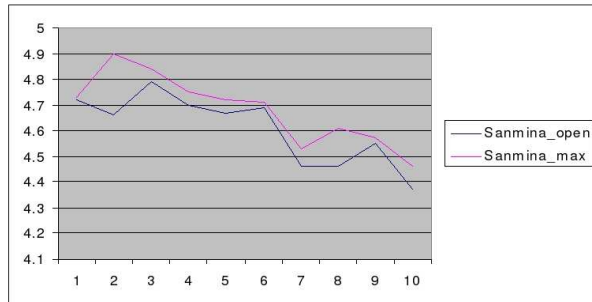


Figure 13: Stock time series snapshot for Sanmina multiple events

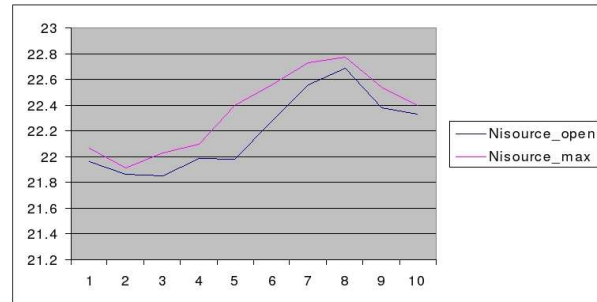


Figure 15: Stock time series snapshot for Nisource multiple events

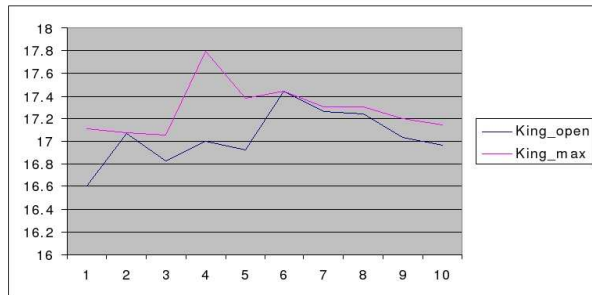


Figure 14: Stock time series snapshot for King multiple events

restrictions on values and time, which can be seen as an episode. It is possible to define episodes using different kind of events which are not associated with the time series values, for example “whenever the company splits its share”, or using events across different areas, e.g., “when the company declares a dividend and the prices goes up”. Given such an episode, it is possible to encode a stock’s time series as a ternary string and then, similarity between two stocks’ movements can be efficiently found by applying string distance metrics between the stocks’ time series’ ternary string representations. We have used this framework on Standard and Poor 500 data, with a practical multi-event episode. The results detected the similarities quite efficiently, and difficult to find dependencies between companies from totally unrelated areas often appeared. This can be extremely useful for investors. It was also shown that the proposed method works well compared to the well known pairs trading strategy. We believe that the current technique will prove quite useful in practice. To the best of our knowledge, this is the first attempt to find similarity between stock trends based on user defined multiple event criteria.

References

- [1] http://en.wikipedia.org/wiki/algorithmic_trading.
- [2] <http://kumo.swcp.com/stocks/>.
- [3] <http://www.incrediblecharts.com>.
- [4] D. R. Aronson. *Evidence Based Technical Analysis: Applying the Scientific Method and Statistical Inference to Trading Signal*. Wiley Trading, 2006.
- [5] J. A. Bollinger. *Bollinger on Bollinger Bands*. McGraw Hill, 2001.
- [6] B. Bollobas, D. G. Gautam Das, and H. Mannila. Time-Series Similarity Problems and Well-Separated Geometric Sets. In *13th ACM Symp. Computational Geometry*, pages 454–456, 1997.
- [7] P. J. Brockwell and R. A. Davis. *Intro. to Time Series and Forecasting*, volume 2nd edition. Springer-Verlag, NY, 2002.
- [8] C. Coronello, M. Tumminello, F. Lillo, S. Micciche, and R. N. Mantenga. Sector identification in a set of stock return time series traded at the london stock exchange. *Acta Physica Polonica B*, 36(9):2653–2679, 2005.
- [9] G. Das, D. Gunopulos, and H. Mannila. Finding Similar Time Series. *Principles of Data Mining and Knowledge Discovery*, pages 88–100, 1997.
- [10] A. Dattasharma and P. K. Tripathi. Identifying Stock Similarity Based on Episode Distances. In *IEEE Int’l Workshop on Data Mining and Artificial Intelligence (DMAI 2008)*, 2008.
- [11] A. Dattasharma and P. K. Tripathi. Practical Inter-stock Dependency Indicators using Time Series and Derivatives. In *The 6th ACS/IEEE Int’l Conf. on Comp. Sys. and Appln. (AICCSA-08)*, 2008.
- [12] M. Gavrilo, D. Anguelov, P. Indyk, and R. Motwani. Mining the Stock Market: Which Measure is Best? In *6th ACM Int’l Conf. on Knowledge Discovery and Data Mining*, pages 487–496, 2000.
- [13] P. Gopikrishnan, B. Rosenow, V. Pleroul, and E. Stanley. Quantifying and interpreting collective behavior in financial markets. *Physical Review E*, 64(3), 2001.

- [14] M. A. Jaro. Advances in Record Linking Methodology as Applied to the 1985 Census of Tampa Florida. *Journal of the American Statistical Society*, 84(406):414–420, 1989).
- [15] V. I. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. *Soviet Physics Doklady*, 10:707–710, 1966.
- [16] H. Lu, J. Han, and L. Feng. Stock Movement Prediction and N-dimensional Inter Transaction Association Rules. In *ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, pages 12.1–12.7, 1998.
- [17] H. Mannila and a. A. I. V. Hannu Toivonen. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 3(1):259–289, 1997.
- [18] R. J. Miller and Y. Y. Y. Association Rules over Interval Data. In *ACM SIGMOD Conf. on Mgmt. of Data*, pages 452–461, Tucson, Arizona, USA, May 1997.
- [19] J. Rodgers and W. Nicewander. Thirteen ways to look at the correlation coefficient. *The American Statistician*, 42:59–66, 1988.
- [20] R. Srikant and R. A. R. Mining Quantitative Association Rules in Large Relational Tables. In *ACM SIGMOD Conf. on Mgmt. of Data*, pages 1–12, Montreal, Canada, June 1996.
- [21] G. Vidyamurthy. *Pair Trading: Quantitative Methods and Analysis*. Wiley Finance, 2004.
- [22] H. White. Economic Prediction Using Neural Networks: The Case of IBM Daily Stock Returns. In *2nd Annual IEEE Conf. on Neural Networks II*, volume 2, pages 451–458, 1998.
- [23] J. W. Wilder. *New Concepts in Technical Trading Systems*. Trend Research, 1978.

Evaluation of Malware clustering based on its dynamic behaviour

Ibai Gurrutxaga, Olatz Arbelaiz, Jesús M^a Pérez, Javier Muguerza, José I. Martín, Iñigo Perona

Dept. of Computer Architecture and Technology
University of the Basque Country
M. Lardizabal, 1, 20018 Donostia, Spain

{i.gurrutxaga, olatz.arbelaiz, txus.perez, j.muguerza, j.martin, inigo.perona}@ehu.es

Abstract

Malware detection is an important problem today. New malware appears every day and in order to be able to detect it, it is important to recognize families of existing malware. Data mining techniques will be very helpful in this context; concretely unsupervised learning methods will be adequate. This work presents a comparison of the behaviour of two representations for malware executables, a set of twelve distances for comparing them, and three variants of the hierarchical agglomerative clustering algorithm when used to capture the structure of different malware families and subfamilies. We propose a way the comparison can be done in an unsupervised learning environment. There are different conclusions we can draw from the whole work. Concerning to algorithms, the best option is average-linkage; this option seems to capture better the structure represented by the distance. The evaluation of the distances is more complex but some of them can be discarded because they behave clearly worse than the rest of the distances, and the group of distances behaving the best can be identified; the computational cost analysis can help when selecting the most convenient one.

Keywords: malware, hierarchical clustering, representation based on dynamic behaviour.

1 Introduction

Many of the most visible and serious problems facing the Internet today are related to malicious software and tools.

Malicious code, commonly called malware is software designed to infiltrate or damage a computer system without the owner's informed consent. It is considered malware based on the perceived intent of the creator rather than any particular features. Malware is generally the source of spam, phishing, denial of service attacks, botnets, and worms. New malware appears every day. In recent years, the first real viruses and worms for MacOS as well as Trojans for the J2ME mobile platform appeared. These last were designed to steal money from mobile user accounts.

The trends in malware evolution continue: Trojans are far more numerous than Worms, and the number of new malicious programs designed to inflict financial damage

have increased. As a consequence, the effect of malware has great economical cost.

Modern malware is very complex; many variants of the same virus with different abilities appear every day which makes the problem more difficult. Understanding this process and how attackers use the backdoors, key loggers, password stealers and other malware functions is becoming an increasingly difficult and important problem.

At this stage, the classification of new malware by human analysis, where memorization, looking up description libraries or searching sample collections is required, is not effective (Lee, and Mody 2006); it is too time consuming and subjective. As a consequence, automated and robust approaches to understanding malware are required in order to successfully face the problem. These processes will consist on representing and storing the knowledge in an adequate manner and learning from the stored information.

The representation used by antivirus focuses primarily on content-based signatures, which are inherently susceptible to inaccuracies due to polymorphic and metamorphic techniques. Their detection is based on static analysis, that is to say they represent the malware based on the structural information of a file. This kind of analysis fails to detect inter-component/system interaction information and would difficultly manage to detect malware created by collaboratively working packages (Bailey, Oberheide, Mao, Jahanian, and Nazario 2007). Code and data obfuscation also poses considerable challenges to static analysis.

The mentioned problems suggest that the use of data from runtime analysis, the collection of system calls (e.g., files written, processes created), is more invariant and directly useful than abstract code sequences. The dynamic behaviour can be directly used in assessing the potential damage malware can cause and it will enable detection and classification of new threats.

Clustering malware based on the dynamic analysis approach based on the execution of malware in controlled real environments can be very helpful for anti-malware professionals to address the limitations of existing automated classification and analysis tools and better analyze the new malware appearing every day. New malware could be assigned to existing clusters or malware families, so variations of known malware will be easily recognized. Clustering malware is not an evident task; many decisions have to be made:

1. Data needs to be collected, represented and stored so that the smallest amount of information is lost.

2. Distances for comparing the stored data need to be selected.
3. Finally, a clustering algorithm needs to be selected to be used with the chosen data representation and distance.

The idea of the use of the dynamic behaviour has been applied in other security areas. For example, papers applying this idea can be found in Gao, Reiter, and Son 2005, Yeung, and Yuxin 2003, Brugger 2004, Christodorescu, Jha, and Kruegel 2007; or in commercial systems such as Norman (Norman Solutions 2003), and CWSandbox (Willems, and Holz 2007). However, few works have been done based on the dynamic behaviour of malware. For example, Bailey, Oberheide, and Mao 2007 cluster malware and evaluate the obtained results based on 5 antivirus' output, and in Lee, and Mody 2006 malware is clustered based on an opaque representation, edit distance and k-medoid clustering algorithm, but not comparison of different strategies is presented.

In this sense, no analysis has been done on which are the best representation, distance functions and clustering algorithms to use.

The aim of this work is to make the analysis of the behaviour of two representations, a wide set of distances and some clustering algorithms when used to clustering malware based on its dynamic behaviour. This is not an easy task in an unsupervised context but we have designed an adequate experimentation and an evaluation methodology that can help doing this work properly.

As we have mentioned, lots of new malware are generated every day. In this context, the detection of malware needs to be efficient. This has two main consequences: on the one hand the used representations will be as simple as possible as far as they do not lose too much information. On the other hand, when evaluating the distances and algorithms, added to the achieved accuracy, the computational cost will also be an important feature to be taken into account.

The paper proceeds describing in Section 2 the different options for clustering malware we have evaluated, how data has been processed, the used representations, distances and algorithms. Section 3 is devoted to describe the details about the experimental methodology for evaluating distances on the one hand, and algorithms on the other one. In Section 4 we present an analysis of the experimental results where comparison of distances and algorithms are shown, together with an analysis of the computational cost. Finally Section 5 is devoted to show the conclusions and further work.

2 Malware analysis tools

As we mentioned in the introduction, the process of automatically clustering malware and later detecting new attacks is divided in three main steps: data needs to be collected, represented and stored; distances for comparing the stored data need to be selected, and finally, a clustering algorithm needs to be chosen to be used with the selected data representation and distance.

This section is devoted to describe the options that will be evaluated for each one of the processes in this work.

2.1 Data processing

In any data mining process, the starting point is the data collection, representation and storage process. This work is also part of the data mining process and, as a consequence, to be able to cluster malware based on dynamic analysis of code, it is compulsory to get some data.

The data used in this work has been provided by S21sec lab a leading security company in our environment. It has been generated in a real environment; no virtual machines have been used in the process. Malware code has been executed in monitorized machines; Pentium IV of 3.20 GHz with 1 GB RAM and Windows XP SP2 operating system. The information we have about the kind of malware we are working with is the output of Kaspersky antivirus. Some examples of the treated malware (based on the antivirus) are: Backdoor.Win32.Rbot, Trojan.Win32.Agent, Net-Worm.Win32.Mytoob, SpamTool.Win32.Agent, Email-Worm.Win32.Mydoom ...

The code of the malware has been executed during one minute in a controlled environment and information about the system calls generated during this period has been collected. The security experts in S21sec company have decided to distribute the detected system calls in 45 groups (events), each one with its corresponding code; Table 1 shows an example of event codification.

Code	Event
1	File create
2	File open
...	...
44	Reg. Key Set Val.
45	Network Connect

Table 1: Example of event codification

The information stored for each of the executions is a sequence of events, that is to say, a list of codes, corresponding to the events detected during the minute the monitorization lasted. An example could be: <2, 44, 23, 23, 11, 16, 2, 16, 16, 16>.

This representation has been chosen because we find it adequate to represent the dynamic behaviour of the malware. It allows storing information related to the kind of the generated system calls, and the order the actions took place. This is a double edged sword; on the one hand, some meaningful information is stored, but, on the other hand, when using this data for clustering malware, sequence comparison methods will be required to work with it, which are in general very time consuming.

Simpler representations (vector representations) of the data can also be used; these methods will have some disadvantages, they will in general miss information, the information about the order of the events disappears, but on the contrary, the comparison strategies will be faster.

In this sense, in order to evaluate the accuracy/computational cost trade-off we have also worked with a projection vector of the stored data where the projection of the information stored as event sequences is done by representing the frequency of the different events (each one of the 45). This option has been called Count. As an example, the vector projection

obtained for the previous example (<2, 44, 23, 23, 11, 16, 2, 16, 16, 16>) is represented in Table 2.

Event	Frequency
1	0
2	2
...	...
11	1
...	...
16	4
...	...
23	2
...	...
44	1
45	0

Table 2: Example of the vector projection called Count

2.2 Tools for comparing the stored data

The aim of this work is the evaluation of representations, distances and clustering methods and this subsection is devoted to describe the distances used for comparing examples. The kind of distances that can be used depends on the used representation. To work with sequences, different distances have been evaluated, whereas in the case of vectorized data, Count, the Euclidean distance has been used as distance. Added to those options, some special cases have been studied in order to use them as baseline for the comparison.

2.2.1 Distance and similarity functions for comparing sequences

When the chosen representation has been event sequences, the comparison could not be done using the same distances used to compare vector representations. Thus, distances for sequence comparisons need to be selected. The experimentation has been done with four well known distances and some variants (normalized options). The used distances are described in the following paragraphs.

1. Edit Distance (ED): the edit distance between two strings of characters is the minimum number of edit operations on individual characters needed to transform one string to the other (Gusfield 1997). The permitted edit operations are the insertion, deletion and substitution (replacement) of a character.
2. Longest Common Substring (LCSg): the length of the longest common substring existing between two strings of characters. A substring common to two strings must consist of contiguous characters in both strings.
3. Longest Common Subsequence (LCSq): the length of the longest common subsequence existing between two strings of characters. A subsequence common to two strings needs not to consist of contiguous characters in any of the strings but the characters must be in the same order.

4. Normalized Compressed Distance (NCD): distance based on compression techniques to compare two character sequences (Li, Chen, Li, Ma, and Vitanyi 2004, Wehner 2005). NCD is a Kolmogorov complexity based metric. Since in practice Kolmogorov complexity cannot be computed directly, it is approximated with real compression algorithms. $C(x)$ is defined as the length of the string obtained compressing x , and $C(xy)$ as the length of the string obtained compressing the concatenation of x and y . The Normalized Compression distance is defined as follows:

$$NCD(x,y) = \{C(xy) - \min(C(x), C(y))\} / \max(C(x), C(y)).$$

2.2.2 Normalized option

Some of the options such as ED and Count will be very dependent on the sequence length. In fact we think that the use of these distances will be very similar to the comparison of sequences by their length. As a consequence, on the one hand we have added the normalized options: EDN, LCSgN and LCSqN, dividing the corresponding value with the length of the shortest sequence in the comparison, and EDX, LCSgX and LCSqX dividing the corresponding value with the length of the longest sequence in the comparison.

The vector projection has also been normalized in two different ways. In CountL option the frequency of each variable is normalized dividing it with the length of the sequence (relative frequency). In order to obtain CountM normalization, for each variable i the mean, m_i , and the standard deviation, s_i , in the whole database are computed. Each value x_{ij} is replaced by $(x_{ij} - m_i) / s_i$.

Notice that LCS functions are similarity functions, and, as a consequence, to be used in the clustering algorithms, they need to be converted to distances. The conversion has been done after the normalization process. The similarity value S has been converted into $D=1-S$ distance value.

2.2.3 Special case

Added to the mentioned options, two simple representations have been used to confirm some suspicions, on the one hand, and as baseline, on the other one.

1. Length: each sequence is represented by its length and the used distance is the Euclidean one. This option has been used in order to compare results with the not normalized options of ED and Count and confirm that their behaviour is similar.
2. Random: a random distance value has been generated for each compared pair of sequences. This will be helpful to show whether the rest of the distances supply any information or not.

2.3 Selected algorithms

The aim of this work is to evaluate how different strategies work when clustering malware. We could say malware has a hierarchical structure (families, subfamilies, variants, types...). Taking into account this structure, we have selected hierarchical clustering algorithms for the experimentation. The results that can

be obtained from them go further than just some malware families; they can provide a hierarchy of the malware.

This work compares three hierarchical clustering algorithms (Jain, and Dubes 1988, Sneath, and Sokal 1973, Mirkin 2005), concretely three options of the agglomerative clustering algorithm differentiated by the method used for calculating distances between clusters.

In agglomerative hierarchical clustering, initially all instances are located in individual clusters and in each iteration the nearest clusters are merged. The process finishes when only one cluster containing all the instances is left. In order to compute the distance between two clusters we will use three of the most used definitions.

1. Single-linkage (or nearest neighbour): the distance between two clusters is computed as the distance between the two closest elements in the two clusters.
2. Complete-linkage (or furthest neighbour): the distance between two clusters is computed as the distance between the two furthest elements in the two clusters.
3. Average-linkage (or group average): the distance between two clusters is computed as the average distance between all the pairs of elements each one from a different cluster.

The output of these algorithms is a cluster hierarchy which is suitable to be represented in a graphical way. The most usual procedure is to draw them as a tree diagram called dendrogram. The root node of this tree contains the topmost partition (the cluster containing all the data points in the dataset) while the leaf nodes contain the partition at the lower level of the hierarchy where all the data points are usually clustered in singleton clusters.

3 Experimental methodology

The security company S21sec provided us with a database that contains 500 malware executions. As a consequence, the data we will use for this work are 500 event sequences, each one belonging to a different malware execution. Since the one minute execution is not always resulting in the same amount of relevant system calls, the sequences contain different number of events;. This length goes from 1 to 12,944 where the mean value is 1,008 and the standard deviation 2,173. The dataset has been randomly divided in 5 parts of 100 sequences and the same experimentation has been repeated with each one of them. This methodology allows proving that the obtained results are not biased to a particular sample.

One of the out comings of this work will be to propose an evaluation methodology that includes a meta-learning step for such an environment. When designing the experimentation, we have to take into account two aspects. On the one hand, we want to evaluate to what extent different distances are able to mark differences between malware examples belonging to different families and, on the other hand, we want to evaluate how the structure represented by these distances is captured by different clustering algorithms. The methodology has been designed having into account that the work has to be done in an unsupervised learning environment: that is to say, the relation existing between the 500 malware examples is unknown.

3.1 Methodology for evaluating distances

The first step has been to build a distance matrix for the malware examples to be clustered with each one of the 12 distances (Count, CountL, CountM, ED, EDN, EDX, LCSgN, LCSgX, LCSqN, LCSqX, Length, NCD and Random) to be compared.

In order to compare the 12 distance matrices, the correlation between them has been calculated. This will help to evaluate the strength and direction of the relationship existing between the different distances. The best known correlation coefficient is the Pearson product-moment correlation coefficient (Jain, and Dubes 1988), which is obtained by dividing the covariance of the two variables by the product of their standard deviations. This information has been used to try to evaluate differences and relationships between the evaluated distances.

3.2 Methodology for evaluating algorithms

The second objective is to evaluate how the three options for hierarchical clustering capture the structure represented by the distances.

The first step in this part has been to use the distance matrices to build a dendrogram with each one of the three algorithms we mean to evaluate, that is to say, they will be used to cluster malware based on different distances and algorithms. In order to evaluate to what extent the algorithms are able to capture the structure represented by each one of the distance matrices, the correlation between the distance matrix and the cophenetic matrix of the corresponding dendrogram has been calculated. The cophenetic matrix is the matrix of values $[dc(xi, xj)]$ where $dc(xi, xj)$ is the cophenetic proximity measure: the level in the dendrogram at which objects xi and xj are first included in the same cluster. The closer the cophenetic matrix and the given distance matrix are, the better the hierarchy fits the data (Jain, and Dubes 1988, Halkidi, Batistakis, and Vazirgiannis 2001).

Another important clue when evaluating distances is the computational cost. As we mentioned in the introduction, new malware appears every day and this malware will also need to be classed into a family. As a consequence a time consuming system would not be adequate.

4 Results

We have executed many runs with different data of, 100, 200, 300, 400 and 500 sequences. The general conclusions have been similar in all of them and we will therefore show results for the first run executed with a sample of 100 sequences. In the section devoted to computational cost analysis execution times for incremental samples of 100, 200, 300, 400 and 500 are also shown.

4.1 Evaluation of distances

The values in Table 3 represent the correlation values between all the possible pairs of distance matrices (12×12). Just the lower diagonal matrix is shown because the complete matrix would be symmetrical. A pair wise comparison has been done and the obtained values could be used to decide how similar or different the structure captured by different pairs of distances is.

Count	1.00															
CountL	0.47	1.00														
CountM	0.59	0.39	1.00													
ED	0.99	0.44	0.60	1.00												
EDN	0.40	0.29	0.20	0.41	1.00											
EDX	0.43	0.69	0.41	0.43	0.23	1.00										
LCSgN	0.18	0.02	0.30	0.17	-0.31	-0.06	1.00									
LCSgX	0.34	0.49	0.40	0.34	0.16	0.75	0.30	1.00								
LCSqN	0.21	0.34	0.41	0.17	-0.12	0.12	0.56	0.14	1.00							
LCSqX	0.44	0.67	0.40	0.43	0.23	0.98	-0.12	0.72	0.04	1.00						
Length	0.97	0.40	0.55	0.99	0.43	0.42	0.13	0.33	0.09	0.43	1.00					
NCD	0.50	0.57	0.57	0.52	0.29	0.75	0.07	0.76	0.02	0.75	0.50	1.00				
Random	0.00	0.02	-0.02	0.00	-0.01	0.01	0.00	0.01	-0.01	0.01	0.00	0.01	0.00	0.01	1.00	
	Count	CountL	CountM	ED	EDN	EDX	LCSgN	LCSgX	LCSqN	LCSqX	Length	NCD	Random			

Table 3: Correlation values for all possible pair wise comparisons of the 12 distance matrices evaluated in the experimentation

The first point to underline is that, as it could be expected, the values in the matrix show that there is some similarity in the structures captured by all the distances, whereas this correlation does not exist in the case of Random. The values in the last row of the table, where Random option is compared to the rest, are nearly 0, whereas for the rest of the matrix the values are greater. This means that all the proposed distances make sense. That is to say they are able to capture similar characteristics of the structure of the data.

The rest of the values could be analyzed one by one and decide to what extent different pairs of distances are correlated. On the one hand, the distances whose distance matrices have greater correlation values would be the ones having more similar behaviour, whereas negative values would mean inversely correlated distances.

Although for the distance comparison we are doing in this work, this last option does not make much sense: we can observe in Table 3 that, apart from the ones appearing in the row corresponding to Random, there are some more negative values. In general, these values appear when the distances compared have been normalized with the length of the shortest compared sequence. As a consequence, even at this stage we could say that there is some unexpected behaviour with this kind of normalization.

In order to obtain results of a greater semantic level compared to those that could be obtained from just analyzing the pair wise comparison values, a dendrogram has been built from the correlation matrix in Table 3. This can not be done directly because correlation values measure similarity and distances are required to build dendrograms. In order to convert correlation values to distance values the following function has been used:

$$dist(i, j) = \begin{cases} 1 & \text{if } corr(i, j) < 1 \\ 1 - corr(i, j) & \text{otherwise} \end{cases}$$

This process has been repeated with the three options for the agglomerative clustering algorithm that we will study in this work and the structure captured by the three

algorithms is the same. As an example, next figure (Figure 1) shows the structure of the dendrogram obtained with average linkage.

The first conclusion we can obtain from the dendrogram is the same we obtained from Table 3, that is to say, that any of the distances have more similarity between them than the similarity they have with Random. As a consequence we could say that all the distances we are evaluating in this work are able to capture some kind of structure in the data.

For the rest of the distances, the dendrogram can help to identify families. On the one hand, as we suspected, the behaviour of distances ED and Count is very similar to the behaviour of Length. That is to say they are too dependent on the length of the sequence and they do not add any improvement to Length which is considerably simpler to obtain. The CountM option is not so similar to Length but we could also say that it adds little more. We consider all these distances useless due to the little and meaningless information they use.

The dendrogram shows that the structure captured by EDX and LCSqX distances is practically the same and even if there are more differences, the EDX, LCSqX, LCSgX, NCD and CountL distances form a compact group and seem to be the more promising distances.

Finally, it seems that the distances normalised using the length of the shortest sequence are not able to capture the structure of the data very well, they are not similar to any of the other distances and either between them. This confirms somehow the intuition we had when analysing results in Table 3.

At this stage we have evaluated the behaviour of different distances. Next step will be to study which algorithm has greater ability to extract the structure represented by each one of the distance matrices.

4.2 Evaluation of algorithms

The basic idea used to design this evaluation process has been the one published in Jain, and Dubes 1988: the closer the cophenetic matrix and the given distance matrix the better the hierarchy fits the data.

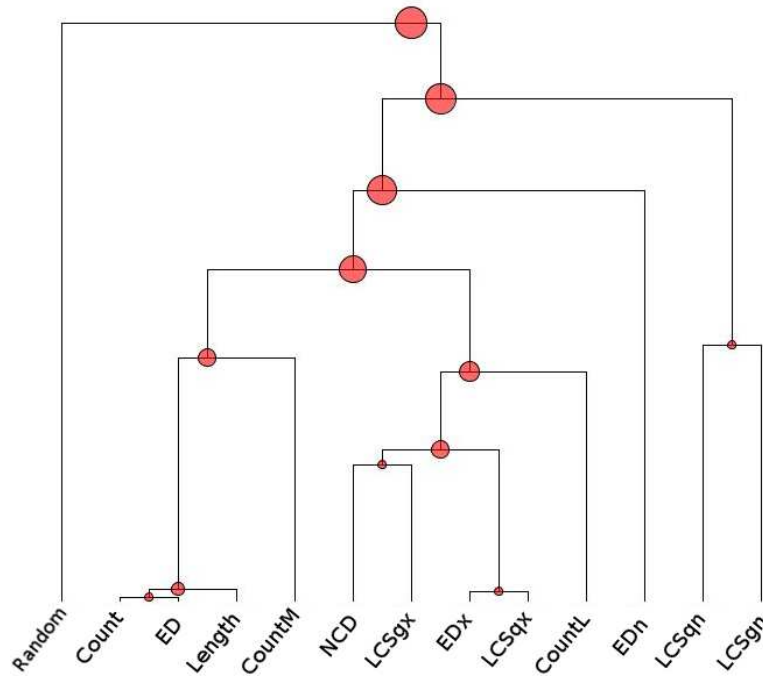


Figure 1: Dendrogram obtained from the correlation matrix converted into distance matrix, using the agglomerative clustering algorithm with average linkage

The evaluation procedure has been divided in three steps for each of the evaluated algorithms:

1. Based on the corresponding distance matrix, a dendrogram has been built for each one of the 12 distances.
2. The cophenetic matrix of each dendrogram has been calculated.
3. The correlation between the distance matrix and the corresponding dendrogram (the cophenetic matrix obtained from the dendrogram) has been calculated.

Table 4 helps to identify how well the different options for agglomerative hierarchical clustering behave with each one of the distances.

	Average	Single	Complete
Count	0.99	0.97	0.99
CountL	0.88	0.62	0.76
CountM	0.98	0.97	0.95
ED	0.99	0.98	0.99
EDN	0.34	0.33	0.34
EDX	0.95	0.82	0.90
LCSgN	0.50	0.34	0.33
LCSgX	0.97	0.89	0.90
LCSqN	0.48	0.00	0.36
LCSqX	0.95	0.81	0.91
Length	0.97	0.95	0.98
NCD	0.92	0.83	0.81
Random	0.25	0.07	0.22
Mean	0.83	0.71	0.77

Table 4: Correlation values for the distance matrices and the corresponding cophenetic matrices obtained based on three different clustering algorithms

There is no doubt independently of the used distance: the best algorithm is average-linkage. It is the algorithm achieving the greatest correlation value, not only in average but for each one of the evaluated distances.

This study also shows that the Random distance is different to the others because none of the compared algorithms can obtain a representative dendrogram.

Apart from the evaluation of different algorithms, the information in Table 4 helps to make some discrimination between distances. The values in the table show that hierarchical agglomerative algorithms can hardly represent the structure of the distances normalized using the shortest length: EDN, LDSgN and LCSqN are by far the distances obtaining the smallest values for the three options being these values under 0.5. Besides, these distance functions have shown the greater instability in the 5 runs executed. We could say that for the rest of the distances the clustering algorithm is able to extract the structure of the distance matrix and besides, if the used option is average linkage, the correlation between the distance matrix and the cophenetic matrix is in average 0.96.

4.3 Computational cost

As we mentioned, having into account that malware detection needs to be done efficiently, and thousands of new malware appear every day, another important issue to be taken into account when designing a malware detection tool is the execution time.

Obviously, fastness will be important to discriminate distances when they present similar behaviour but it won't be the only desired feature of a distance. For instance, even if Length and Random are by far the fastest options, they would never been chosen. We have used them just as baseline.

When evaluating the obtained results, it has to be taken into account, that apart from the used distance, the only issue affecting to the time used to build the distance matrix is not the number of examples of the sample; the number of events of each of the examples will have in some cases a lot of influence. This influence will be greater in ED, LCSq and LCSg distances.

Next table (Table 5) shows the time required to build the distance matrices for the evaluated distances in milliseconds. The executions have been done in a Pentium IV, 3.2 GHz, 1G RAM. No times are shown for normalized distances because they will be similar to the original ones. We can observe in the table that there are huge differences between the times required to obtain the distance matrix for different algorithms.

Distance function	Time (ms)	speedup
ED	1,102,673	1
LCSq	63,648	17
LCSg	33,782	33
NCD	2,716	406
Count	347	3,178

Table 5: Time required building the distance matrices in milliseconds. The speedup column shows the speedup of the rest of the distances in respect to Edit Distance (ED)

As expected, the values in the table show that, in general, sequence comparisons take longer than vector projections. On the other hand, we should remember that sequence representation has semantically more meaning. Anyway, the differences between the distances applied to sequences are also very large. ED is the most expensive algorithm and the required time is significantly larger than the next couple: LCSq and LCSg, subsequence and substring comparison. The time spent by the last distance for sequences, NCD, is an order of magnitude smaller than LCSg. This time is reduced again an order of magnitude when the used distance is Count, that is to say, when the vector projection option is combined with the Euclidean distance.

The use of the 5 samples of 100 executables has given us the possibility to make an incremental analysis of the required execution time. Next table shows the time required to obtain distance matrices for the evaluated distances when the size of the sample varies from 100 to 500, 100 by 100. When analyzing these results, it has to be taken into account that even if the number of sequences increases gradually, due to the different length of the sequences, the number of events does not and this will also affect to the obtained times.

The second row in Table 6 shows the total number of events (calculated adding the number of events in each one of the sequences in the sample) of the samples used for the experimentation. No prior predictions can be done; it could happen that different options fit better to different distances, or the same option being always the one fitting the best. Table 4 shows the results for the evaluated distances and algorithms. The values for Random have not been included when calculating the mean (last row in the table).

Sequences	100	200	300	400	500
num. events	85,602	228,501	341,418	434,764	504,223
ED	1,102,673	8,298,168	19,195,187	31,030,387	42,197,967
LCSq	63,648	463,660	1,033,027	1,695,707	2,258,664
LCSg	33,782	246,621	563,657	912,167	1,231,900
NCD	2,716	11,728	26,077	45,657	64,564
Count	347	760	887	937	1,308

Table 6: Time required to building the distance matrices for samples of 100, 200, 300, 400 and 500 examples in milliseconds

We can observe that when computationally more expensive the calculation of a distance is, more affects the number of used sequences to how much time the process needs. This increment is more or less proportional to increase in the size of the sample for the NCD case, whereas it increments more than the proportion the sample increases for the rest of the sequence based distances.

5 Conclusions

Being aware of how important the automatic detection of new malware is, this work presents a methodology and a comparison of the behaviour of two representations, a set of twelve distances and three clustering algorithms when used to clustering malware based on its dynamic behaviour. All this work has been done in an unsupervised learning context and as a consequence the validation of the results is a difficult task. In this context, we propose a way this comparison can be done, adding a metalearning step to obtain more semantic information from the distance comparison part.

There are different conclusions we can draw from the whole work. Concerning to algorithms, there is not doubt average-linkage option behaves better than single linkage and complete linkage. This option works better independently of the used distance; it seems to capture better the structure represented by the distance.

The evaluation of the distances is more complex. There are huge variations from the two points of view: the structure they capture and the computational cost. Even if the choice of the best distance is not evident some of them, such as Length, ED, Count, CountM, EDN, LCSqN and LCSgN, can be discarded because they behave clearly worse than the rest of the distances. On the other hand, being the behaviour of EDX similar to the one of LCSqX, we should never use the first option because of the computational cost.

We concluded that the most interesting group is formed by CountL, NCD, LCSgX and LCSqX distances. Their distance matrices are highly correlated, and the clustering algorithms seem to be able to capture their structure. This could be somehow surprising. We would expect the first one, CountL, to behave the worst because it is based on the vector projection, whereas the rest are based on the sequence representation. Evidently from the computational cost point of view, CountL is by far the fastest option.

In order to evaluate the differences of the dendrograms built with CountL, NCD, LCSgX and LCSqX, a

preliminary analysis of the obtained trees has been done. We could observe that all the dendrograms were similar in their lower parts but differences arise at higher levels, when heterogeneous clusters are forced to merge. This suggests that the interesting part of the tree is practically the same for the four distances. As a consequence, after this analysis the best strategy to use would be the normalized vector projection CountL combined with average linkage option for the agglomerative clustering algorithm. However, we consider that a validation done by security experts or the use of labelled (totally or partially) data would help to clarify the differences among these four distances and choose the best one.

As a consequence, this work could be extended in different senses, on the one hand, due to the great amount of new malware appearing lately, it would be interesting to extend the work to new greater samples and more distances and algorithms could be evaluated using the same methodology. On the other hand, we used just system call sequences to represent malware, but other information such as file numbers or registry key names could be included in order to evaluating the trade-off between the obtained improvement and the efficiency. Besides, this new representation of malware will provide more detailed descriptions of the malware's dynamic behaviour that could be compared to the descriptions provided by known systems such as Norman (Norman Solutions 2003) and CWSandbox (Willems, and Holz 2007). As the reader will have realized, this work has been done using only malware code executions and it would be also interesting to get track of normal code executions and compare their behaviour. Finally, we should try to obtain feedback from security experts in order to confirm the obtained results.

6 Acknowledgements

The authors are grateful to the company S21sec labs for providing the dataset used in this paper and to their security experts for the help they offered to understand and identify the problem. This work has been (partially) supported by the GISD project (FIT-360001-2006-6) under the PROFIT-proyectos-tractores program from the Spanish Ministry of industry, tourism and trade.

This work was partly funded by the Diputación Foral de Gipuzkoa and the E.U.

7 References

- Lee T., Mody J.J. (2006): Behavioral classification. *Proc. of the EICAR*.
- Bailey M., Oberheide J., Mao Z.M., Jahanian F., Nazario J. (2007): Automated classification and analysis of internet malware. *Proc. of the 10th Symposium on Recent Advances in Intrusion Detection*, 178–197.
- Gao D., Reiter M. K., Song D. X. (2005): Behavioral distance for intrusion detection. *Proc. of the 10th Symposium on Recent Advances in Intrusion Detection*, 63-81.
- Yeung D., Yuxin D. (2003): Host-based intrusion detection using dynamic and static behavioral models. *Pattern Recognition* 36:229-243.
- Brugger T. (2004): Data mining methods for network intrusion detection. Technical Report. University of California at Davis.
- Christodorescu M., Jha S., Kruegel C. (2007): Mining specifications of malicious behavior. *ESEC/SIGSOFT FSE* 5-14.
- Norman Solutions (2003): Norman sandbox whitepaper. http://download.norman.no/whitepapers/whitepaper_Norman_SandBox.pdf. Accessed 29 Sep 2008.
- Willems C., Holz T.: CWSandbox, <http://www.cwsandbox.org/>, Accessed 29 Sep 2008.
- Gusfield D. (1997): *Algorithms on strings, trees, and sequences*. Cambridge University Press.
- Li M., Chen X., Li X., Ma B., Vitanyi P.M.B. (2004): The similarity metric. *IEEE Transactions on Information Theory* 50:3250-3264.
- Wehner S. (2005): Analyzing worms and network traffic using compression. Technical Report. CWI, National research institute for mathematics and computer science in the Netherlands.
- Jain A.K., Dubes R.C. (1998): *Algorithms for clustering data*. Prentice-Hall.
- Sneath P.H.A., Sokal R.R. (1973): *Numerical taxonomy*. Freeman W.H. (eds). San Francisco.
- Mirkin, B. (2005): *Clustering for data mining: a data recovery approach*. Chapman & Hall/CRC.
- Halkidi M., Batistakis Y., Vazirgiannis M. (2001): On clustering validation techniques. *Journal of Intelligent Information Systems* 17:107-145.

Service-independent payload analysis to improve intrusion detection in network traffic

Iñigo Perona, Ibai Gurrutxaga, Olatz Arbelaitz, José I. Martín, Javier Muguerza, Jesús M^a Pérez

Dept. of Computer Architecture and Technology
University of the Basque Country
M. Lardizabal, 1, 20018 Donostia, Spain

{inigo.perona, i.gurrutxaga, olatz.arbelaitz, j.martin, j.muguerza, txus.perez}@ehu.es

Abstract

The popularity of computer networks broadens the scope for network attackers and increases the damage these attacks can cause. In this context, Intrusion Detection Systems (IDS) are included as part of any complete security package. This work focuses on nIDSs which work by scanning the network traffic. A service-independent payload processing approach is presented to increase detection rates in non-flood attacks. Three different techniques for payload processing are proposed and they are shown to be able to efficiently detect some of the attack types. Moreover, the proper integration of the knowledge of the different techniques, payload-based and packet header-based, always improves the results. This work leads us to conclude that payload analysis can be used in a general manner, with no service- or port-specific modelling, to detect attacks in network traffic.

Keywords: Intrusion detection systems, unsupervised anomaly detection, payload, AUC.

1 Introduction

The use of computer networks has become very popular lately. This fact broadens the scope for network attackers and increases the damage these attacks can cause. Network attacks compromise the stability of the network or the security of the information stored on computers connected to it. Therefore, it is very important to build systems that are able to detect attacks before they cause damage. Intrusion Detection Systems (IDS) form part of any complete security package and their goal is to detect any action that violates the security policy of a computer system. In this work we will focus on a particular kind of IDS that works by scanning the network traffic and is able to automatically detect intrusions: network Intrusion Detection System (nIDS).

The detection of network attacks by human analysis, where memorization, looking up description libraries or searching sample collections is required, is not effective; it is too time consuming and subjective. As a consequence, automated and robust nIDSs are required in order to successfully confront the problem. In this sense,

data mining techniques have been used mainly to train on labelled data to detect attacks. Analyzing the bibliography, two main approaches can be found in nIDS systems.

The misuse detection approach, which is used in systems such as MADAM/ID (Lee, Stolfo, and Mok 1999) where machine learning techniques are used on labelled data: the classifier learns from a set of labelled connections, where there is normal traffic and attacks, and in subsequent use it recognizes known attacks. These methods have two main problems. On the one hand, it is very difficult to obtain completely labelled network traffic and, on the other hand, these methods are not able to detect new attacks. They need to be revised each time a new kind of intrusion appears and this happens every day. These methods can not solve the “zero-day” problem and as a consequence, the new attacks will always succeed in damaging the system. Nevertheless, the primary objective will be to detect the first occurrence of intrusions and prevent it from damaging any victim.

The second approach, anomaly detection, was originally proposed by Denning 1987 and a survey can be found in Warrender, Forrest and Pearlmuter 1999. This method profiles normal network traffic behaviour and is able to successfully detect attacks when the observed traffic deviates from the modelled behaviour.

When the anomaly detection approach is used, classifiers learn how normal traffic behaves and any anomalous connection is considered to be an attack. As a consequence, if not all the kinds of normal traffic are modelled, high false positive rates can appear in the system. Moreover, they need purely normal data in order to model normal traffic and it is not usual in real systems to have either purely normal data or labelled data. If any attack is left in the hypothetical purely normal data, this attack will be learned as normal traffic and the IDS will never produce an alert related to it.

Taking into account the problems of the two previous approaches, a third one is becoming popular: unsupervised anomaly detection (Portnoy, Eskin and Stolfo 2001). This option works under the assumption that the volume of normal traffic is much greater than the traffic containing attacks, and, furthermore, the intrusions’ behaviour is different from normal data’s behaviour. If these assumptions are true the intrusion detection problem can be confronted in terms of outlier detection. This approach can be used as a stand-alone system, or, even more effectively, it can be combined with a misuse detection or anomaly detection process.

These kinds of systems have two main advantages, they do not need purely normal data and unlabelled data can be used, which is easy to obtain.

Unsupervised anomaly detection methods usually build probabilistic models of the data that will help them to decide whether or not the connections are attacks. In this context, clustering methods can be used as a tool for anomaly detection. The connections will be clustered and instances appearing in small clusters, i.e. anomalies, will be considered intrusions.

The mentioned assumptions make unsupervised anomaly detection methods inadequate for detecting flood attacks. These kinds of attacks usually need to send a large number of packets in a short time, and are used for many kind of “Denial of Service” (DoS) and “Probe” attacks. Since they are based on the emission of many similar packets they will naturally form large groups and, as a consequence, the clustering process will group them in large clusters. Nevertheless, flood attacks are easy to detect and high detection rates can be often achieved by simpler systems that scan network traffic or analyze headers (Noh, Jung, Choi and Lee 2008).

Although most of the flood attacks can be successfully detected by scanning the TCP/IP headers of network packets, this information is not enough to detect most of the non-flood attacks. In these kinds of attacks, mostly “User to Root” (U2R) and “Remote to Local” (R2L), the intruder only has to send very few packets (often, a single one is sufficient) and, as a consequence, it is nearly impossible for systems to use traffic models to detect such anomalies. In this context, the use of the data transported in the packages –payload– becomes crucial to detect intrusions. Notice that R2L and U2R attacks are actually the only ones that allow the intruder to obtain complete control of the attacked system and therefore, they can lead to catastrophic consequences.

The payload of different network connections can be very different. The transferred information usually depends on the kind of service, and, as a consequence its global analysis becomes complicated. This is probably the reason why there are few works where the payload is used to model network traffic and detect the possible intruders. When payload is used with this aim service-specific approaches are developed. For example Krügel, Toth, and Kirda 2002 present a work that focuses on R2L attacks and uses service-specific knowledge to increase the detection rate of intrusions. They have implemented a prototype that can process HTTP and DNS traffic although they only present results for DNS.

Wang, and Stolfo 2004 base their work on profile byte frequency distribution and they compute the standard deviation of the application-level payload flowing to a single host and port during a training phase. The Mahalanobis distance is used during the detection phase and if the distance exceeds a certain threshold the system generates an alarm. This model is also host- and port-specific and also conditioned by the payload length.

These service-specific methods have the disadvantage that they are very context dependent. That is to say, as they are moved to machines offering different services or as new services appear, the system will need to be rebuilt.

Another example of payload processing can be found in the content variables of Kddcup99 (Lee 1999). In this

case, it has been used to obtain some information based on the experts’ experience. This kind of processing is very context dependent and it can only be done for some well known services and protocols. The processing is totally static; it has no learning capability at all. In order for it to be adapted to new situations the experts need to manually analyze the network data and adapt their knowledge to new attacks.

Each kind of method has its advantages and disadvantages: signature-based methods have the advantage of generating few false alarms, whereas anomaly-based systems generally produce a lot of false alarms for unusual but authorized activities, which is not recommendable at all. Anyway, the false negatives (attacks not detected) generated by signature-based methods are far more problematic than the false alarms generated with anomaly detectors.

In order to obtain the best from each kind of intrusion detection system, a combination of some of them is usually the best option. For example, a flood detecting firewall could first filter most flood attacks; a signature-based IDS could then be used to remove the known attacks and unsupervised anomaly detection could finally focus on detecting the unknown attacks.

Our aim in this work is to present an approach where service-independent payload processing can be used to increase detection rates in non-flood attacks. With this aim we have first analyzed different approaches to payload processing in order to see to what extent just the information provided by the payload can be used to detect intrusions. Next, we have evaluated the combination of the payload-based approaches with the information that can be obtained from network packet headers. After this preliminary work it seems that the techniques for payload processing are able to efficiently detect some of the attack types. Furthermore, they can be used to complement techniques based on packet header analysis, since the combinations tried always improve the results.

The paper proceeds to describe, in Section 2, the approximation that will be used in this work for outlier detection. Section 3 is devoted to describing the three global approaches proposed to process payload without any context knowledge. In Section 4 we describe the data used in the experimentation and experimental results are presented in Section 5. Before the conclusions we summarize in section 6 the schema of the proposed system. Finally, Section 7 is devoted to presenting the conclusions and further work.

2 Detecting outliers

As we mentioned in the introduction, unsupervised anomaly detection strategies can be formulated as outlier detection problems. The approximation we have used to detect outliers could be based on any clustering algorithm. The idea is to first perform the clustering over the points in the feature space and assign a score to each of them based on their size. The examples in each cluster will have the same score the cluster has, and this score will be used to determine the degree of anomaly of the example. The points with lower scores will be labelled as anomalous. Although many clustering algorithms could be used, based on the experience of other authors (Eskin, Arnold, Prerau, Portnoy, and Stolfo 2002, Leung, and

Leckie 2005), we have selected the fixed-width clustering algorithm (Eskin, Arnold, Prerau, Portnoy, and Stolfo 2002), also known as the leader algorithm (Spath 1980). The fixed-width algorithm has the advantage that it scales linearly to the number of examples of the database and the number of clusters, but, on the other hand, it has the disadvantage that the quality of the clusters is sensitive to the definition of the radius w and often several runs are needed to select the best w in any particular environment. Moreover, this algorithm does not accurately fit to databases with clusters of different sizes; the largest clusters are usually over-partitioned. Nevertheless, in the unsupervised anomaly detection context we are interested just in the small groups, so this drawback of the algorithm is not a real problem.

3 Payload processing

Network Intrusion Detection Systems usually focus on the analysis of the TCP/IP headers of the packets detected on the net. The format of these headers is well-known and, therefore, this data can be easily processed. This information might not be enough to detect intruders and as a consequence, the analysis of the transferred data, the payload, will need to be performed. The format of the payload data in a packet depends on the application protocol used, so that in this case data processing becomes a difficult task. Moreover, many protocols have fields where any kind of data can be stored. Many previous works have solved this problem by performing the data processing in a specific way for each service. Obviously, this method requires knowledge of the different protocols and has many drawbacks: it only works for a reduced set of connections, the used protocol is not always known, new services are not automatically treated...

Payload data can generally be seen as a sequence of bytes, so we think that it could be processed, regardless of the service used, by using sequence comparison techniques. The aim of this work is to use this payload information as a tool to help detecting attacks in nIDSs. In this context, it is important for the selected payload processing method to be efficient in detecting attacks and to achieve low false positive rates, but, it also requires having some other characteristics such as:

1. To be automatic. That is, not requiring human intervention.
2. To be general. That is to say, service-independent, and, as a consequence, usable in different environments and adaptable to changing situations.
3. To be computationally efficient so that it can operate in real time, in environments with large bandwidth.

It is not easy to build a system with all the required skills; it seems, on the one hand, that more complex or computationally expensive systems would better model the payload, and, on the other hand, that service- and models are easier to build. As we mentioned in the introduction, some work has already been done in this context, as for example Krügel, Toth, and Kirda 2002, Wang, and Stolfo 2004, Leung, and Leckie 2005.

The aim of this work is to build a system that achieves all the required characteristics, as far as possible, and to demonstrate that payload can be used to improve the behaviour of IDSs in a general and computationally effective way. In this context, several payload processing strategies have been tried:

1. The first and probably most intuitive option seems to directly use the payload as a byte sequence and use sequence comparison methods to model different payloads. It can be easily concluded that distances such as the Edit Distance (Gusfield 1997), where the distance between two strings of characters is the minimum number of edit operations on individual characters needed to transform one string to another, are computationally too expensive. As a consequence, this approach will only be useful if a more efficient method for comparing sequences can be used. In this context we have used the Normalized Compression Distance (NCD) proposed in Li, Chen, Li, Ma, and Vitanyi 2004, Wehner 2005. This distance is based on the Kolmogorov complexity. Since in practice Kolmogorov complexity cannot be computed directly, it is approximated with real compression algorithms. In this work we used the standard compression algorithm GZip. $C(x)$ is defined as the length of the string obtained by compressing x , and $C(xy)$ as the length of the string obtained by compressing the concatenation of x and y . The Normalized Compression Distance is defined as follows:

$$NCD(x,y) = \{C(xy) - \min(C(x), C(y))\} / \max(C(x), C(y)).$$

2. The well known n-gram analysis is another option that could be used to model payload. To make it computationally efficient a 1-gram model, where the frequency of each one of the 256 values is computed, could be used. This method has been used by some other authors (Wand, and Stolfo 2004). Even the 1-gram option can have some limitations on the size of the database used, since $256 \text{ variables} \times \text{number of examples}$ can be too large when working with large databases. On the other hand, it is improbable that all 256 variables would be significant in each of the payloads and the treatment of all of them could even be counterproductive.
3. Based on the byte-frequency idea, we think that the sequence of the most frequent bytes could in some sense be more significant for representing the payload. Similar connections will probably have similar payload patterns and as a consequence we could expect also the most frequent bytes to be similar. That is why the 1-gram representation of the payload has been calculated and the 256 possible values ordered from the most frequent to the least frequent. The information referred to the N most used bytes has been used (we have tried 3 different values for N : 15, 30 and 50). The payload byte-ordered

vectors PBO_1 and PBO_2 could be two examples of the representation of two different payloads (P_1 and P_2):

PBO_1	8A	F1	05	AE	87		91
1	2	3	4	5	...		N

PBO_2	F1	8A	05	AE	90		8C
1	2	3	4	5	...		N

In this way, a new representation of the payload is achieved which can be used in different ways:

1. Each byte in the new vector could be treated as an independent variable and the Euclidean Distance used to compare two payloads. The distance between two byte values has been set to 1 if they are different and 0 otherwise. We will call this option Freq1.
2. The representation can be considered a sequence where the position in which each character appears influences the distance. In this context, a distance has been defined to compare the representation of two payloads. If two payloads are compared and their corresponding payload character-ordered vectors (PBO_1 and PBO_2) are obtained, the distance could be computed with the expression:

$$Dist(PBO_1, PBO_2) = 1 - \frac{\sum_{i=1}^N Sim(i)}{N^2}$$

where

$$Sim(i) = \begin{cases} 0 & \text{if } PBO_1(i) \neq PBO_2(j), \forall j \ 1 \leq j \leq N, j \neq i \\ N - |i - j| & \text{if } \exists j \mid PBO_1(i) = PBO_2(j), 1 \leq j \leq N, j \neq i \end{cases}$$

We will call this option Freq2.

4 Data generation

As we mentioned in the introduction, it is not easy to obtain labelled data for network traffic, and neither a database with purely normal data. As a consequence, it is also difficult to evaluate intrusion detection systems and compare results. Even if unsupervised anomaly detection techniques do not require labelled data to work, this kind of data is required so that the system can be evaluated. In order to generate comparable results, we have decided to use some standard data such as Kddcup99 from the UCI repository (KDD99-Cup 1999). This database was built by processing the DARPA98 dataset (DARPA 1998), which was generated by the Information System Technology Group (IST) of the Lincoln Laboratory of the MIT with the collaboration of DARPA and ARFL. They built a network to simulate a real situation of network traffic containing normal traffic and attacks. Tcpdump (Jacobson, Leres, and McCanne 1989) was used to sniff the network and store all the packets belonging to network traffic in a tcpdump file. Based on this information the UCI format Kddcup99 database was generated by identifying connections and aggregating information belonging to them. Based on the information in DARPA98, three kinds of features were generated for

each connection: intrinsic variables (those obtained by examining the packets' TCP/IP structure such as protocol, length, urgent bit...); content variables which were obtained by examining the payload of some particular services, such as number of failed logins, number of operations generated as root, number of file creations; and, finally, traffic variables which take into account header information of preceding connections contained in a window of some specific size.

The advantage of the KddCup99 database is that it processes the huge amount of information in the DARPA98 dataset and it stores it in a format suitable for most machine learning algorithms. The problem is that the original payload information is not stored, and just some manually defined attributes (content variables) keep part of the payload-based information. This manual solution is obviously not a general solution. Based on Lee 1999 and using Bro (Paxson 1998), we have reprocessed the DARPA98 database to embellisher Kddcup99 database with information from the original DARPA98. In this new database, each connection will have the intrinsic and traffic variables found in the original database added to all the payload data corresponding to it. The content variables have not been computed and added to the new database, because the aim of this work is to replace the information they provide by automatic payload processing.

Attacks	Quantity	percentage
anomaly	9	0.23
dict	879	22.33
dict_simple	1	0.03
eject	11	0.28
eject-fail	1	0.03
ffb	10	0.25
ffb_clear	1	0.03
format	6	0.15
format_clear	1	0.03
format-fail	1	0.03
ftp-write	8	0.20
guest	50	1.27
imap	7	0.18
land	35	0.89
load_clear	1	0.03
loadmodule	8	0.20
multihop	9	0.23
perl_clear	1	0.03
perlmagic	4	0.10
phf	5	0.13
rootkit	29	0.74
spy	2	0.05
syslog	4	0.10
teardrop	1085	27.56
warez	1	0.03
warezclient	1749	44.42
warezmaster	19	0.48
Total attacks	3937	

Table 1: Names and quantities of attacks that appear in the database used for experimentation

Due to the huge size of the original Kddcup99 database (about 5,000,000 connections), most of the experiments found in literature have been performed using a sample of the original dataset. This sample contains about the 10% of the connections found in the complete dataset. Similarly, we have extracted a stratified sample of about 10% of the size of the original one. Since our goal is to find the non-flood attacks, and the DARPA98 is overloaded with flood attacks, we have filtered all the flood attacks in the dataset. Thus, we have been working with a database of 178,810 examples, where 3,937 examples belong to intrusions of 27 different kinds. The information about the kind of attacks appearing in the database used for experimentation and their frequency is shown in Table 1.

5 Experimental results

5.1 Evaluation of different representations

Our objective is to use the payload in a general way for unsupervised anomaly detection so that intrusions are differentiated from normal traffic. The idea is to combine the information the payload can provide with the TCP/IP header information (intrinsic and traffic variables in Kddcup99). As we mentioned in Section 3, we have tried several options for processing the information provided by the payload.

The first step has been to analyze how each one of the representations works separately for anomaly detection with no more information than just payload and to compare the results obtained with those obtained with TCP/IP header information i.e. the equivalent of the intrinsic and traffic variables of KddCup99. Each of the representations has been used to build a model of the system using a fixed-width algorithm with an adequate value for w .

The experimentation has been done with the whole database, that is, normal traffic data plus data from 27 different kinds of attacks. But, in order to present the results obtained in our experimentation we have chosen to examine the detection rates of each type of attack separately instead of examining a global result for all of them. We have noticed that in many previous works where Kddcup99 was used for experimentation a single global value was used to measure the accuracy of a particular detection system. Due to the high number of connections belonging to certain attacks (mostly Smurf and Neptune) the results are heavily biased to the detection level the system obtained for those particular attack types. Although we have removed all the flood attacks from our dataset, differences in the number of attacks of each type exist (see Table 1). Our results will be always analyzed for each attack type, and, therefore, all attack types will have the same weight when presenting the final results. Otherwise the overall results would practically ignore the attack types other than Warezclient, Teardrop and Dict. The evaluation has been done by analyzing the ROC curves and the Areas Under ROC Curves, or AUC values, obtained (Fawcett 2004). To compute the ROC of just a single attack type, the examples belonging to other attack types have been ignored.

In general terms, the first conclusion that can be drawn from this processing is that although no context knowledge is used and simple processing is performed, the three options for modelling payload (NCD, Freq1, Freq2) are, to a certain extent, able to differentiate between normal traffic and intrusions; the average AUC values are specifically 0.77 for NCD, 0.74 for Freq1 and 0.72 for Freq2 (see Table 3 for AUC values in each attack). Furthermore, the model built using header information, IT, achieves an AUC value of 0.80, which is not very different from the previous ones. In order to take the conclusions a bit further we have made a pair wise comparison of the 4 different techniques. In each comparison we have compared the AUC values obtained for each attack type and counted the number of times each technique obtains greater values than the other. The cells in Table 2 show the number of attacks (out of 27) the technique of the corresponding row detects better than the technique of the corresponding column. The last column shows the number of times the corresponding technique behaves better than any of the rest.

	IT	NCD	Freq1	Freq2	Total
IT	0	10	14	16	40
NCD	17	0	17	13	47
Freq1	13	10	0	16	39
Freq2	11	14	11	0	36

Table 2: Pair wise comparison of the different techniques. Summary of the number of times each technique behaves better than the other

In order to interpret the results it is important to notice that if a certain technique was always better than another one, the value in the cell corresponding to their comparison would be 27 (values of around 13-14 would mean similar behaviour), and if that happened for all three comparisons that can be made for each classifier, the total value (column Total) would be 81. Values in Table 2 are, in general, not very far from 13. This means that the number of attacks each classifier is able to detect better is similar. However, if we observe the general behaviour of different options it seems that one of them, NCD, behaves better than the rest. This means that, we would be able to detect more attacks with general payload processing, than by analyzing header information (IT).

5.2 Combining results

If the ROC curves and AUC values obtained for each of the attacks are analyzed, it can be observed, that, even if similar detection rates are achieved on average, each one of the classifiers built specializes better in detecting some kinds of attacks. As an example, we have observed that Teardrop, Warez, Format-clear and Warezclient attacks are better detected by IT, NCD, Freq1 and Freq2 approaches respectively (see Table 3).

This leads us to think that a good option could be to combine the knowledge acquired in all of them; but, how could be this combination be done? The output of each of the classifiers built can be seen as a list of all of the classified patterns with their corresponding scores.

Attacks	IT	NCD	Freq1	Freq2	IT- NCD- Freq1	IT- NCD- Freq2	IT- Freq1- Freq2	IT- NCD- Freq1- Freq2
anomaly	0.76	0.88	0.35	0.74	0.72	0.9	0.68	0.77
dict	0.76	0.82	0.64	0.83	0.83	0.93	0.84	0.89
dict_simple	0.65	0.81	0.69	0.83	0.81	0.91	0.81	0.88
eject	0.76	0.82	0.8	0.80	0.9	0.9	0.88	0.92
eject-fail	0.99	0.48	0.99	0.58	0.94	0.8	0.98	0.9
ffb	0.8	0.88	0.72	0.93	0.9	0.97	0.92	0.94
ffb_clear	0.65	0.81	0.71	0.67	0.82	0.84	0.73	0.84
format	0.79	0.93	0.81	0.95	0.95	0.98	0.96	0.98
format_clear	0.52	0.81	0.88	0.83	0.84	0.85	0.84	0.9
format-fail	0.98	0.81	0.8	0.67	0.98	0.95	0.93	0.96
ftp-write	0.88	0.88	0.76	0.56	0.94	0.88	0.82	0.89
guest	0.77	0.85	0.81	0.83	0.92	0.94	0.92	0.95
imap	0.9	0.7	0.97	0.68	0.94	0.85	0.96	0.92
land	0.92	0.48	0.99	0.58	0.9	0.76	0.95	0.87
load_clear	0.65	0.81	0.12	0.14	0.54	0.56	0.19	0.43
loadmodule	0.7	0.71	0.69	0.68	0.77	0.79	0.77	0.8
multihop	0.72	0.78	0.63	0.71	0.76	0.8	0.73	0.77
perl_clear	0.95	0.81	0.52	0.87	0.87	0.98	0.91	0.93
perlmagic	0.66	0.83	0.86	0.86	0.88	0.91	0.9	0.93
phf	0.9	0.71	0.99	0.72	0.96	0.89	0.97	0.95
rootkit	0.88	0.77	0.86	0.77	0.93	0.9	0.94	0.93
spy	0.71	0.81	0.66	0.52	0.8	0.77	0.68	0.77
syslog	0.82	0.48	0.97	0.58	0.87	0.7	0.92	0.84
teardrop	0.96	0.48	0.76	0.58	0.83	0.78	0.88	0.81
warez	0.82	1.00	0.12	0.85	0.68	0.98	0.64	0.82
warezclient	0.81	0.86	0.86	0.86	0.95	0.96	0.96	0.97
warezmaster	0.94	0.87	0.96	0.88	0.98	0.98	0.98	0.99
Average	0.80	0.77	0.74	0.72	0.86	0.87	0.84	0.87

Table 3: AUC values achieved for all the attack types in different classifiers; single ones and combined ones

As we mentioned in section 2, the score assigned to each of the patterns is used to determine whether the pattern is anomalous or not. As a consequence, the most direct way of combining the different techniques will be to combine the different scores obtained for each connection. Since the distribution of the score values assigned by each technique depends on the number and size of the clusters, it varies from one model to another. This fact means that the direct combination of the score values is probably not adequate (this option has been tried and suspicions have been confirmed). As a consequence, some normalization is required to put the scores of the different classifiers in the same situation. Two options for normalizing the scores in each one of the examined techniques have been tried in this work:

1. Linear 0-1 normalization. Normalize all scores linearly, so that the minimum value is 0 and the maximum is 1.
2. Rank normalization. In this normalization all the connections are sorted by their score (connections with equal score are sorted arbitrarily). The rank for each connection will be

its new score. If more than one connection has the same score, the average rank value of the group of connections with equal score is computed and the result assigned as the new score for all the connections in the group.

The second option is more appropriate (as we have observed in the results), since it is more independent of the specific scores obtained by each of the classifiers and it depends on the position each connection has in the ranking.

Independently of the values used to compare the scores, how should we combine the different values? We could probably think of many different combination strategies but for this work we have tried three of them:

1. Select the minimum score for each connection.
2. Select the maximum score for each connection.
3. Average the score of the combined techniques for each connection.

The three proposed combining methodologies achieve reasonable results, but in general, better results have been obtained by averaging the scores.

The last point to decide at this stage is to select which classifiers' results to combine. In the results presented in the previous section we worked with 4 classifiers and we could combine all of them or select pairs or trios and combine them. We tried all the possible combinations between the four techniques evaluated in Section 5.1. The effect of the combination is in most cases positive but it becomes more positive when more than two techniques are combined.

Table 3 shows AUC values achieved for every attack type with each one of the techniques used: the four options used independently and all the possible combinations of three and four techniques where IT (connection header information) is always maintained. IT has not been ignored in any of the combinations because, as we mentioned in the introduction, our aim is to provide our system with both header and payload information. The best AUC for each of the attacks (taking into account all decimal values) is marked in bold.

IT	72
NCD	72
Freq1	61
Freq2	55
IT-NCD-Freq1	114
IT-NCD-Freq2	130
IT-Freq1-Freq2	109
IT-NCD-Freq1-Freq2	143

Table 4: Pair wise comparison of the 8 different techniques. Total number of times each technique behaves better than the other

Average AUC values show that the general behaviour of any of the combinations improves the behaviour of the classifiers obtained with each independent option. We could consider as the best option the combination of the 4 techniques because, even if the average AUC value obtained is similar to the one obtained with the IT-NCD-Freq2 option, it achieves the best AUC for more different kinds of attacks (9 out of 27).

The same kind of pair wise comparison presented in Table 2 has been performed for all the evaluated techniques. The global results —number of times each classifier behaves better than the other out of 189— for the eight techniques evaluated in Table 3, are presented in Table 4 and they confirm the conclusions drawn from Table 3. Results for combinations of two techniques are in general between the results of single techniques and results of the combination of three or more techniques.

6 Schema of Intrusion detection process

For clarity, in this section we summarize the steps the payload based intrusion detection tool we propose needs. Figure 1 shows a schema of the process.

Once network data is collected we will divide it in two main parts: the connections' headers information on the one hand, and the transferred information or payload on the other one. The TCP/IP headers' information will be processed to obtain a tabular representation with intrinsic variables and traffic variables. The payload part will be processed to obtain information about the frequency of the transferred bytes and represent it also in a tabular manner.

Next step is to apply fixed width clustering algorithm with the distances defined in Section 3 to the headers' information, frequency based representation of payload and payload without processing. Four different partitions will be obtained at this point, IT, Freq1, Freq2 and NCD, with the corresponding scores for the connections.

Finally the scores will be combined to obtain the final score for each connection. These scores will be the ones used to determine the degree of anomaly of the connection.

7 Conclusions and further work

In this work we have proposed three different techniques for payload processing in order to use it to detect attacks in network traffic. The three options proposed have been shown to be able to efficiently detect some of the attack types. Although most of the previous works were based on packet header analysis, this work has shown that general payload analysis can also be effective.

Since the different techniques have shown the ability to detect different kinds of attacks, we have decided to combine them. The results obtained have shown that it is possible to integrate the knowledge of the payload-based techniques and the packet-header-based technique and improve the original results. The combination of all the options tried is in particular the one obtaining the best results.

Based on the results presented we can state that in the data set we have used for experimentation, payload analysis can be used in a general manner, with no service- or port-specific modelling, to detect attacks in network traffic. Obviously, this technology needs still to be tested in real environments.

This is a preliminary work where just some of the possible payload information processing options have been tried. Other techniques could be tried in the future. The way in which classifiers can be combined is another area where a deeper analysis can be carried out and more sophisticated approaches tried. The possibility of using other clustering algorithms and the optimization of their parameters is also an area where more work can be done.

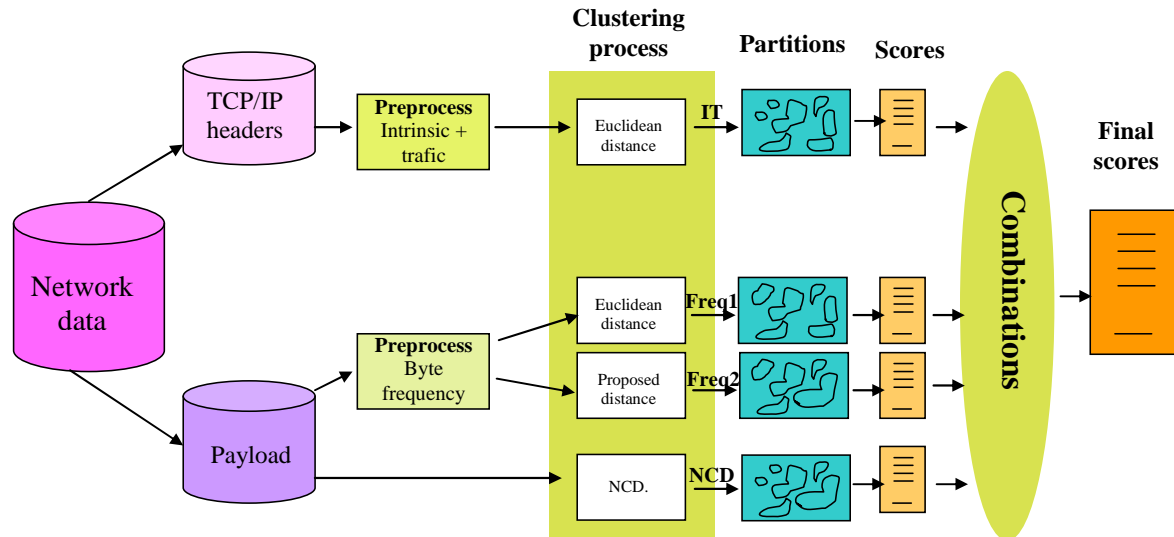


Figure 1: Schema of the proposed intrusion detection process

8 Acknowledgements

This work was partly funded by the Diputación Foral de Gipuzkoa and the E.U.

9 References

- Lee W., Stolfo S.J., Mok K. (1999): Data mining in work flow environments. Experiences in intrusion detection. *Proc. of the Conference on Knowledge Discovery and Data Mining*.
- Denning D.E. (1987): An intrusion detection model. *IEEE Transactions on Software Engineering* **13**:222-232.
- Warrender C., Forrest S., Pearlmuter B. (1999): Detecting intrusions using system calls: alternative data models. *Proc. IEEE Symposium on Security and Privacy*, 133-145.
- Portnoy L., Eskin E., Stolfo S (2001): Intrusion detection with unlabeled data using clustering. *Proc. ACM Workshop on Data Mining Applied to Security*.
- Noh S., Jung G., Choi K., Lee C. (2008): Compiling network traffic into rules using soft computing methods for the detection of flooding attacks, *Applied Soft Computing* **8**(3):1200-1210.
- Krügel C., Toth T., Kirda E. (2002): Service specific anomaly detection for network intrusion detection. *Proc. ACM Symposium on Applied Computing*, Madrid, Spain, 201-208, ACM Press.
- Wang K., Stolfo S. (2004): Anomalous payload-based network intrusion detection. *Proc. International Symposium on Recent Advances in Intrusion Detection*, LNCS, 203-222.
- Lee W. (1999): A data mining framework for constructing features and models for intrusion detection systems. Ph.D. thesis. Columbia University.
- Eskin E., Arnold A., Prerau M., Portnoy L., Stolfo S. (2002): A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. *Data Mining for Security Applications*.
- Leung K., Leckie C. (2005): Unsupervised anomaly detection in network intrusion detection using clusters. *Proc. Australian Computer Science Conference*.
- Spath H. (1980): *Cluster analysis algorithms*. Ellis Horwood, Chichester, UK.
- Gusfield D. (1997): *Algorithms on strings, trees, and sequences*. Cambridge University Press.
- Li M., Chen X., Li X., Ma B., Vitanyi P.M.B. (2004): The similarity metric. *IEEE Transactions on Information Theory* **50**:3250-3264.
- Wehner S. (2005): Analyzing worms and network traffic using compression. Technical Report. National research institute for mathematics and computer science in the Netherlands.
- KDD99-Cup (1999): The third international knowledge discovery and data mining tools competition dataset <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>. Accessed 29 Sep 2008.
- DARPA (1998): MIT Lincoln Laboratory - DARPA Intrusion Detection Evaluation <http://www.ll.mit.edu/IST/ideval/index.html>. Accessed 29 Sep 2008.
- Jacobson V., Leres C., McCanne S. (1989): Tcpdump. Available via anonymous ftp to <ftp://ee.lbl.gov>. Accessed 29 Sep 2008.
- Paxson V. (1998). Bro: a system for detecting network intruders in real-time. *Computer Networks* **31**:23-24.
- Fawcett T. (2004): ROC graphs: notes and practical considerations for researchers. Technical Report HPL-2003-4. HP Laboratories, Palo Alto, CA, USA.

Graphics Hardware based Efficient and Scalable Fuzzy C-Means Clustering

S.A. Arul Shalom¹ Manoranjan Dash² Minh Tue³

^{1,2} School of Computer Engineering,
Nanyang Technological University,
Block N4, Nanyang Avenue, Singapore 639798

¹ sall10001@ntu.edu.sg ² asmdash@ntu.edu.sg

³ NUS High School of Mathematics and Science,
20 Clementi Avenue 1
Singapore 129957

³ h0630082@nus.edu.sg

Abstract

The exceptional growth of graphics hardware in programmability and data processing speed in the past few years has fuelled extensive research in using it for general purpose computations more than just image-processing and gaming applications. We explore the use of graphics processors (GPU) to speedup the computations involved in Fuzzy *c*-means (FCM). FCM is an important iterative clustering algorithm, and usually performs better than *k*-means. But for large data sets it requires substantial amount of time, which limits its applicability. FCM is an iterative algorithm that involves linear computations and repeated summations. Moreover, there is little reuse of the same data over FCM iterations (i.e., the centre of the clusters change in each iteration) and these characteristics make it a good candidate to be mapped to the parallel processors in the GPU to gain speed. We look at efficient methods for processing input data, handling intermediate results within the GPU with reusability of shader programs and minimizing the use of GPU resources. Two previous implementations of FCM on the graphics-processing unit (GPU) are also analysed. Our implementation shows speed gains in computational time over two orders of magnitude when compared with a recent generation of CPU at certain experimental conditions. This computational time includes both the processing time in the GPU and the data transfer time from the CPU to the GPU.

Keywords: Fuzzy *c*-means, GPGPU, Clustering, Parallel Computation

1 Introduction

Clustering finds out hidden patterns in the data set by grouping similar data objects together. It does not require any prior knowledge of the data objects and about the groups they belong to. Typically there are three types of

clustering algorithms: partitioned, hierarchical and density based. (Jain, Murty and Flynn 1999). In the partitioned clustering algorithm (MacQueen 1967) the number of clusters is specified and the clustering algorithm uses similarity measure to determine the clusters. Hierarchical clustering (Guha, Rastogi and Shim 1998) can be divisional or agglomerative. In the agglomerative hierarchical clustering algorithm, each data object is considered as a cluster and the closest pair of clusters is merged in iteration repeatedly until there remains only one cluster, thus producing a dendrogram. The dendrogram can be used to obtain the clusters as per the required number of clusters. Density based clustering (Ester, Kriegel, Sander and Xu 1996) is based on density parameters. Data objects are considered connected to a cluster or disconnected depending on the density parameters. In this paper we focus on an important partitioned clustering algorithm called fuzzy *c*-means (FCM) (Bezdek 1981).

The *k*-means clustering method or the hard *c*-means algorithm groups *n* objects in a data set into *c* clusters. To begin this iterative process, the initial *c* cluster centres are predetermined. In hard clustering, data is divided into distinct clusters, where each data element belongs to exactly one cluster. In fuzzy clustering, such as FCM, data elements can belong to more than one cluster. Each data element is associated with a set of membership values. These indicate the strength of the association between that data element and a particular cluster.

As will be shown in later section, FCM is based on the standard least squared errors model. FCM is very popular due to several reasons. It can be generalized in many ways. Arguably it is much easier to generalize FCM than the hard *c*-means clustering. For example, the memberships are generalized to include possibilities; the distance used has been generalized to include Minkowski (non-inner product induced) and hybrid distances; there are versions of FCM for very large data sets that utilize both progressive sampling and distributed clustering; there are many techniques that use FCM clustering to build fuzzy rule bases for fuzzy systems design; and there are numerous applications of FCM in virtually every major application area of clustering (Wiki 2008). The various high volume data visualization applications of FCM include image

Copyright © 2008, Australian Computer Society, Inc. This paper appeared at conference Seventh Australasian Data Mining Conference (AusDM 2008), Glenelg, Australia. Conferences in Research and Practice in Information Technology, Vol. 87. John F. Roddick, Jiuyong Li, Peter Christen and Paul Kennedy, Eds. Reproduction for academic, not-for-profit purposes permitted provided this text is included.

segmentation, multi-spectral image compression, remote sensing, object recognition, biological sequence analysis, clustering co-expressed genes, and to hybridise various other data mining algorithms.

Clustering large amounts of data takes a long time. To cluster these large data sets, either sampling is required to fit the data in memory or the time will be greatly affected by disk accesses making iterative clustering (e.g., *k*-means, fuzzy *c*-means) an unattractive choice for data analysis. In this paper we focus on how to ease the computational bottleneck of FCM on large data sets using graphics processors (GPU).

Many researchers are able to use GPU for data mining algorithms over large data sets (Owens, Luebke, Govindaraju, Harris, Krüger, Lefohn and Purcell 2005). This area of research is known as GPGPU (General Purpose Computations using GPU). Although GPUs are quite powerful due to their internal architecture they favour algorithms that can be structured as streaming computations often realizing notable performance gains (Fatahalian, Sugerman and Hanrahan 2004). Streaming computations can be characterized as being highly parallel and numerically intensive. One such suitable application is FCM. It is streaming in nature but its data (the centroids) change from iteration to iteration. Hard *c*-means (*k*-means) which is the primate of the FCM is efficiently implemented in the GPU (Arul, Dash and Tue 2008).

In Section 2 we briefly discuss the GPU hardware features that enhance its application in GPGPU. Section 3 describes briefly the previous FCM implementations on the GPU and their results. In Section 4 our proposed implementation and its novelties are discussed addressing scalability issues. Section 5 discusses experimental setup, shows the analysis and results. Section 6 is the conclusion with a brief discussion on future expansions on FCM and other clustering methods using GPU.

2 Exploiting the Modern Graphics Hardware for General-Purpose Computations

The GPU has tremendous image processing capabilities such as vertex transformation, lighting computations, clipping and culling of images using its highly parallel hardware pipeline. For instance, the massively parallel GPU, Nvidia's GeForce 8800 GTX, consists of 128 individual stream processors each running at 1.35 GHz clock frequency, with very high memory bandwidth of 86.4 Gigabytes per second. (NVIDIA: GeForce 8800 Architecture Technical Brief 2008). The GeForce 8800 GPU's shader architecture is designed for extreme 3D graphical performances, producing near reality image quality for delighted gaming performances, which is its traditional forte. Figure 1 shows the block diagram of the GeForce 8800 GPU, which shows the various stages of the parallel programmable processors.

2.1 GPU as a Low Cost High Power Computational Processor

In Figure 1 the host forms the interface block between the CPU and the GPU. In graphics processing, the host receives the commands from the CPU, geometric data and other display data. The input data from the CPU is assembled and formatted before the next stage of graphics

processing. Each of the GPU's internal processors could be assigned to a specific shader program. Shaders are short lines of codes that run on the stream processors which process incoming stream data and send the computed data to output buffers or textures. The stream processors are grouped in a manner so that computational resources can be efficiently mapped to these processors. The processed data can be sent as stream data to other stream processors for further processing. Such computations are possible due to data independency in graphics processing. This also permits multiple shader programs to run on the processors, each shader accessing data in parallel that is linked to the stream processor.

The stream processing capabilities of the GPU makes it highly applicable to implement general-purpose computations. Computations to be implemented in the GPU will need to be mapped appropriately using the hardware resources such as textures and frame buffers.

The programmability of the stream processor is achieved using shader programs. Various general-purpose computations such as physical simulations, image processing and data mining algorithms have been implemented on GPU, harnessing its computational power and programmability to improve computational efficiency as compared to the CPU (Owens, Luebke, Govindaraju, Harris, Krüger, Lefohn and Purcell 2005). The GPU thus has become a low cost commodity processor with high computational power, for which the growth is heavily driven by the gaming industry. The cost of the speed gained from using such a GPU is much lower than the CPU based massive parallel processors. We intend to efficiently implement the FCM computations using GPU.

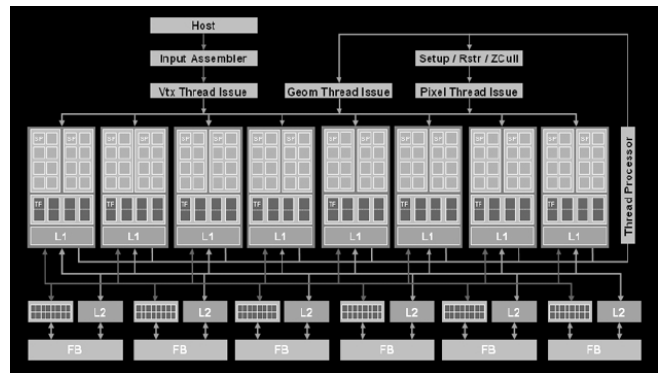


Figure 1: GeForce 8800 GTX Block Diagram

3 FCM Algorithm and Existing GPU based Implementations

The FCM algorithm can be summarized using the following simple steps:

1. Initialise cluster memberships
2. Calculate cluster centres
3. Update cluster memberships
4. Check stopping condition, else go to Step 2.

The FCM algorithm partitions a set of feature vectors x_i into c clusters by minimizing the objective function given by $J(U_{ij}, C_j)$ in equation 0, where m is a real integer greater than 1, U_{ij} denotes the degree of membership of the d -dimensional vector x_i in the cluster j and C_j is the centre

of that cluster. The norm $\|*\|$ expresses the closeness of the vector to its cluster centre.

$$J(U_{ij}, C_j) = \sum_i \sum_j U_{ij}^m \|x_i - c_j\|^2 \quad (0)$$

In this iterative fuzzy partitioning optimization process on the data set of size n , the cluster memberships U_{ij} of each observation i , to the c clusters is computed by equation (1).

$$U_{ij} = 1 / \sum_{k=1}^c \left[(x_i - c_j) / (x_i - c_k) \right]^{2/(m-1)} \quad (1)$$

This equation is also used for cluster centroid updates, where m is the fuzzifier. The value m determines the amount of fuzziness. The value of m can be chosen from $(1, \infty)$. A value of $m=1$ produces a hard clustering. As m approaches ∞ the solution approaches its maximum degree of fuzziness. It is often chosen on empirical grounds to be equal to 2. In FCM, the fuzzy centroids depend on the current membership values and all the individual observations i . The fuzzy centroid C_j is computed using the equation (2).

$$C_j = \sum_{i=1}^n U_{ij}^m x_i / \sum_{i=1}^n U_{ij}^m \quad (2)$$

This iteration will stop when the termination criterion given in equation (3) is fulfilled, ε is a termination criterion between 0 and 1.

$$\text{Max}_{ij} \{ |U_{ij}^{k+1} - U_{ij}^k| \} < \varepsilon \quad (3)$$

The steps of the FCM algorithm are further briefly explained here. The first step involves the initialisation of the initial cluster memberships and also includes the initialisation of the clustering variables. The value for m is chosen to be 2, the number of clusters c is set as predefined for the n number of observations in the data set. The distance between the c initial clusters and the individual observations are computed, which is the inner product norm between the vectors. To end step 1, the values of the cluster memberships for each observation is computed using equation (1).

In the second step the centre of the clusters is calculated using equation (2). The fuzzy centroid C_j represents the vector location of the centre of the j^{th} cluster. The cluster centres are thus computed for the all the c clusters.

In the third step the memberships are updated with the new values based on the distances of each observation to each of the cluster centre. Equation (1) is again used for this computation.

In the last step, the algorithm is checked for its stopping condition using equation (3). The stopping condition should be predefined. If the error between the current cluster centres and the corresponding previous centres are less than 0.00001, the computations of the algorithm ends.

In fuzzy c -means iterations, the utilization of computational resources is high and is mainly contributed by:

1. Distance computation between the objects and the cluster centres.

2. Computing the degree of membership for all objects in every cluster.

3. Computing new cluster centres as a function of the degree of membership.

The implementation of FCM in the GPU will reduce the computational time, utilizing the computational resources of the Graphics hardware.

3.1 Previous GPU Implementations of FCM

There are two previous works where FCM has been implemented in the GPU (Harris and Haines 2005), (Anderson, Luke and Keller 2007). Reduction in computational time in the order of 2x times has been achieved from a non-iterative GPU implementation when compared to the CPU FCM (Harris and Haines 2005). This implementation is able to handle huge number of observations, but not scalable in terms of dimensions and the number of clusters. In the second work, the authors present a FCM with non-Euclidean distance computation metric and have demonstrated processing time gains of over two orders of magnitude for certain configurations of FCM, where different combinations of data size, dimension size and clusters are used.

While implementing the FCM on the GPU the following considerations are to be carefully made so as to avoid the drawbacks of the previous FCM implementation (Anderson, Luke and Keller 2007). (1) Limitation on the number of textures that can be fetched by the fragment programs: For instance, if the GPU has 16 fragment processors, the maximum number of textures that can be accessed at any one time is limited to 16. (2) Minimum use of textures per cluster to avoid memory constraints: For instance, while computing large number of clusters, if enough care is not taken, the number of textures required for handling cluster membership values will be large. (3) Maximum reuse of shader programs to increase portability.

4 Efficient and Scalable Implementation of FCM on the GPU

In our GPU-based FCM implementation, the various iterative components of the algorithm are executed in the fragment processor using shaders. Textures are memory locations in the GPU, which are used to store the distance and the membership matrices. Multiple dimensions in the incoming data are handled by using partial sum of squared distance computations and stored in the distance textures. All textures use 'Luminance' as internal data format. A speed gain over two orders of magnitude has been achieved for a 79 dimensional yeast gene expression dataset which has about 64k observations. Figure 2 shows the FCM scheme that is implemented on the GPU. The CPU provides the control on the execution of the algorithm; the required number of iterations, control loop branching and the checking of stop condition. The inherent parallelism of the GPU is exploited and used for the iterative computations in the FCM algorithm such as distance computations, membership computations, and computation of cluster centres. The execution steps of FCM in GPU are quite similar to the implementation in CPU. In the next section we discuss the steps involved in our GPU based FCM implementation briefly.

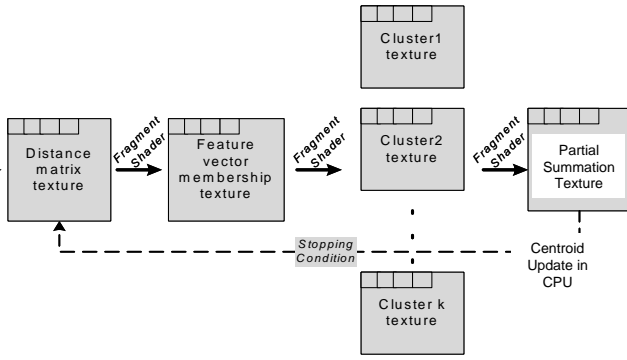


Figure 2: GPU Implementation Scheme of Fuzzy c -means Clustering Algorithm

4.1 GPU based FCM Functions

Parts of the algorithm are computed in a way that the parallelism of the GPU hardware can be exploited to make it efficient. The major steps in our implementation of the FCM algorithm on the GPU are stated below and further discussed.

1. Create initial membership matrix for all n data observations with respect to each cluster.
2. Initialize the c cluster centers from the n data vectors.
3. Compute sum partial deviations between the c cluster centers and the n data vectors.
4. Compute the ratio between the sum partial deviations of the cluster being compared to each other cluster.
5. Store ratio of sum partial deviations in textures, one per cluster.
6. Compute exponentiation of all the deviation textures.
7. Compute partial memberships for all observations per cluster.
8. Compute the membership values via summation of partial memberships.
9. Transfer the summed membership values to CPU.
10. Compute new cluster centers in the CPU.

The initial membership matrix U_{ij} is randomly generated for all the n observations with respect to each cluster. The initial U_{ij} is made the same for both the GPU and CPU implementations by using the same seed in the random generation of membership values. Initial cluster centres (C_j) are identified. The deviations between these cluster centres and each of the data vectors are computed and summed. The deviations between the cluster centre and data vectors are computed partially. The partial computation of deviations is repeated d times and

summed, where d is the number of dimensions in the data vector. Texture reduction technique is employed for all summations. The ratio of the deviations between each cluster centre being compared with each individual data point and the other deviations as in equation (1) is computed. These ratios are stored in textures one per cluster. After the computations are complete for d dimensions, all these textures are simultaneously raised to the power of $2/(m-1)$. The inverse of the resultant texture will produce the iterated membership texture matrix U_{ij} . Using the membership values the new cluster centres are computed. For this operation, the summation of the product of membership texture and the input data objects and the summations of the membership textures are obtained in the GPU. These summations per cluster per iteration are transferred to the CPU and repeated for d dimensions. So the number of transfers is in the order of $d * c * \text{number of iterations}$. In the CPU the new cluster centres (C_j) are compared with the previous cluster centres (C_k) and the decision is made whether to continue or stop the iteration based on the stopping condition. The error between the current cluster centres and the previous should be less than 0.00001. Table 1 lists the shader programs used to accomplish these computations and the major purpose of each shader program.

4.2 Scalability in the GPU based FCM

In data mining scalability means to take advantage of the existing parallelism and design solutions to solve a wide range of problems without needing to change the underlying implementation. We realize scalability via (1) data representation in the GPU memory, (2) operational flexibility on data dimensions and (3) ability to accommodate data sets with higher dimensions. Data representation is handled by accessing individual dimensions across all data objects. So it is easier to perform computations on huge data sets. Moreover, using our GLSL implementation it is easier to perform various operations on the dimensions. It is also simpler to reconfigure the shaders for higher dimensions, since partial computations are done across the data vectors and large number of clusters, thus being more adaptable and flexible, compared to previous implementations. Most notably, scalability is achieved since there is no necessity to define huge textures and redesign the fragment shader codes, as was the case for the earlier algorithms.

No.	FCM Functions	Function call	Fragment Shaders	Purpose of the steps in GPU based FCM
I	Distance Computations (GPU)	Computation0()	glslProgram0()	Sets initial textures with zeros
			glslProgram1()	Computes the distances; summation of partial distances
II	Calculating the exponential (GPU)	Computation1()	glslProgram4()	Computes the exponentials of the distance deviations in the membership matrix
			glslProgram0()	Sets initial textures with zeros
III	Partial Summations (GPU)	Computation2()	glslProgram2()	Computes partial summation across all textures
			glslProgram3a()	Computes partial memberships based on distance and partial sums
			glslProgram3b()	Multiplies the memberships with coordinates to obtain the membership * cluster member product
			glslProgram3c()	Computes the summation of the membership values
V	Update of new cluster centroids (CPU)	-	-	Computes the new centroids by dividing the membership * cluster member product by the summed membership values to obtain fuzzy centroids

Table 1: Summary of the FCM Steps and the Fragment Shaders used for Computations

5 Experimentations on GPU based FCM

The objective of this experiment is to implement the traditional FCM algorithm on a GPU to form fuzzy clusters. Compare its performance with an equivalent implementation of the same algorithm on a desktop CPU. In both implementations initial membership values were randomly generated and made the same using a common random generated seed. The detailed experimental setup and the evaluation of the results are discussed in the next three sections below. The novelties of our implementation and the challenges are discussed subsequently.

5.1 The Experimental Setup

The algorithm is executed on 2 Nvidia's GPUs; viz. GeForce 8500 GT, which is considered as a mid-range graphics processor, and a GeForce 8800 GTX, which is considered as a high-end graphics processor. The results obtained are compared with that obtained from their corresponding CPU counter parts, which are Pentium4 (D), 3.0 GHz CPU and a Pentium(R), 1.5 GHz CPU respectively. The performance of the GPU on the computations heavily depends on the hardware characteristics and hence the GPU configurations are described in this section. The 8500 GPU has 16 fragment shaders processing texels to pixels at a memory clock rate of 800 MHz and 512MB of video memory. The peak memory bandwidth is 12.8 GB/sec. The 8800 GPU has 128 total stream processors with a memory clock rate of 900 MHz and 512MB of video memory. The peak memory bandwidth is 86.4 GB/sec. The experiments will involve the following:

1. Complete the GPU based FCM iterations until the stopping criterion is satisfied and measure the computational time (GPU processing time + data transfer time) for various combinations of n , d and c . Repeat the same on the corresponding CPU and measure the computational time.
2. Use synthetic data to conduct efficiency studies. The size of data, size of dimensions and the cluster numbers will be varied in order to understand the computational efficiency, and the GPU to CPU data processing time ratio.
3. Use the yeast gene expression data set, which has 79 dimensions with about 65k genes (Arul, Dash and Tue 2008), to compare the performance of both the GPUs over their CPU counterparts.
4. Analyse the data transfer time (CPU to GPU) and computational time (time/iteration).
5. Perform regression on the data summarized from the 8800 GPU studies to understand the effect of the cluster parameters n , d and c on computational efficiency.

From the above experiments we intend to analyse the following performance metrics to compare and understand the benefits and challenges in using GPU for the FCM computations, which can be then generalized to other general-purpose computations.

1. Accuracy of the cluster centres formed by the GPU as compared to the centres from the CPU.
2. Raw computational time (C_r) comparison between the GPU and the CPU.

3. Comparison of the computational efficiency which is obtained from the ratio of (CPU computational time per iteration) / (GPU computational time per iteration).

4. Comparison of the processing speed gain, (P_r ratio) which is the ratio of the (CPU computational time) / (GPU computational time – Data transfer time, T_d)

5. The influence of the FCM factors such as c clusters, input data size n and d dimensions on the GPU computational time per iteration.

5.2 Experimental Results Evaluation

The accuracy of the GPU cluster centres as compared to that from the CPU was determined using Mean Square Error (MSE) between the cluster centres from both the CPU and the GPU. With data sizes more than 65k, dimensions up to 10 and for 4 clusters the MSE was in the order of 10^{-8} to 10^{-11} . This insignificant error is due to the lower floating-point precision in GPU compared to that of the CPU. This insignificant error shows that the clusters formed by the GPUs are reliably accurate.

The results obtained from the GeForce 8800 GTX GPU as compared with the Pentium(R), 1.5 GHz CPU are substantial. In our implementation we show that the computational efficiency (CPU computational time per iteration) / (GPU computational time per iteration) ranging from 20x to 94x times. Note that number of iterations required satisfying the stopping criterion may vary between CPU and GPU because of GPU's limited precision. So, if we consider the total time ignoring the difference in number of iterations, then we get the following equation: overall speed gain = (total time taken by GPU/time taken by CPU) which is almost of the same order: 20x to 95x times based on various combinations of factors such as d , n and c .

The results obtained from the mid-range GeForce 8500 GT GPU as compared with the Pentium4 (D), 3 GHz CPU are also promising. Results show that the computational efficiency ranges from 14x to 43x times and overall speed gain is almost of the same order: 14x to 53x times.

The results of the various experiments are graphically summarized, shown and discussed in the next section. Figure 3 shows the comparison of the computational time (C_r) between the two GPUs and their corresponding CPU counterparts.

5.3 Discussion on the Results

When the data size is small (say, 2048), the GPU seems to be slower or just the CPU is as good as the GPU in computing as seen in Figure 3. This comparison shows the raw computational time taken by both the CPU and the GPU ignoring the number of iterations. It can be seen that as the data size increases, there is tremendous speed gain in the GPU computation. When the data size is small, many parallel processors and memory resources are left unused, but when the data size is large, the utilization of the GPU resources is maximized. The GeForce 8800 GPU is able to complete the tasks of forming 4 fuzzy clusters from 1 million 4 dimensional data objects within 0.91 seconds where as the corresponding CPU could take up to 87.8 seconds. The GeForce 8500 GPU is able to complete the same task in 6 seconds when its CPU takes 313.8

seconds to complete the task. It can also be noted that almost 77% of computational time (C_t) of GPU is taken up by the data transfer time (T_t) from the GPU to the CPU. Figure 4 shows the comparison of the computational efficiency between the two GPUs. The processing time ratio is also compared to show how fast it is to process data within the GPU.

As the data size increases, the computational efficiency of the GPUs in implementing the FCM algorithm increases ranging from 20x to 94x times. The computational efficiency takes the GPU to CPU data transfer time and the actual processing time into account. To have a fair comparison with the implementation in (Anderson, Luke and Keller 2007) we also compare the processing time ratio from the two GPUs. It can be noticed in Figure 4 that depending on the GPU used, the processing time can be as fast 924x times when the data size is over 1 million. Figure 5 compares the computational efficiency and the processing time ratios of the GPUs in implementing the FCM algorithm for various sizes of clusters and dimensions. This comparison shows that as the size of the dimensions increase, the performance of the GPUs drop slightly, still being about 19x to 31x times faster than the CPUs. This drop is attributed to the scheme of our implementation, where we minimize the use of distance textures for any size of dimensions. By doing so, the implementation becomes more generic and scalable to any number of dimensions. We also used the complete set of yeast gene expression data, which had 79 dimensions and 65k observations, and the results are summarized in Table 2. The P_t ratio which compares the CPU to GPU processing times shows that our implementation outperforms the results from the previous implementation (Anderson, Luke and Keller 2007), for both low and high dimensional data.

Cluster Size	GeForce 8800 GTX		GeForce 8500 GT	
	Efficiency	P_t Ratio	Efficiency	P_t Ratio
4	19.5	112.5	12.5	24.4
8	20.8	129.4	26.8	60.6
16	21.7	141.8	20.7	41.2
32	21.9	137.9	34.2	73.1
64	23.5	131.0	22.1	35.2

Table 2: Comparison of Computational Efficiency and Processing Time between the Two GPUs

5.4 Novelties in our GPU-FCM Implementation and Comparisons

The following are the novelties in our FCM implementation:

1. For any size of d , we use only two input textures for data transfer and distance computations. So the number of texture sizes depends only on the data size and not on the dimensions. Thus the implementation is scalable and there is no need to expand the number of textures and change the fragment shader codes due to increase in d .

In the previous implementation (Anderson, Luke and Keller 2007), individual textures are used to pack the data sets thus limiting the number of dimensions allowed in a texture, while increasing the number of data objects. In this scheme, computation on any data object would require to access a number of textures. For instance, if there are 64

dimensions then in the previous implementation they would divide the dimensions into groups of 4 each, perform partial computations on each group followed by the final computation on these partial computations. But in our scheme, we store each data object in its entirety in a single texture, thus avoiding the extra computation required to perform the final computation from partial computations. The fragment program for such a scheme is complex.

2. During membership computation in FCM effectively only one cluster centre is involved at a time, so there is no need to maintain a huge membership texture of size $c * n$. In our implementation we use an $\sqrt{n} * \sqrt{n}$ texture per cluster centre per membership computation and we repeat this c times. It helps in better management of GPU memory (textures).

3. In the membership computation step, the summations of the ratio of deviations of each data point to the cluster centre and the deviation of each data point to the previous cluster centre are stored in an $\sqrt{n} * \sqrt{n}$ texture per cluster centre. The novelty is that all the c textures are simultaneously raised to the power of $2/(m-1)$. This helps in speeding up of the computations in the GPU.

5.5 Research Challenges

In our implementation of the FCM in GPU, we find the following research challenges:

1. The fuzzy cluster partial sums are transferred from the GPU to the CPU and the fuzzy centroids are updated in the CPU. This GPU to CPU data transfer time is about 70 to 77% of the total computational time taken by the GPU, which is a potential issue for further research and shader optimization.

2. The computational efficiency varies with the data size- n , number of dimensions- d and number of clusters- c . We intend to determine which of these factors influence the GPU efficiency the most, so that textures and shaders could be rearranged to maximize efficiency. From the studies conducted so far using GeForce 8800 GTX, we use regression analysis to identify the significance and contribution of each of these factors to the computational efficiency of GPU. Regression analysis showed that all the four factors including the intercept are significant with 95% confidence and the R^2_{adjusted} is 97.4%. The coefficients of the factors are plotted in Figure 6 for relative comparison.

From Figure 6 it can be noted that the dimension size d and the cluster size c have more influence on the GPU efficiency. The positive coefficients denote that as d and c in a given data set increases, the GPU is expected to be more efficient in performing the computation.

3. The use of very large sets of data may pose a limitation on execution speed depending on the size of the graphical memory. The speed also depends on the graphics processor hardware being used. The Nvidia 8800 used in our implementation supports textures of sizes up to 8192 x 8192; which means data sizes as big as 67 million could be handled. More care should be taken while implementing shaders to handle huge data sets which may exceed the size of the textures, which can be further explored. It is also vital to note the number of textures that could be simultaneously used by the fragment shader depends on

the number of independent stream processors available in the GPU used.

These issues will lead us into the next stage of research to identify and generalise suitable implementation architectures for FCM and other clustering problems. The

optimisation of FCM by selecting optimal number of clusters using standard FCM performance indices is one such application. Experiments will be conducted to explore the limitations posed by the GPU fragment processors and size of textures vs. the size of input data.

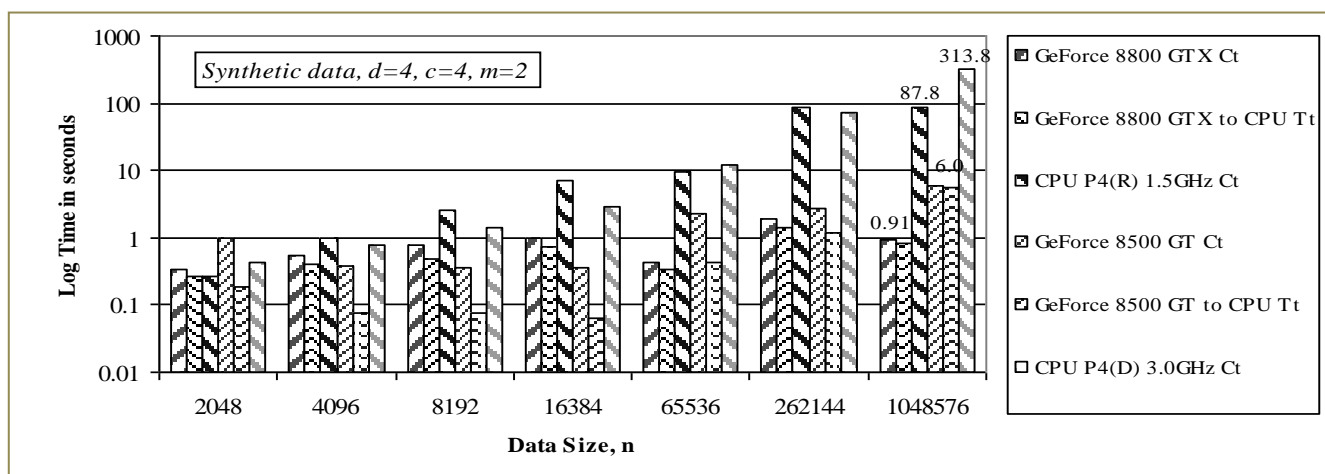


Figure 3: Comparison of Raw Computational Time between the GPU and CPU in Implementing FCM

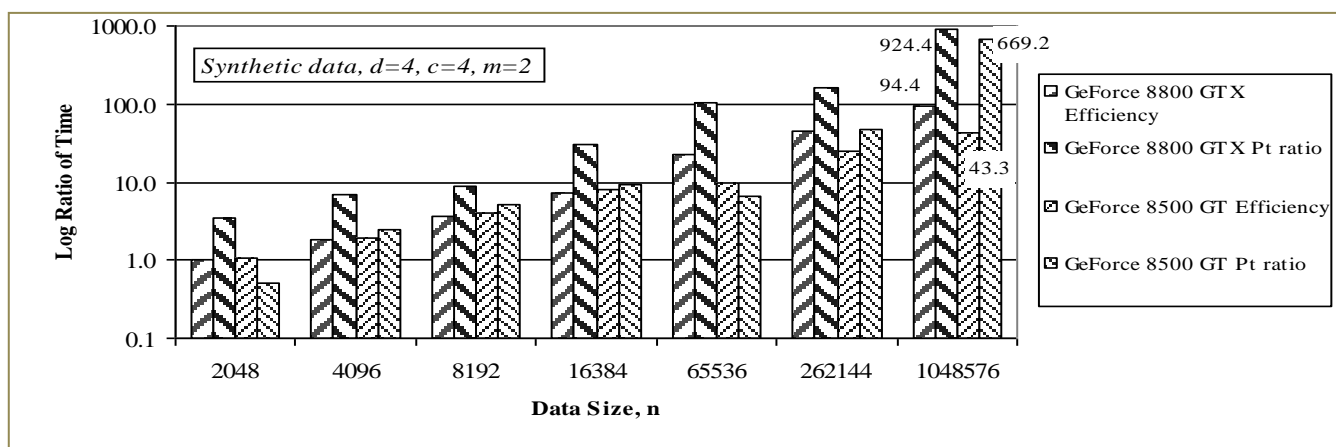


Figure 4: Comparison of Computational Efficiency and Processing Time between the two GPUs used in Implementing FCM

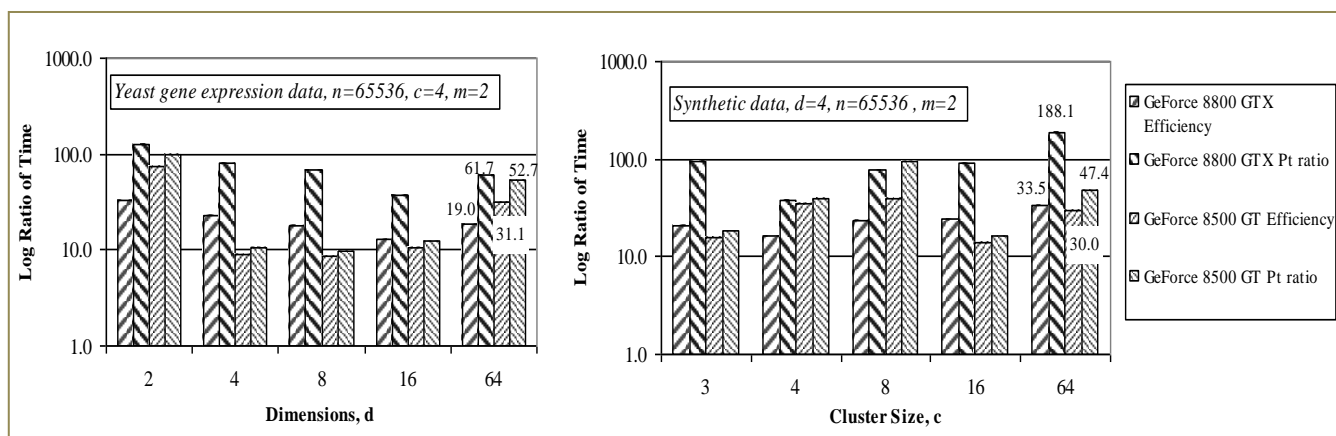


Figure 5: Comparison of Computational Efficiency and Processing Time between the two GPUs used in implementing FCM with Various Dimension Sizes and Cluster Sizes.

6 Conclusion and Future Extendibility

The scheme we have devised for implementing the FCM algorithm in GPU is giving very significant gains over its counterpart CPU. Speed gains up to 140x times on GPU 8800GTX and up to 73x times on GPU 8500GT is realized. In our analysis with the earlier implementations we found that for FCM in high dimensional large data sets, many factors need to be considered very carefully to improve GPU effectiveness. For instance, if the number of dimensions is large then it benefits by keeping all the dimensions in one texture rather than splitting it into many textures. We are able to handle any number of dimensions and clusters without the need for defining new textures and change of fragment programs. Thus we make the implementation scalable.

We effectively use the GPU resources for membership computations by exponentiation all the textures per cluster centre simultaneously by using a single execution of the shader. This helped in performance boosting. Using our implementation scheme any distance metrics such as Manhattan and other non-Euclidean distances can be implemented easily in our program, without the need to change other fragment programs which do involve in distance computations.

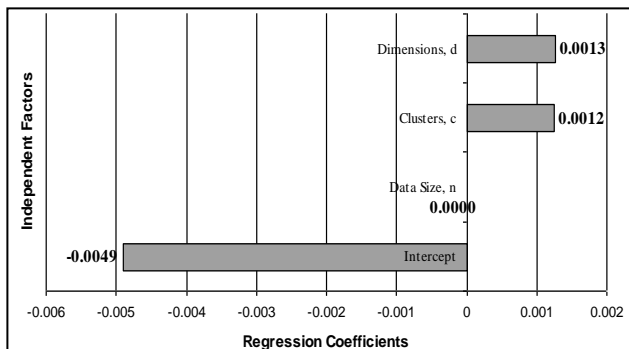


Figure 6: FCM Factors Influencing Computational Efficiency of GPU

Transfer of cluster summation data in iterations to the CPU for computing the new cluster centres takes about 77% of the total time while performing the computation in the GPU. Instead, this computation can also be implemented in the GPU so as to reduce this heavy overhead due to data transfer, especially when handling high dimensions and possibly large number of clusters.

The shader programs we have used for this implementation of FCM are not optimised for the latest GeForce 8800 GTX and beyond. The decoupling of mathematical operations and the texture operations of the latest GPU architectures will be utilized to further improve the efficiency, leveraging on CUDA. Performance index computations on GPU to identify optimal number of fuzzy clusters will also be investigated.

7 References

- Jain, A.K., Murty M.N., and Flynn P.J. (1999): Data Clustering: A Review, *ACM Computing Surveys*, Vol 31, No. 3, 264-323.
- MacQueen, J. B. (1967): Some Methods for classification and Analysis of Multivariate Observations, In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, University of California Press, 1:281-297.
- Guha, S., Rastogi, R., and Shim, K. (1998): CURE: An efficient clustering algorithm for large databases. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 73--84, New York.
- Ester, M., Kriegel, H., Sander, J., Xu, X., (1996): A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, In *Proceedings of 2nd International Conference on KDD*. AAAI Press.
- Bezdek, J. C. (1981): *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York.
- Free Encyclopaedia: GNU Free Documentation, Wiki Software, <http://www.wikipedia.org/>. Accessed 20 May 2008.
- Owens, J. D., Luebke, D., Govindaraju, N., Harris, M., Krüger, J., Lefohn, A. E., Purcell, T. J. A (2005): Survey of General-Purpose Computation on Graphics Hardware. *Eurographics*.
- Fatahalian, K., Sugerman, J., Hanrahan, P. (2004): Understanding the efficiency of GPU algorithms for matrix-matrix multiplication, In *Proceedings of the ACM SIGGRAPH/ Eurographics conference on Graphics hardware*.
- Arul, S., Dash, M., and Tue, M. (2008): GPU-based fast *k*-means clustering of gene expression profiles", In *Proceedings of 12th Annual International Conference on Research in Computational Molecular Biology (RECOMB)*. Singapore.
- Arul, S., Dash, M., and Tue, M. (2008): Efficient *K*-means Clustering Using Accelerated Graphics Processors, Accepted for *International Conference on Data Warehousing and Knowledge Discovery (DAWAK)*.
- NVIDIA: GeForce 8800 Architecture Technical Brief, http://static.tigerdirect.com/pdf/NVIDIA_GeForce8800_GPU_Architecture_Technical_Brief.pdf. Accessed 12 January 2008.
- Harris, C. Haines, K. (2005): Iterative Solutions using Programmable Graphics Processing Units, In *Proceedings of the 14th IEEE International Conference on Fuzzy Systems*, pages: 12- 18.
- Anderson, D., Luke, R. H., Keller, J. M. (2007): Incorporation of Non-Euclidean Distance Metrics into Fuzzy Clustering on Graphics Processing Units, *Analysis and Design of Intelligent Systems using Soft Computing Techniques*.

Indoor Location Prediction Using Multiple Wireless Received Signal Strengths

Kha Tran, Dinh Phung, Brett Adams, Svetha Venkatesh

Department of Computing
Curtin University of Technology,
GPO Box U 1987, Perth, WA, Australia,
{k.tran@postgrad,d.phung,b.adams,s.venkatesh}@curtin.edu.au

Abstract

This paper presents a framework for indoor location prediction system using multiple wireless signals available freely in public or office spaces. We first propose an abstract architectural design for the system, outlining its key components and their functionalities. Different from existing works, such as robot indoor localization which requires as precise localization as possible, our work focuses on a higher grain: location prediction. Such a problem has a great implication in context-aware systems such as indoor navigation or smart self-managed mobile devices (e.g., battery management). Central to these systems is an effective method to perform location prediction under different constraints such as dealing with multiple wireless sources, effects of human body heats or mobility of the users. To this end, the second part of this paper presents a comparative and comprehensive study on different choices for modeling signals strengths and prediction methods under different condition settings. The results show that with simple, but effective modeling method, almost perfect prediction accuracy can be achieved in the static environment, and up to 85% in the presence of human movements. Finally, adopting the proposed framework we outline a fully developed system, named Marauder, that support user interface interaction and real-time voice-enabled location prediction.

Keywords: Indoor positioning, WiFi signal, Naive Bayes, Hidden Naive Bayes, indoor navigation.

1 Introduction

The increasing number of mobile devices has called for a new framework to exploit mobile computing power and to support more intelligent information services. To this end, context-aware applications that model information from users and their surrounding environments have been developed to provide value-added services. Information about context is multi-dimensional: positioning data, proximate people, communication and utility usage. Outdoor positioning is more or less a solved problem for devices equipped with GPS receivers. Indoor positioning, however, offering a myriad potential applications in indoor navigation and social pattern extraction, remains an open research problem, and is our focus in this study.

Copyright ©2008, Australian Computer Society, Inc. This paper appeared at the Seventh Australasian Data Mining Conference (AusDM 2008), Glenelg, Australia. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 87, John F. Roddick, Jiuyong Li, Peter Christen and Paul Kennedy, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

There are two main approaches to solving the indoor positioning problem: (i) installation of specialized indoor positioning systems, and (ii) use of existing radio-frequency infrastructures such as GSM, 802.11 and Bluetooth. Methods in the first category have high accuracy, but are expensive and unsuitable for large scale deployment. Methods using the latter approach are more economical, but suffer from signal instability and noise due to hardware characteristics, exacerbated by environmental factors, such as people in motion. We will focus on methods using 802.11 infrastructures. At the early, RADAR (Bahl & Padmanabhan 2000) applies the Nearest Neighbor algorithm to estimate location but a poor performance is obtained because it could not cover the nature of the variance of WiFi signals. Current approaches (Roos et al. 2002, Ladd et al. 2002, Krumm & Horvitz 2004, Xiang et al. 2004) get a better performance by viewing the problem in terms of probabilistic model which is well dealing with the uncertainty. In these probabilistic approaches Bayes' rule is used for prediction and WiFi signals is in different form such as histogram (Youssef et al. 2003) and smoothed histogram (Roos et al. 2002), exponential functions (Xiang et al. 2004) and Gaussian (Kaemarungsi 2005). However, applying probabilistic model in recent works is empirical and there is no systematical investigation in terms of parameter estimation, prediction model selection as well as experiment environment. We will cast them as cases of Naive Bayes and discuss more in an unified framework. Furthermore, all recent approaches are fully-supervised and therefore the degree of calibration required is also a limiting factor to usability.

Motivated by the potential usefulness of indoor positioning systems to an array of applications, such as navigation of office workspaces, we desire a system generic enough to leverage existing WiFi access points found in an urban environment. Importantly, we examine the practical case where both the training and testing signals are acquired in a mobile fashion. We implement and compare two probabilistic models under a set of different conditions: Naive Bayes, where the signal at each WiFi access point is considered to be independent, and the Hidden Naive Bayes (Zhang et al. 2005), which models the joint relationship among the WiFi access point signals to estimate location by embedding the physical proximity of access points in an environment. We also make use of a Boolean adjacency matrix to impose constraints among moving paths. We perform experiments in different scenarios, including where the wireless device is fixed and in motion, both with and without the presence of humans. Our results demonstrate these models can be potentially deployed in complex environments by design and implementation of the real indoor positioning framework and an applications upon this framework.

The significance of this work is in using available

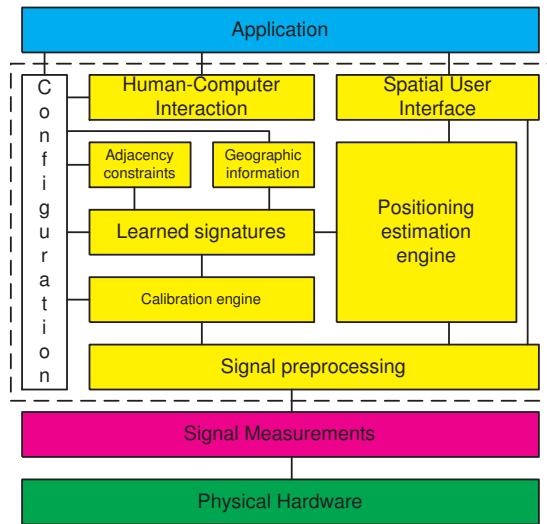


Figure 1: The architecture of Marauder.NET

low-cost infrastructures for location detection in a robust fashion. Importantly, as good performance is obtained for the case where both training and testing data is acquired in a mobile fashion, the model is suitable for general use in urban spaces, and in particular, for fine-grain indoor positioning for the visually impaired.

The layout of the remainder of the paper is as follows. The framework and its principal components are introduced in Section 2. Section 3 discusses about experiments and results. A indoor navigator prototype is demonstrated in Section 4. Session 5 provides a concluding summary.

2 Architecture

We first briefly outline each module in the proposed framework and then discuss in detail two principle components, namely database of learned signatures (signature representation) and positioning estimation engine (prediction model).

2.1 Proposed framework

Figure 1 outlines the architecture of the proposed framework in which the higher the layer, the more abstract the module. Layer Application sits on the top of schema with built-in indoor positioning functions is designed for user-oriental application such as indoor navigator, blind assistant, etc.. The two lowest modules of WiFi hardware and measurement are widely available in the market where most of WiFi adapter is integrated in recent wearable devices such as notebooks and smartphones and its software drivers including signal measurement freely provided to popular operating systems such as PlaceLab¹ and OpenNetCF². The heart of this system is the core engine inside the dashed rectangle which separates into several sub-modules and their relationships are represented as lines between components.

2.1.1 Signal pre processing

There are time-based techniques used to collected and bundled WiFi signals as a collection such as non-overlap window and overlap window (Figure 2). Normally, the window size is in order of seconds for daily office activities.

¹www.placelab.org

²www.opennetcf.com

2.1.2 Calibration Engine

Our system is supervised so that the system requires training data collected in the calibration stage. The steps to collect the data at one location is very simple: user with mobile device is standing at that location and recording the WiFi signals for a given time interval. We introduce three approaches to labeling locations of text, voice and map-click in which two first approaches are positionless (voice is suitable for people with blind while text is absolutely simple and can be automatically transferred to voice using available text-to-speech frameworks) and the last one is supporting the offset coordinates with related to provided partial vector/raster maps. Moreover, the process of calibration can be done incrementally and help the system more flexible and updated.

2.1.3 Trained Signatures

This database is the product of discussed calibration engine. One signature, which is represented for each location, consists of a set of W distributions of signal strengths of W access points and a distribution representing the number of appearance of W access points received at this location. Moreover, weak access points with infrequent number of appearance are also detected and eliminated out of final signatures. It requires mechanism to optimal organize and structure those signature in this database when the number of location is large. We propose a simple method of partitioning the whole database into cluster using access point MAC and geographic relationship. While access points MAC is available in signature and can be computed efficiently, the information of geographic relationship needs to be imported from user and service providers.

2.1.4 Geographic Information

This optional module takes the constraints among physical construction components in urban workspaces such as buildings, levels, sections and areas covered by access points. From that, the large number of locations is partitioned into sub-groups which reduce query processing time from estimation engine. This geographic information is usually stable and could be easily collected by user or service provider.

2.1.5 Adjacency Constraints

We introduce an optional module to keep a set of neighbor locations for particular location for faster retrieval. They are logical constraints that user can only move from a location to its neighboring locations. Once current location is known with a high probability, movement is constrained by topology around a given location, and hence only neighboring locations need be considered. This Boolean adjacency matrix is taken into account in our experiments.

2.1.6 Positioning Estimation Engine

Given a set of access points and their signal strengths, the estimation engine will query the a set of locations, calculate the posterior probabilities and the location with the highest probability is returned as predicted location.

2.1.7 Spatial User Interface

Its roles are for receiving the requests from utilities in layer Application and returning the corresponding location from estimation engine.

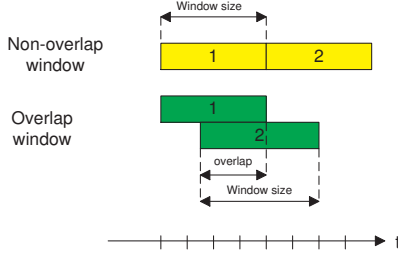


Figure 2: Non-overlap window and overlap window.

2.1.8 Human-Computer Interaction

Besides indoor location returning, Marauder framework also provides rich-informative meta-data warehouse such as vector/raster maps as background and voice guidance library which provides more relaxing for higher layers of application. While background map helps to provide more fancy and friendly to normal enduser, voice function relaxes users out of the device's monitor. Moreover, every piece of voice information about around context is trivial for normal people but is significantly meaningful to disable one so that this module aims to provide superior support to the blind.

2.2 Signature representation

Given a particular location, observed WiFi signals consist of the WiFi access point identifiers (MAC address) and corresponding received signal strength (RSS). We define the *location signature* as distributions of signal strengths over a finite set of access points received at that location. Precisely, the location signature consists of a set of W distributions of signal strengths over W access points and a multinomial distribution representing the number of appearances of these W access points at this location.

While the frequency of appearance of W access points is often modeled as discrete distribution of size W , there are different methods to model the signal strengths over each access points and the chosen method can affect the prediction accuracy significantly. Figure 3 shows the plot of measured WiFi signals of one access points in 5-minute interval at a particular location when mobile device is hold stay still at a position. The blue bar and red curve show the empirical histogram and the estimated Gaussian distribution respectively.

Three methods of modeling, namely histogram, smoothed histogram with kernel Gaussian function and Gaussian are investigated in this works. With a small bin of 1dBm, the signal strength is discrete into $V = 100$ values from -100dBm to 0dBm and counts over all received signals. The histogram signature is the distribution of V normalized values. Kernel Gaussian function $\mathcal{K}(y) = \frac{1}{2\pi\sigma_k^2} \exp(-\frac{(y-\mu_k)^2}{2\sigma_k^2})$ where (μ_k, σ_k^2) is kernel parameters and y is signal strength, is introduced to smooth the histogram. The number of parameters needs for storing a signature in histogram as well as smoothed histogram distributions are the same and equal to $W(1 + V)$ parameters if the pin is 1dBm. Gaussian distribution captures the signals with just two parameters, mean and variance.

2.3 Prediction model and signature parameter estimation

Location prediction in our work is cast as a classification problem. Most previous works has used the Naive Bayes (NB) which the critical assumption is the

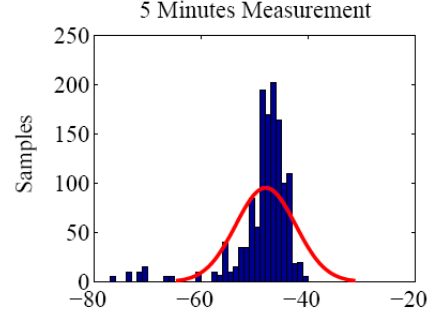


Figure 3: The variance of RSS at a investigated location.

independence of received signals among access points conditionally on the current location. One model to deal with correlation among attributes is Hidden Naive Bayes (HNB) (Zhang et al. 2005). It creates a hidden parent node for each attribute node, capturing the influence from other nodes. Below we briefly outline both the NB and HNB.

Let $C \in \{1, \dots, K\}$ be the location random variable where K is number of locations, $X_m \in \{1, \dots, W\}$ represents the m -th access point random variable, $Y_m \in \{1, \dots, V\}$ represent the signal strength corresponding to m -th access point where W is number of access points, M is number of access points of an observation and V is number of discrete values of signal strength. Signature parameters are estimated from a set of training data of N observations $D = \{\mathbf{o}_1, \dots, \mathbf{o}_N\}$ where $\mathbf{o}_n = (c^{(n)}, x_1^{(n)}, y_1^{(n)}, \dots, x_M^{(n)}, y_M^{(n)})$, $n = 1, \dots, N$. In the prediction phase, the predicted location c^* is inferred based on current observation $\mathbf{o} = (x_1, y_1, \dots, x_M, y_M)$.

2.3.1 Naive Bayes

Figure 4 shows the NB. The joint distribution $P(C, X_1, Y_1, \dots, X_M, Y_M)$ is given by:

$$P(C) \prod_{m=1}^M P(X_m|C)P(Y_m|C, X_m)$$

where the distribution of access point x given a location c , $P(X_m = x|C = c)$ is multinomial (W -size parameter π_c), the probability of signal strength y given location c and access point x , $P(Y_m = y|C = c, X_m = x)$, is normalized histogram (V -size vector parameter $\gamma_{c,x}$), smoothed histogram (V -size vector parameter $\eta_{c,x}$), Gaussian (two parameters $\mu_{c,x}$ and $\sigma_{c,x}^2$) respectively. Without any prior knowledge about the current location c , the distribution $P(C = c)$ could be assigned as uniform.

Let the identify function $\mathbb{I}(a, b) = 1$ if $a = b$ else $= 0$, in maximum likelihood estimation framework, the sufficient statistics are:

$$n_c = \sum_{n=1}^N \sum_{m=1}^M \mathbb{I}(c^{(n)}, c)$$

$$n_{c,x}^{(y)} = \sum_{n=1}^N \sum_{m=1}^M \mathbb{I}(c^{(n)}, c) \mathbb{I}(x_m^{(n)}, x) \mathbb{I}(y_m^{(n)}, y)$$

$$n_c^{(x)} = \sum_{n=1}^N \sum_{m=1}^M \mathbb{I}(c^{(n)}, c) \mathbb{I}(x_m^{(n)}, x)$$

The parameters of $P(X_m|C)$ are estimated as

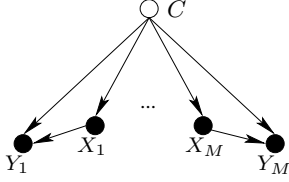


Figure 4: Naive Bayes model.

$$\hat{\pi}_c^{(x)} = \frac{n_c^{(x)} + 1}{n_c + W}$$

The parameters of $P(Y_m|C, X_m)$ are differently estimated according to three methods of representation. In histogram case, the parameters are as follows

$$\hat{\gamma}_{c,x}^{(y)} = \frac{n_{c,x}^{(y)} + 1}{n_c^{(x)} + V}$$

In smoothed case, the parameters are estimated as:

$$\hat{\eta}_{c,x}^{(y)} = \frac{m_{c,x}^{(y)}}{\sum_{y=1}^V m_{c,x}^{(y)}}$$

where

$$m_{c,x}^{(y)} = \sum_{n=1}^N \sum_{m=1}^M \mathbb{I}(c^{(n)}, c) \mathbb{I}(x_m^{(n)}, x) \mathcal{K}(y - y_m^{(n)})$$

In the Gaussian case, the mean and variance are

$$\begin{aligned} \hat{\mu}_{c,x} &= \frac{m_{c,x}}{n_{c,x}} \\ \hat{\sigma}_{c,x}^2 &= \frac{m_{c,x}^2}{n_{c,x}} \end{aligned}$$

where

$$m_{c,x} = \sum_{n=1}^N \sum_{m=1}^M \mathbb{I}(c^{(n)}, c) \mathbb{I}(x_m^{(n)}, x) y_m^{(n)}$$

and

$$m_{c,x}^2 = \sum_{n=1}^N \sum_{m=1}^M \mathbb{I}(c^{(n)}, c) \mathbb{I}(x_m^{(n)}, x) (y_m^{(n)} - \mu_{c,x})^2$$

At the prediction step, the location is found by finding the location having the highest likelihood:

$$\begin{aligned} c^* &\propto \arg \max_{c \in \{1, \dots, K\}} P(\mathbf{o}|c) P(c) \\ &= \arg \max_{c \in \{1, \dots, K\}} P(c) \prod_{m=1}^M P(x_m|c) P(y_m|c, x_m) \end{aligned}$$

where Bayes' rule is used.

2.3.2 Hidden Naive Bayes

HNB relaxes the independent assumption in the NB by letting attributes depends on each other. In our case the HNB approximates the full correlation of access points by creating a hidden parent variable H_m for each variable Y_m and then linearly simplifies the

conditional probabilities. Figure 5.a and 5.b show fully connected node Y_m and its HNB approximation. The joint distribution $P(C, X_1, Y_1, \dots, X_M, Y_M)$ is defined as:

$$P(C) \prod_{m=1}^M P(X_m|C) P(Y_m|Y_{-m}, X_m, C)$$

where distribution of access point x_m given location c $P(X_m = x|C = c)$ is multinomial and the distribution of signal strength y_m of access point x_m given location c and a set of signal strengths $Y_{-m} = \{Y_1 = y_1, \dots, Y_{m-1} = y_{m-1}, Y_{m+1} = y_{m+1}, \dots, Y_M\}$ of other access points $P(Y_m = y_m|Y_{-m}, X_m = x_m, C = c)$ is also a multinomial.

In (Zhang et al. 2005), $P(Y_m|Y_{-m}, X_m, C)$ is represented by $P(Y_m|H_m, X_m, C)$ and is formulated as:

$$\sum_{j=1, j \neq m}^M w_{x_m, x_j|c} P(Y_m|Y_j, X_j, X_m, C)$$

where $\sum_{m=1, j \neq i}^M w_{x_m, x_j|c} = 1$.

The weight $w_{x_i x_j|c}$ of two access points x_i and x_j conditional on location c is defined in (Zhang et al. 2005):

$$w_{x_i, x_j|c} = \frac{I_P(x_i, x_j|c)}{\sum_{j=1, j \neq i}^M I_P(x_i, x_j|c)}$$

where $I_P(x_i, x_j|c)$ is the conditional mutual information

$$I_P(x_i, x_j|c) = H(x_i|c) + H(x_j|c) - H(x_i, x_j|c)$$

and $H(x_i|c)$ is the entropy of access point x_i and $H(x_i, x_j|c)$ is the joint entropy of x_i and x_j :

$$H(x_i|c) = - \sum_{y_i=1}^V P(y_i|x_i, c) \log P(y_i|x_i, c)$$

and

$$\begin{aligned} H(x_i, x_j|c) &= - \sum_{y_i=1}^V \sum_{y_j=1}^V P(y_i, y_j|x_i, x_j, c) \\ &\quad \log P(y_i, y_j|x_i, x_j, c) \end{aligned}$$

Defining the all distributions in HNB as multinomial where the parameters of $P(y_i|y_j, x_j, x_i, c)$ is V -size vector $\tau_{x_i|x_j, y_j, c}$, the parameter of $P(x_i|c)$ is W -size vector π_c , the parameter of $P(y_i|x_i, c)$ is V -size vector $\gamma_{x_i|c}$ and the parameter of $P(y_i, y_j|x_i, x_j, c)$ is $V \times V$ -dimension matrix $\Phi_{x_i, x_j|c}$.

The sufficient statistics in this case are:

$$n_c = \sum_{n=1}^N \mathbb{I}(c^{(n)}, c)$$

$$n_c^{(x_i)} = \sum_{n=1}^N \sum_{m=1}^M \mathbb{I}(c^{(n)}, c) \mathbb{I}(x_m^{(n)}, x_i)$$

$$n_{c, x_i}^{(y_i)} = \sum_{n=1}^N \sum_{m=1}^M \mathbb{I}(c^{(n)}, c) \mathbb{I}(x_m^{(n)}, x_i) \mathbb{I}(y_m^{(n)}, y_i)$$

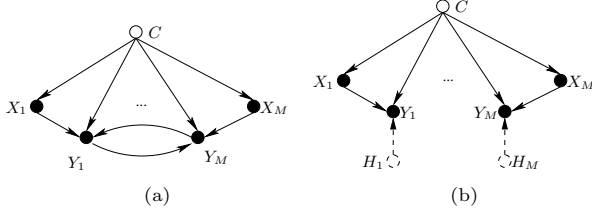


Figure 5: (a) Model when received signals are fully dependent and (b) its approximation, the HNB.

$$n_{c,x_i,x_j}^{(y_i,y_j)} = \sum_{n=1}^N \sum_{m=1}^M \sum_{l=1, l \neq m}^M \mathbb{I}(c^{(n)}, c) \mathbb{I}(x_m^{(n)}, x_i) \mathbb{I}(y_m^{(n)}, y_i) \mathbb{I}(x_l^{(n)}, x_j) \mathbb{I}(y_l^{(n)}, y_j)$$

The parameters of $P(y_i|x_i, c)$, $P(y_i, y_j|x_i, x_j, c)$ and $P(y_i|y_j, x_j, x_i, c)$ are estimated as follows (Zhang et al. 2005):

$$\begin{aligned} \hat{n}_c^{(x_i)} &= \frac{n_c^{(x_i)} + 1}{n_c + W} \\ \hat{\gamma}_{x_i|c}^{(y_i)} &= \frac{n_{c,x_i}^{(y_i)} + 1}{n_c + V} \\ \hat{\Phi}_{x_i,x_j|c}^{(y_i,y_j)} &= \frac{n_{c,x_i,x_j}^{(y_i,y_j)} + 1}{n_c + V^2} \\ \hat{\gamma}_{x_i|x_j,y_j,c}^{(y_i)} &= \frac{n_{c,x_i,x_j}^{(y_i,y_j)} + 1}{n_{c,x_j}^{(y_j)} + V} \end{aligned}$$

Similar to NB model, at the classification step, the location is found by finding the location having the highest likelihood:

$$\begin{aligned} c^* &\propto \arg \max_{c \in \{1, \dots, K\}} P(c|o)P(c) \\ &= \arg \max_{c \in \{1, \dots, K\}} P(c) \prod_{i=1}^M P(x_i|c) \\ &\quad \sum_{j=1, j \neq i}^M w_{x_i,x_j|c} P(y_i|y_j, x_j, x_i, c) \end{aligned}$$

Again, without any prior knowledge about current location, the probability $P(c)$ is assigned to uniform distribution and have no effect during classification step.

3 Experiments

We conducted experiments comparing the NB with the HNB. In the case of NB, we consider three cases wherein the RSS is represented as a histogram (Model I, NB+H), smoothed histogram using a kernel Gaussian function (Model II, NB+K), and the Gaussian (Model III, NB+G). In the case of HNB, the RSS is represented by a histogram (Model IV, HNB+H). In order to investigate realistic settings, three environments were defined: A—no humans present, B—humans present but not moving, and C—humans moving during testing and training. Investigated results

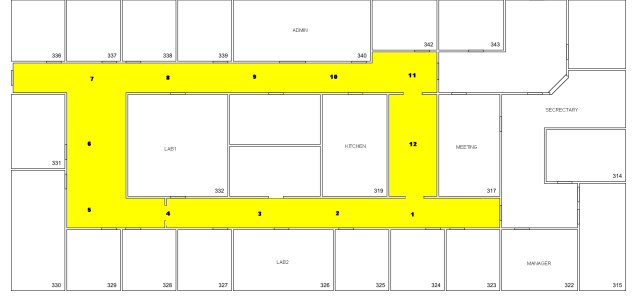


Figure 6: Layout of office space used in the experiments.

illustrate that the system performance will be significantly affected with human presence and especially human in moving (Xiang et al. 2004). RSS is processed using both with and without overlap window. The system is predicted in a time slot of every 2 seconds. The non-overlap window is 2s while overlap window size is 4 seconds with 2 second overlap.

The system was set up in a corridor area whose layout is depicted in Figure 6 (corridor is indicated in yellow). The building is equipped with an IEEE 802.11b wireless network with 2.4 GHz frequency bandwidth consisting of three Cisco Aironet 1200 Series access points. The calibration and testing program was run on a Sony Vaio VGN-UX17GP under Windows XP with a built-in wireless card (Intel(R) PRO/Wireless 3945ABG). We modeled the environment as 12 locations (1-12) with the distance between two neighboring locations being 4 meters. For each scenario described above, training data was collected for each location in 5 minutes intervals with approximately 300 observations. The calibration data $12 \text{ locations} \times 15 \text{ minutes} \times 3 \text{ environments} = 9 \text{ hours}$ is randomly divided to 3 parts, 1 for training and 2 testing.

The system performance is evaluated by using two measures of accuracy (meters) and precision (percentage) adapted in (Liu et al. 2007). While the accuracy is predefined according to the calibration data, the precision is the distribution of distance error between the estimated location and the true location. The data collected for each location belong to a region with the radius of $2m$, therefore the precision with accuracy $2m$ is the recall rate of which is measured as the ratio of estimated location and ground truths.

Table 1 shows the precision for four models in twelve scenarios. Overall, model NB+H is marginally better compared to the other models, especially in noisy environments. The rate decreases gradually as noise is introduced as a result of allowing moving humans and objects in the environment, and increases when tuning techniques are integrated.

In terms of tuning techniques, overlap window yields improved results of approximately 10% in scenarios where humans are moving. While the use of an adjacency matrix improves only 2% in performance, it does reduce computation time considerably in case of large-scale environment because of its role of clustering.

Surprisingly, the HNB with more complex and computational model, does not demonstrate superior performance compared to simpler models. On the other hand, although obtained performance is slightly lower, model NB+G shows potential opportunity to be deployed as large-scale system in real environment because of its useful characteristics such as compressed signature and simple prediction engine.

Table 1: Precision rate (%) when accuracy is 2 meters.

Environment	A (no humans)				B (humans static)				C (humans moving)			
Window(overlap)	2s		4s(2s)		2s		4s(2s)		2s		4s(2s)	
Adjacency constraint	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes
I (NB+H)	99.09	99.19	99.34	99.39	93.25	93.92	96.88	97.10	74.22	77.14	85.36	85.64
II (NB+K)	99.19	99.29	99.14	99.14	92.06	93.84	95.77	96.07	70.16	72.89	79.98	80.54
III (NB+G)	95.22	95.62	96.83	96.89	90.88	92.14	94.51	94.81	67.51	68.90	77.43	78.73
IV (HNB+H)	99.39	99.19	99.39	99.39	87.69	89.54	92.95	93.03	70.25	72.61	78.94	80.92

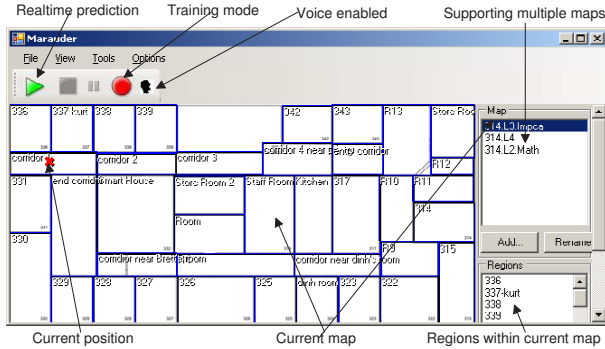


Figure 7: Indoor navigator prototype is developed using Marauder.NET framework.

4 Marauder

Several potential applications can be developed based on our proposed framework such as indoor navigator and blind positioning assistant. Figure 7 shows an indoor navigator application named Marauder, working under Windows XP platform and running in portable device Sony Vaio VGN-UX17GP. This application support end-users to import background maps, setup calibration regions in preparation step and locate where we are in the building. Existing background map is listed in top-right panel where their names represent hierarchical relationship such as building, floor, section so so on. In the bottom-right panel, set of calibrated regions of the current map are easy to adjust/add/remove. The wide center area shows the current map where the red crossing sign tells us where we are in this map. While the green triangle button is enable for realtime location prediction, the red circle button supports for recording the signatures in training phase. To release user out of application monitor or support visual impaired, voice guidance assistant could be triggered with black human button on the toolbar. Besides, there are several other functions such as navigating and zooming (menu View) and prediction engine mode (menu Tools) and managing parameter and reporting (menu Options) built in the menubar on the top of GUI.

5 Conclusion

A framework is proposed to provide indoor positioning capabilities for an array of potential applications such as indoor navigator, visual-impaired assistant or indoor surveillance system. We present a systematic study of two probabilistic models, the Naive Bayes and Hidden Naive Bayes, for positioning classification using WiFi signals. We also have experimented with various methods of modeling signal strengths, histogram, smoothed histogram and Gaussian in sev-

eral different conditions of real environments. The results show that simple Bayesian models can be used to provide a reliable location detection accuracy. The precision is nearly perfect in non-human environment, around 95% while people are still and 85% in moving circumstance. Surprisingly, HNB model only shows same performance in non-human case and slightly less accuracy in most of remaining scenarios.

References

- Bahl, P. & Padmanabhan, V. (2000), RADAR: an in-building RF-based user location and tracking system, in 'Proceedings of The 19th Annual Joint Conference of The IEEE Computer and Communications Societies (INFOCOM)', Vol. 2.
- Kaemarungsi, K. (2005), Design of Indoor Positioning Systems Based on Location Fingerprinting Technique, PhD thesis, University of Pittsburgh.
- Krumm, J. & Horvitz, E. (2004), Locadio: Inferring Motion and Location from Wi-Fi Signal Strengths, in 'Proceedings of International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous)'.
- Ladd, A., Bekris, K., Rudys, A., Kavraki, L. & Wallach, D. (2002), Robotics-Based Location Sensing Using Wireless Ethernet, in 'Proceedings of The 8th ACM International Conference on Mobile Computing and Networking (MOBICOM)'.
- Liu, H., Darabi, H., Banerjee, P. & Liu, J. (2007), 'Survey of Wireless Indoor Positioning Techniques and Systems', *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews* **37**(6), 1067–1080.
- Roos, T., Myllymäki, P., Tirri, H., Misikangas, P. & Sievänen, J. (2002), 'A Probabilistic Approach to WLAN User Location Estimation', *International Journal of Wireless Information Networks* **9**(3), 155–164.
- Xiang, Z., Song, S., Chen, J., Wang, H., Huang, J. & Gao, X. (2004), 'A Wireless LAN-based Indoor Positioning Technology', *IBM Journal of Research and Development* **48**(5/6), 617–626.
- Youssef, M., Agrawala, A. & Udaya Shankar, A. (2003), WLAN location determination via clustering and probability distributions, in 'Proceedings of the 1st IEEE International Conference on Pervasive Computing and Communications (PerCom)', pp. 143–150.
- Zhang, H., Jiang, L. & Su, J. (2005), Hidden Naive Bayes, in 'Proceedings of the 20th National Conference on Artificial Intelligence (AAAI)', pp. 919–924.

Clustering and Classification of Maintenance Logs using Text Data Mining

Brett Edwards Michael Zatorsky Richi Nayak

CRC for Integrated Engineering Asset Management

Faculty of Information Technology

Queensland University of Technology

PO Box 2434, Brisbane 4001, Queensland

`bj.edwards@aanet.com.au`

`miczat@gmail.com`

`r.nayak@qut.edu.au`

Abstract

Spreadsheets applications allow data to be stored with low development overheads, but also with low data quality. Reporting on data from such sources is difficult using traditional techniques. This case study uses text data mining techniques to analyse 12 years of data from dam pump station maintenance logs stored as free text in a spreadsheet application. The goal was to classify the data as scheduled maintenance or unscheduled repair jobs.

Data preparation steps required to transform the data into a format appropriate for text data mining are discussed. The data is then mined by calculating term weights to which clustering techniques are applied. Clustering identified some groups that contained relatively homogeneous types of jobs. Training a classification model to learn the cluster groups allowed those jobs to be identified in unseen data. Yet clustering did not provide a clear overall distinction between scheduled and unscheduled jobs.

With some manual analysis to code a target variable for a subset of the data, classification models were trained to predict the target variable based on text features. This was achieved with a moderate level of accuracy.

Keywords: Text Data Mining.

1 Introduction

Relational databases allow data to be stored in a clean and consistent format that allows reports to be developed using well understood software development techniques. Often though, the skills to design and develop structured databases are not held by or cheaply available to individuals with the need to store and report on data. Where the value of the data is great enough, the individuals will “find a way” even if the techniques used to store the data would not be considered best practice by information technology specialists.

A common result of this situation is the creation of basic spreadsheet applications requiring no software

development experience. Unfortunately such applications rarely enforce data quality standards. Users add information into unstructured free text fields with little consistency between values. Unlike standard database applications, reporting on information in unstructured formats is not a trivial software development task.

This paper presents a case study of reporting on unstructured free text data using text data mining techniques. The aim is to determine if textual features of the data can be used to classify records into structured attributes. If the accuracy of the classification is high enough, then business decisions can be based on data rather than best guesses.

The client in this study maintains and repairs pump stations for dams and weirs; including the pump motors, electrical systems, fire extinguishing systems, air conditioning systems and external buildings and grounds. They recorded fault logs dating from 1994 to 2006 in a Microsoft Excel spreadsheet where all relevant information was kept in free text fields with little data quality controls.

The client's desire was to compare scheduled maintenance work to unscheduled fault repairs. That is, they wanted to distinguish work that is expected and easily budgeted compared to unexpected work that is potentially avoidable. Any further information specifically about the type of component that failed would make the results more useful to the client.

To perform this analysis manually would be labour intensive. A subject matter expert to read through and classify twelve years of data relating to pump stations supporting more than two dozen dams. Instead of performing the analysis record by record, an investigation was started into whether the desired results could be obtained by through clustering and prediction of the free text data.

A sample of data consisting of 842 records from three pump stations was provided for this study. Given the small data set and low data quality, this study became an investigation into what results can be expected when usual standards of data quality and size are not met.

The analysis followed the CRISP-DM model (Chapman et al. 1999). First steps were to define the business issue and evaluate the input data available. The input data was found to be missing a target column, which was then coded by the authors to denote scheduled versus unscheduled jobs for training a classifier. Data was then prepared for data mining by spell checking,

Copyright (c) 2008, Australian Computer Society, Inc. This paper appeared at conference Seventh Australasian Data Mining Conference (AusDM 2008), Glenelg, Australia. Conferences in Research and Practice in Information Technology, Vol. 87. John F. Roddick, Jiuyong Li, Peter Christen and Paul Kennedy, Eds. Reproduction for academic, not-for profit purposes permitted provided this text is included.

combining the relevant text columns in to a single field, removing punctuation and replacing phrases to ensure their intention is maintained. Modelling involved determining the text weights and the corresponding singular value decomposition (SVD) components followed by the generation of text clusters. A series of classification models were then trained to predict the clusters or the scheduled jobs target variable. This paper presents the results of the classification models and set of findings from the study. The data mining tool set used in this case study was SAS Enterprise Miner 9.

2 Related Work

Prior to beginning the case study, previous examples of applications using text weights to classify documents were sought. A number of useful examples were found that reported useful accuracy for their application.

Kolyshkina and van Rooyne (2006) described an insurance claim cost prediction application. This application predicted if claims result in a top 10% pay-out using a combination of free-format narrative text and codified database fields. Their training data contained a binary target variable that identified if the claim was in the top 10%. Their mining process was iterative with much experimentation to find the optimal algorithm settings and required input from subject matter experts to derive domain relevant concepts. Overall, Kolyshkina and van Rooyne found that by combining text concepts from the free-format text with the codified database fields, prediction of the target insurance claim outcome was improved by 10% using a decision tree classifier.

Likewise, Drucker *et al* (1999) used text weights to train an email spam filter. Drucker evaluated the performance of a number of different classifiers on the task of distinguishing normal email messages from unsolicited spam. Best performance was an overall equal error rate of 0.0193 using a support vector machine (SVM).

Grivel (2005) applied text mining to categorise customer feedback on new cars obtained through phone survey results and transcribed phone calls. The system assigned documents into predefined and dynamically created categories respectively. Grivel claimed a 90% recall and precision.

The previous approaches used bag of words based approaches that analyse the frequency of words within text. In similar problems, Popowich (2006) and Grivel (2005) found that that structured linguistic analysis yields more accurate results than pure stochastic analysis. The linguistically driven process requires more input however from subject matter experts.

While linguistic analysis may be preferable, like many of the common tools currently available, the tool set used for this study (SAS Enterprise Miner 9) only supports the bag of words approached. This study aims to determine what is easily possible for this kind of problem today, hence will use the algorithms available in the tool set.

3 Problem Definition and Objective

The ability to accurately predict maintenance budgets is of financial importance to the client. To do this they need to know how often maintenance is required and how

regularly systems fail. This information would help identify locations where improved maintenance would reduce the number of unexpected system failures.

Unfortunately the data required to accurately perform this analysis had been recorded in free text maintenance logs for the previous twelve years. Text data mining techniques were required to divide the data into scheduled maintenance jobs versus unscheduled repairs.

To achieve this task the data was first prepared by transforming into a format suitable for text analysis. Text mining methods were then applied to determine the text weights and create a feature vector based on the free text data. From the text weights cluster analysis was performed to identify describe natural groupings within the data. At this point we were specifically looking for fault and maintenance activities and groups that describe types of equipment. Next, a number of classification models were trained using decision trees and neural networks algorithms to learn the cluster groups or a target variable.

Finally, we assess how well the classification models allow the records to be classified as scheduled and unscheduled maintenance.

4 Input Data Description

A sample of data from three pump stations consisting of a total of 842 maintenance records was provided as a Microsoft Excel spreadsheet. Records from each pump station were stored in separate worksheets within the one spreadsheet. While the data in each worksheet contained effectively the same columns, small differences in column names and number of columns meant all three worksheets needed to be treated separately before extracting into a common format.

An initial assessment of the data was completed to identify data quality problems that impacted the analysis of the data. Problems identified included long text fields split over multiple columns, missing white space between words, spelling errors, inconsistent capitalisation and punctuation, inconsistent use of acronyms and terms, repeated words, and missing values in many columns.

The input data also lacked a target column to reliably differentiate scheduled maintenance jobs from unscheduled faults. The target column was required to train the classification models and to assess performance assessment of text clustering results. Ideally subject matter experts with knowledge of pump station maintenance and repair would be asked to categorise a sample of the data into either scheduled or unscheduled jobs. In this case, the authors have made a “best guess” at the meaning of the rows and created a simple text file with the unique identifier of each job alongside a target variable where 1 means scheduled maintenance and 0 identifies unscheduled maintenance. The distribution of this variable was split 65/35% between scheduled and unscheduled tasks respectively. The target column was coded to contain no missing values, which allows classification algorithms to use all data rows. But the data set contained noise in a form where the text is not clear. This required significant pre-processing of the data to get it ready for mining.

5 Data Preparation

The text mining algorithms require a data set with a single text column containing all the data to be analysed. The data preparation task involved all steps required to reformat the provided input data into this format.

5.1 Select Text Data

All text columns that described the type of maintenance or repair job were included in the analysis. Columns that identified the specific location, person performing the repair or the time the repair occurred were excluded to avoid the clustering and classification algorithms learning a specific repair job that could not be generalised to other pump stations.

5.2 Clean Data

Misspellings were common in the fault logs. A common error in the input data was words concatenated together where spaces should have been used to separate the words, e.g. “fixedthe” was typed where the words should be “fixed the”.

Since the data set was relatively small Excel's spell checking feature was employed. Excel performs a reasonable job of detecting errors and proposing fixes. While the task required user guidance, fixing the spelling errors was completed quickly and accurately without requiring other external tools.

5.3 Construct Data

For text analysis, a single data file with all the relevant text in a single field is required. Constructing the data involved deriving a combined text column from the relevant text fields in the source data.

LONG_TEXT columns in the pump station data files contained the free text description of the job written by the maintenance worker. Each column was limited to 256 characters. Where the description of the job was longer than 256 characters the data was spread across multiple columns in Excel, only the first of which was labelled. There were effectively eight long text columns in each worksheet that need to be concatenated to recover the actual long text value.

Finally, all other text columns included in the analysis were concatenated with the combined long text column to form the into a single text column data set.

5.4 Integrate Data

Data from the pump stations was given in three separate Excel worksheets. These worksheets were combined into a single flat file table. Columns for each worksheet were identical requiring no special mappings to integrate the data. The hand coded target variable was then joined the pump station data using the unique identifier. These steps created a single table encompassing all data from all three pump stations.

5.5 Formatting Data

Formatting refers to syntactic transforms that do not change the meaning of the data, but assist the modelling algorithms. The first formatting step was to set all text to lower case. While not strictly necessary for text analysis,

setting all fields to lowercase assists other formatting functions.

Punctuation in the text fields, particularly in the long text description, was often omitted or used extraneously. No consistent information was apparent in the use of punctuation. Therefore, to simplify the text mining all punctuation was removed and replaced with spaces. Specifically, the following characters were replaced:

~`!@#\$%^&*()_+~={}|\\;':<>?/,.

Cause Text	Transformed to
no code	nocause
no cause code	nocause
no code - preventative maintenance	nocause - preventative maintenance
no code - electrical trip	nocause - electrical trip
information unavailable from list	Information unavailable from list

Table 1: Cause Text transforms

Three short free text columns contained values that implied that a record was a maintenance or repair job. On their own they contained too many null values and inconsistent use of other values to accurately classify the records. These fields included a Damage Text field that indicated whether there was or was not damage to equipment, a Cause Text field that indicated the possible cause of a fault, and a Breakdown field that indicated if the job was a part of a larger job.

Values in these fields were typically single “yes” or “no” values. Inconsistent use meant the values “no code”, “no cause code” and “no cause code - preventative maintenance” all used in the Cause Text column to signify the same value. The Damage Text column used similar values replacing the word “cause” for “damage”. If left as-is and the text analysis performed on words rather than phrases, then the modifier “no” would be lost and the meaning of “cause” or “damage” may be misinterpreted. To avoid the loss of meaning, values that included the term “no” before the word “code” had the text between these two terms replaced with the single words “nocause”, “nodamage”, and “nobreakdown” for the Cause Text, Damage Text and Breakdown fields respectively. Examples of the cause text transform are shown in table 1.

Finally as part of the formatting functions, common phrases that comprise of more than a word were combined into a single word to assist the text analysis. Common phrases included “Pump station”, “Air condition” and “Programmable logic controller”. Frequently these terms were used inconsistently in the data due to the use of various forms of spelling, abbreviations and acronyms. Table 2 presents examples of the phrases replaced with concatenated words. Note that the list in table 2 was compiled knowing the words would be stemmed during later text analysis. Suffixes to the phrases such as “er” and “ing” were ignored at this point.

6 Text Mining and Clustering

The prepared input data lacked numeric or categorical fields suitable for traditional clustering and classification models. Instead, the text descriptions of pump station

maintenance jobs were transformed into vectors of term weights, which could then be used for clustering and classification. This task was performed using SAS Enterprise Miner's Text Miner component. SAS Text Miner creates three outputs:

1. Converts the free text into term weights
2. Performs singular value decomposition (SVD) on term weights
3. Creates text clusters

Input data was split into training and validation sets using simple random partitions since the distribution of the target variable was not considered skewed enough to require a stratified sample. Data volumes in each group were selected by balancing the training sets need to have sufficient data to create a reliable model against the validation sets need to have enough samples to provide a useful estimation of the model's performance. The final split was a training set containing 75% of the data or 632 records and a validation set with the remaining 25% or 210 records.

Phrase	Replace With	Reason
"pump station"	"pumpstation "	Phrase pump station as a single word to avoid confusion with the singular terms "pump" and "station".
" pstn "	" pumpstation "	Abbreviation for pump station.
"pump stat "	"pumpstation "	Abbreviation for pump station.
"air condition"	"aircondition "	Phrase air conditioner as a single phrase to avoid confusion with the singular terms "air" and "condition".
"air con "	"aircondition "	Abbreviation for air conditioner
"aircon "	"aircondition "	Abbreviation for air conditioner
"program logic controller"	"plc "	Expanded version of the acronym PLC, which is also widely used in the data
"program logic contr"	"plc "	Synonym for PLC
"programmable logic controller"	"plc "	Synonym for PLC

Table 2: Example replacements for common phrases

6.1 Text Mining

The SAS Text Miner component requires input in the form of a single text column from a tabular data. From this it performs all text processing functions from extraction of terms to the creation of clusters. The functions are broadly grouped into parse, transform and cluster steps.

Parse tokenizes the text into individual terms. During the parsing process the number of extracted terms is reduced by stemming, filtering by a stop word list, and combining terms using a synonym list.

During the transform step Text Miner creates numeric vectors from the text terms, thus converting the text into a format suitable for clustering or predictive mining. SAS Enterprise Miner calculates two useful feature vectors from text documents: the term frequency by inverse document frequency (tf×IDF) term weights plus the singular value decomposition (SVD) of the term weights. Tf×IDF weights (Grossman & Frieder 2004, p. 13) calculate a vector for each record with each element representing a single term identified in the parse step. Terms that are frequent within a document but rare across all documents are weighted highest. SVD dimensions (Grossman & Frieder 2004, p. 129) reduce the dimensionality of the tf×IDF vectors by projecting them onto a smaller set of dimensions.

To limit the complexity of the models, only the 100 highest weighted terms were kept. Terms in the top 100 that were irrelevant to the classification goal were excluded from the analysis by adding the terms to the stop word list. This was repeated until all terms in the top 100 were found relevant.

The goal of text clustering was to produce an interpretable number of clusters with a high level of intra-class similarity and a low level of inter class similarity. For the pump station data, clusters that clearly split the records into scheduled maintenance or fault repairs were preferred. Expectation maximisation clustering was used as this method returns an easy to interpret "bag of words" to describe the clusters. Clusters were based on SVD dimensions, the default, as the process of dimension reduction simplifies the work of the clustering algorithm.

The number of terms to describe the clusters was set to 30. At this number of terms the cluster descriptions contain the key terms with little overlap between the clusters. Initially, Text Miner was set to automatically determine the number of clusters. After many clustering attempts, the best separation of the target variable was found when the number of clusters was set to 14. This approach was based on one of Francis' (2006, p70) methods for determining the number of clusters.

The final output of the text mining was a database table containing all fields from the input data, the cluster label, the top 100 term weights, and the SVD dimension components. Descriptions of the final clusters are presented in section 8. While some clusters clearly indicated either the type of task or the equipment involved, the complete set could not conclusively determine if a record was a scheduled or unscheduled job.

7 Classification of Text Mining Results

The clusters provided potentially useful information to the client. Further to this information we wanted to predict the cluster of a new record based on the text mining term weights or SVD components.

First a decision trees was trained using the term weights. While the accuracy of the tree was not expected to be high as it uses one word at a time in classification, it was expected the decision tree would provide some insight into which words best described each cluster. The "cluster" decision tree attempts to predict which of the 14 natural clusters a new unseen record would fit into.

A second classifier was trained to predict the cluster groups, this time aiming for accuracy over interpretability. For this purpose a neural network was trained using the 17 SVD dimensions as inputs and the cluster groups as outputs.

As the clusters could not accurately determine the scheduled maintenance target variable, two more classifiers were trained to predict the target (scheduled maintenance or unscheduled maintenance). As per the cluster classification models, a decision tree was trained using the term weights as input while a neural network was trained using the SVD components. Using the term weights for the decision tree identifies terms that best predict the target variable. Using the SVD weights simplifies the training of the neural network by decreasing the number inputs, thereby increasing the

No.	Cluster Description	No. of Records (% of total)
1	Various Maintenance	119 (18.8%)
2	Repair Jobs	64 (10.1%)
3	Discharge and Actuators	49 (7.8%)
4	Gantry Cranes	23 (3.6%)
5	Cooling Water Pumps	37 (5.9%)
6	Pump Impeller Maintenance	15 (2.4%)
7	Upgrade Projects	32 (5.1%)
8	Maintenance Tests	71 (11.2%)
9	PLC and Pumps	55 (8.7%)
10	Vibration Tests	43 (6.8%)
11	Flowmeter Servicing	50 (7.9%)
12	Battery Replacement	18 (2.8%)
13	Information Unavailable	34 (5.4%)
14	Fire Alarm and Ventilation Repairs	22 (3.5%)
Total		632 (100.0%)

accuracy of the model.

Table 3: Cluster descriptions

8 Results

8.1 Cluster Descriptions

A set of 14 clusters were identified and described based on the common terms within the cluster, the equipment involved and the type of job. The cluster descriptions are shown in table 3. Of the 14 clusters, only 6 can be considered homogenous. These are gantry cranes (4), pump impellers maintenance (6), upgrade projects (7), maintenance tests (8), vibration tests (10), and battery replacement (12). The remaining clusters were heterogeneous, containing jobs that covered multiple infrastructure components and a mixture of scheduled maintenance and repair work.

Figure 1 compares the text clusters to the scheduled maintenance target variable. Clusters 2 (29.7%), 9 (25.5%) and 11 (34) contain relatively low percentages of scheduled maintenance records indicating clusters containing predominantly faults. Looking at the percentages, all other clusters consist of a majority of scheduled maintenance records. Clusters 1 (73.9%), 3

(63.3%), 12 (66.7%) and 14 (72.7%) are too close to random chance to clearly class as scheduled maintenance. Since the total percentage of scheduled maintenance is 65.7%, then taking a random sample from the training set would lead to similar results. This leaves clusters 4 (78.3%), 5 (81.1%), 6 (100.0%), 7 (84.4%), 8 (87.3%), 10 (88.4%), and 13 (82.4%) that can be clearly considered scheduled maintenance. It can be concluded that 40.3% of records form the 7 clusters that can be classified as scheduled maintenance; 32.9% of records form the 3 clusters that can be classified as scheduled maintenance; and 26.8% of records form the 4 clusters that can not be deterministically classified.

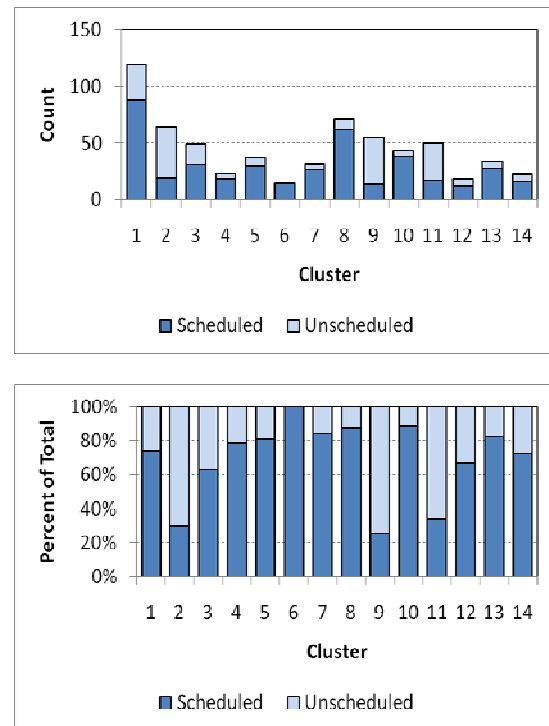


Figure 1: Clusters by scheduled maintenance

Model	Inputs	Output	Misclassification Rate	
			Training Set	Validation Set
Decision Tree	Term Weights	Cluster Labels	0.367	0.443
Neural Network	SVD components	Cluster Labels	0.120	0.167
Decision Tree	Term Weights	Target	0.148	0.150
Neural Network	SVD components	Target	0.147	0.172

Table 4: Misclassification rates on training and validation sets

8.2 Classification Results

Misclassification rates for the four classification models are shown in table 4. For the decision tree trained to recognise the cluster labels the misclassification rate on

the test set was 44.3%. This result is worse than random chance since selecting all records as scheduled would result in a misclassification rate of 34.3%, the total percentage of unscheduled repairs.

The problem search space was too large with 100 input terms used to learn 14 cluster labels and the search too restricted with small sample sizes and splitting on one term at a time for the decision tree to learn the mapping accurately.

It was hoped that the cluster decision tree would provide some insight into the words that best describe the cluster. As expected from the high misclassification rate suggests, the rules learnt are ambiguous. For example, table 5 shows the rules that split the cluster 9, moderately heterogeneous class regarding PLCs, and cluster 6, a completely homogenous cluster regarding pump impeller maintenance. It is not clear from a business context why a greater use of the word “pump” would lead to one class containing all scheduled jobs over a second class containing mostly unscheduled work.

Cluster 6	Cluster 9
IF 7.39 >= + pump AND + cool < 2.8 AND flowmeter < 1.81 AND + vibration < 2 AND + valve < 2 AND + center < 1.68	IF 3.10 >= + pump < 7.39 AND + cool < 2.8 AND flowmeter < 1.81 AND + vibration < 2 AND + valve < 2 AND + center < 1.68

Table 5: Examples of cluster decision tree rules

Scheduled Maintenance	Unscheduled Repairs
Leaf 16: n=124, 87% correct IF + reset < 2.61 AND high < 2.53 AND + damage < 1.60 AND + repair < 1.70	Leaf 17: n=7, 57% correct IF 2.601 >= + reset AND high < 2.53 AND + damage < 1.60 AND + repair < 1.70
Leaf 13: n=5, 60% correct IF 2.72 >= + initiate AND + sign < 2.11 AND 1.70 >= + repair	Leaf 9: n=4, 100% correct IF 2.53 >= high AND + damage < 1.60 AND + repair < 1.70
Leaf 14: n=2, 100% correct IF + damage < 1.60 AND 2.11 >= + sign AND 1.70 >= + repair	Leaf 12: n=33, 91% correct IF + initiate < 2.72 AND + sign < 2.11 AND 1.70 >= + repair
	Leaf 15: n=32, 78% correct IF 1.60 >= + damage AND 1.70 < + repair

Table 6: Examples of target decision tree rules

The other three classifiers all provided misclassification rates in the 15% to 17% range. Both neural network classifiers provided comparable misclassification rates when trained to learn the cluster labels, 16.7%, or the target variable, 17.2%, and using SVD values as inputs.

Interestingly, the best performance on this data set was obtained from the decision tree trained to learn the target variable using the term weights as input. For this classifier the misclassification rate was 15.0%. The difference between the decision tree and the neural network is more likely due to choice of input than the training algorithm. While the SVD components lead to a

smaller problem for the neural networks to learn, dimension reduction smooths the data possibly losing some descriptive detail.

Rules from the target variable decision tree provide a degree of comprehensibility (table 6). Leaf 16 predicts the most scheduled maintenance jobs and is associated with the low weights for the words “repair” and “damage”. Similarly, leaf 12 predicts the high volume of unscheduled repairs and is associated with high weights for the term “repair”.

9 Discussion and Conclusion

Misclassification rates around 15% to 17% are too high for many applications. Yet this rate can be acceptable for business decisions such as budgeting when compared to alternative decision making methods.

In context of the business objective, the text clusters found 14 distinct types of jobs. While these cannot clearly be categorised as scheduled or unscheduled from the cluster descriptions, some clusters were found to contain relatively homogeneous types of jobs. That is, they contained mostly one type of job or affected a specific type of equipment. Classification of these types of jobs can be predicted using the neural network cluster classification model. This would be useful if, for example, all battery replacements need to be identified.

Using a target variable for training scheduled maintenance versus unscheduled repairs could be predicted using a classification model. This case study tested a decision tree trained using term weights and a neural network trained on SVD components and obtained misclassification rates of 15.0% and 17.2% respectively. Therefore, with some input from subject matter experts, text features can be used to classify documents on small data sets with a moderate level of accuracy.

The case study provided a number of insights on how to use text data mining techniques on low quality data sets often found in spreadsheet applications.

- Formatting the data for text mining required appending many columns into a single text column. Semi-structured data values could be transformed by including a token, e.g. the column name, to identify the column. This ensured the context of the value was not lost when combined with other free text columns.
- Determining text mining stop words and common phrases replacements is an iterative process. Clusters needed to be recomputed until all of the top terms were relevant to the data mining goal and the clusters matched expectations.
- Clustering provided interesting classes that covered subsets of the desired task. Clusters did not provide mutually exclusive splits on desirable characteristics such as scheduled versus unscheduled jobs or by different types of equipment. The subsets provide useful information for the business, but use of clusters in this fashion is more opportunistic than by design.
- Classification models were able to learn both the cluster labels and a binary target variable using information derived from free text fields. This was

achieved with moderate levels of accuracy relative to the application. To train the classification models required manual classification of a subset of the data to create the target variable. This activity would best be performed by subject matter experts.

- When trained to learn the cluster labels a decision tree using text terms as inputs failed to learn the mapping. The problem search space was too large for the decision tree to learn one term at a time. A neural network using the SVD components as inputs learnt the mapping successfully.
- When learning a binary target variable the decision tree trained using text terms as inputs marginally outperformed a neural network trained using SVD components. The reduction to a single binary output variable simplified the problem search space allowing the decision tree to learn the mapping successfully. The process of dimension reduction, while simplifying the classification task, smooths detail in the inputs reducing the accuracy of the neural network.

This case study demonstrates that applying text data mining techniques in low quality data situations is viable provided the value of the data justifies the effort required applying text data mining.

10 Acknowledgement

We will like to sincerely thank CRC for Integrated Engineering Asset Management (CIEAM) to provide us the dataset to conduct this case study. We will also like to thank Colin Fidge and Lin Ma to support this research.

11 References

- Chapman, P., Clinton, J., Khabaza, T., Reinartz, T., & Wirth, R. (1999): The CRISP-DM process model. Technical Report, Crisp Consortium. <http://www.crisp-dm.org/>. Accessed on 11 Jul 2008.
- Drucker, H., Wu, D. & Vapnik, V. (1999): Support Vector Machines for Spam Categorization. *IEEE Transactions on Neural Networks*, **10**(5):1048-1054.
- Francis, L.A. (2006): Taming Text: An Introduction to Text Mining. *Casualty Actuarial Society Forum*, Winter, 51-88.
- Grivel, L. (2005): Customer feedbacks and opinion surveys analysis in the automotive industry. In Zanasi, A. (ed). (2005): *Text Mining and its applications to intelligence, CRM and Knowledge Management*., WITpress.
- Grossman, D. & Frieder, O. (2004): *Information Retrieval: Algorithms and Heuristics*. 2nd edn., Springer.
- Kolyshkina, I., & van Rooyen, M. (2006) Text Mining for Insurance Claim Cost Prediction. In G.J. Williams, & S.J. Simoff (Eds.), *Data Mining LNAI 3775*, Berlin, 192-202, Springer-Verlag.
- Popowich, F. (2005) Using Text Mining and Natural Language Processing for Health Care Claims Processing. *SIGKDD Explorations*, **7**(1):41-48.

Rayid, G., Probst, K., Liu, Y., Krema, M. and Fano, A. (2006) Text Mining for Product Attribute Extraction. *SIGKDD Explorations*, **8**(1), pp41-48.

Categorical Proportional Difference: A Feature Selection Method for Text Categorization

Mondelle Simeon

Robert Hilderman

Department of Computer Science
University of Regina
Regina, Saskatchewan, Canada S4S 0A2
Email: {simeon2m, hilder}@cs.uregina.ca

Abstract

Supervised text categorization is a machine learning task where a predefined category label is automatically assigned to a previously unlabelled document based upon characteristics of the words contained in the document. Since the number of unique words in a learning task (i.e., the number of features) can be very large, the efficiency and accuracy of the learning task can be increased by using feature selection methods to extract from a document a subset of the features that are considered most relevant. In this paper, we introduce a new feature selection method called categorical proportional difference (CPD), a measure of the degree to which a word contributes to differentiating a particular category from other categories. The CPD for a word in a particular category in a text corpus is a ratio that considers the number of documents of a category in which the word occurs and the number of documents from other categories in which the word also occurs. We conducted a series of experiments to evaluate CPD when used in conjunction with SVM and Naive Bayes text classifiers on the OHSUMED, 20 Newsgroups, and Reuters-21578 text corpora. Recall, precision, and the F-measure were used as the measures of performance. The results obtained using CPD were compared to those obtained using six common feature selection methods found in the literature: χ^2 , information gain, document frequency, mutual information, odds ratio, and simplified χ^2 . Empirical results showed that, in general, according to the F-measure, CPD outperforms the other feature selection methods in four out of six text categorization tasks.

Keywords: Text categorization, feature selection, supervised learning, categorical proportional difference.

1 Introduction

Due to the consistent and rapid growth of unstructured textual data that is available online, text categorization, the machine learning task of automatically assigning a predefined category label to a previously unlabelled document, is essential for handling and organizing this data. Widely used and well studied text categorization methods include Naive Bayes (Kim et al., 2006), support vector machines

(SVM) (Joachims, 1998), and k-nearest neighbor (k-NN) (Han et al., 2001) methods. All of these methods use a collection of pre-labelled examples to build a predictive model for each distinct category contained in the examples. For a survey of these and other automated text categorization methods and applications, see (Sebastiani, 2002).

If the number of unique words (i.e., the number of features) encountered by a text categorization task is large, the efficiency and accuracy of the method may be adversely affected. For example, accuracy can be reduced if a method is used where features with low predictive value are included in the model, and efficiency can be improved if a method is used where less computation and/or memory is required to categorize a given text corpus (Forman, 2008). As a result, feature selection methods are used to address efficiency and accuracy by extracting from a document a subset of the features that are considered most relevant. When using a feature selection method, each word is scored using some predefined measure and the most relevant words are selected based upon this measure.

1.1 Related Work

Many feature selection methods have been proposed and studied by various authors. In (Yang and Pedersen, 1997), document frequency, two information gain methods, mutual information, term strength, and three chi-square methods are evaluated on the Reuters and OHSUMED text corpora using k-nearest neighbor and linear least squares fitting classifiers. Here it is suggested that document frequency, one of the information gain methods, and one of the chi-square methods are the most effective feature selection methods, with strong correlations found between the results obtained using these methods.

The odds ratio was proposed as a feature selection method and compared to a variety of other feature selection methods in (Mladenic and Grobelnik, 1999). Here the evaluation was somewhat limited as only a multinomial Naive Bayes text classifier was used and only on the Reuters text corpus. They did find that their proposed method performed best. However, their results disagreed with the conclusions in (Yang and Pedersen, 1997) in regard to the strength of IG as a feature selection method. They attributed this discrepancy to differences in domain definitions and the classifiers used.

In (Galavotti et al., 2000), a simplified chi-square feature selection method was proposed and compared to the original chi-square method. Again, the evaluation was somewhat limited as only different variants of a k-NN classifier were used and only on the Reuters text corpus. They did find, though, that their proposed method outperformed two chi-square feature selection methods under conditions that would be characterized as extremely aggressive feature se-

lection.

In (Ng et al., 1997), a correlation coefficient based variant of the chi-square feature selection method, was compared to a chi-square feature selection method. In this study, a perception learning classifier and the Reuters text corpus were used. They found that their method outperformed other methods.

A group of scoring measures for feature selection were proposed in (Montanes et al., 2005). The measures were evaluated using an SVM classifier on the Reuters and OHSUMED text corpora. Here it was found that results were mixed with their scoring measures outperforming both information gain and TF*IDF in some situations.

In (Forman, 2003), a study of the effects of various feature selection methods on an SVM text classifier is described. Here, an evaluation methodology is proposed for determining the feature selection method or methods that are most likely to provide the best results. In addition, a new feature selection method, called bi-normal separation, is shown to outperform other commonly known methods in some circumstances.

The collected results from various feature selection studies are described in (Sebastiani, 2002). Here, some general recommendations are made regarding the relative performance of numerous feature selection methods. However, it is suggested that in order to make more conclusive statements on the relative performance of the feature selection methods studied, that comparative experiments under controlled conditions using a variety of text corpora and classifiers are required.

1.2 Our Contribution

In this paper, we introduce a new feature selection method called categorical proportional difference (CPD), a measure of the degree to which a word contributes to differentiating a particular category from other categories in a text corpus. The CPD for a word in a particular category is a ratio that considers the number of documents of the category in which the word occurs and the number of documents from other categories in which the word also occurs. We conducted a series of experiments to evaluate CPD when used in conjunction with SVM and Naive Bayes classifiers on the OHSUMED, 20 Newsgroups, and Reuters-21578 text corpora. The F-measure, a measure that combines both recall and precision, was used as the measure of performance. The results obtained using CPD were compared to those obtained using six feature selection methods commonly found in the literature: χ^2 (Yang and Pedersen, 1997), information gain (Yang and Pedersen, 1997), document frequency (Yang and Pedersen, 1997), mutual information (Yang and Pedersen, 1997), odds ratio (Mladenic and Grobelnik, 1999), and simplified- χ^2 (Galavotti et al., 2000). Empirical results showed that, in general, according to the F-measure, CPD is an effective feature selection method, outperforming the other feature selection methods in four out of six text categorization tasks.

The remainder of this paper is organized as follows. In Section 2, we introduce the CPD feature selection measure. In Section 3, we provide an overview of relevant details regarding our methodological approach to evaluating CPD. In Section 4, we present the experimental results from a comprehensive series of text categorization tasks. We conclude in Section 5 with a summary of our results and suggestions for future work.

2 A New Method for Feature Selection

In this section, we introduce the CPD feature selection measure and provide a brief example to demonstrate its use on a sample text corpus.

2.1 Categorical Proportional Difference

We define CPD (and the other feature selection methods described in Section 3.2) in reference to a typical 2×2 contingency table. A example contingency table is shown in Table 1, where A is the number of times word w and category c occur together, B is the number of times word w occurs without category c , C is the number of times category c occurs without word w , D is the number of times neither word w nor category c occur, and $N = A + B + C + D$.

Table 1: An example contingency table

	c	$\neg c$	$\Sigma \text{ Row}$
w	A	B	$A + B$
$\neg w$	C	D	$C + D$
$\Sigma \text{ Column}$	$A + C$	$B + D$	N

CPD measures the degree to which a word contributes to differentiating a particular category from other categories in a text corpus. The possible values for CPD are restricted to the interval $(-1, 1]$, where values near -1 indicate that a word occurs in approximately an equal number of documents in all categories (it approaches -1 as the number of categories increases) and a 1 indicates that a word occurs in the documents of only one category. More formally, the categorical proportional difference for word w in category c is defined as

$$\text{CPD}(w, c) = \frac{A - B}{A + B}.$$

That is, the calculation is simply the ratio of the difference between the number of documents of a category in which a word occurs and the number of documents of other categories in which the word also occurs, divided by the total number of documents in which the word occurs. The CPD for a word is the ratio associated with the category c_i for which the value is greatest. That is,

$$\text{CPD}(w) = \max_i \{\text{CPD}(w, c_i)\}.$$

2.2 Example

Words and their frequency of occurrence in labelled documents from a sample text corpus are shown in Table 2. In Table 2, the *Word* column contains a subset of the words occurring in the documents, the *No. in Grain*, *No. in Trade*, *No. in Interest*, and *No. in Agriculture* columns contain the number of documents in which a word occurs that have been assigned the corresponding category label, the *No. of Documents* column contains the total number of documents in which a word occurs, and the *CPD* column contains the value calculated for a word using the CPD feature selection measure.

For example, consider the word “wheat”, where all occurrences of the word are in documents of the *Grain* category. Here, $\text{CPD}(\text{wheat}, \text{Grain}) = (25 - 0) / (25 + 0) = 1$, and $\text{CPD}(\text{wheat}, \{\text{Trade}, \text{Interest}, \text{Agriculture}\}) = (0 - 25) / (0 + 25) = -1$. Thus, $\text{CPD}(\text{wheat}) = \max \{1, -1, -1, -1\} = 1$. Now consider the word “economy”, where the word occurs in the same number of documents in each category. Here we have

Table 2: Word distribution and CPD in a sample text corpus

Word	No. in Grain	No. in Trade	No. in Interest	No. in Agriculture	No. of Documents	CPD
wheat	25	0	0	0	25	1.00
economy	15	15	15	15	60	-0.50
surplus	18	5	0	2	25	0.44
quotas	1	50	1	1	53	0.89
feed	7	9	4	11	31	-0.29

CPD(economy, {Grain, Trade, Interest, Agriculture}) = $(15 - 45) / (15 + 45) = -0.5$ and CPD(economy) = -0.5. Finally, consider the word “surplus”. Here CPD(surplus, Grain) = $11 / 25 = 0.44$, CPD(surplus, Trade) = $-15 / 25 = -0.6$, CPD(surplus, Interest) = $-25 / 25 = -1.0$, CPD(surplus, Agriculture) = $-21 / 25 = -0.84$ and CPD(surplus) = 0.44.

3 Methodological Overview

In this section, we describe relevant details and issues related to the underlying methodological approach used to obtain the experimental results.

3.1 Text Classifiers

Text categorization in this, and other work, is essentially a two-step process. In the first step, a category model for each category in a text corpus of labelled training documents is built. In the second step, a text categorization algorithm compares an unlabelled document to the learned category models to determine the “best” category label to assign to the unlabelled document. In this work, we used the SVM and Naive Bayes classifiers provided in the Weka collection of machine learning algorithms (Witten and Frank, 2005) to generate all the experimental results.

3.2 Feature Selection Methods

Seven feature selection methods were used in conjunction with the SVM and Naive Bayes classifiers: CPD, χ^2 , information gain (IG), document frequency (DF), mutual information (MI), odds ratio (OR), and simplified- χ^2 (S- χ^2). CPD was previously defined in Section 2.1, and the variables A , B , C , D , and N used below are the same as those described in that section.

χ^2 measures the lack of independence between a word w and a category c if it is assumed that the occurrence of a word is actually independent from the category label. For a word w and a category c ,

$$\chi^2(w, c) = \frac{N(AD - BC)^2}{(A + C)(B + D)(A + B)(C + D)}.$$

After χ^2 for every combination of the word w and category c is determined, we take the maximum χ^2 value over all the categories c_i as the χ^2 value for the word. That is,

$$\chi^2(w) = \max_i \{\chi^2(w, c_i)\}.$$

IG measures the decrease in entropy when a selected feature is present versus when it is absent. For a word w and a category c ,

$$\begin{aligned} \text{IG}(w, c) = & e(POS, NEG) - [p(w)e(TP, FP) + \\ & p(\neg w)e(FN, TN)], \end{aligned}$$

where

$$e(x, y) = -\frac{x}{x+y} \log_2 \frac{x}{x+y} - \frac{y}{x+y} \log_2 \frac{y}{x+y},$$

$$POS = A + C,$$

$$NEG = B + D,$$

$$p(w) = \frac{A + B}{N},$$

$$p(\neg w) = 1 - p(w),$$

and TP , FP , TN , and FN are the number of true positives, false positives, true negatives, and false negatives, respectively. We take the sum of the IG values for a word over all the categories c_i as the IG value for the word. That is,

$$\text{IG}(w) = \sum_i \text{IG}(w, c_i).$$

DF simply measures the number of documents in which a word occurs and is determined without reference to category labels. So,

$$\text{DF} = A + B.$$

MI measures the mutual dependence of a word w and a category c . For a word w and a category c ,

$$\text{MI}(w, c) = \log \frac{AN}{(A + B)(A + C)}.$$

We take the maximum of the MI values for a word over all the categories c_i as the MI value for the word. That is,

$$\text{MI}(w) = \max_i \{\text{MI}(w, c_i)\}.$$

OR measures the odds of a word occurring in the positive class normalized by that of the negative class. To avoid a situation where division by zero may occur, one is added to any zero in the denominator. For a word w and a category c ,

$$\text{OR}(w, c) = \frac{AD}{BC}.$$

We take the sum of the OR values over all categories c_i as the OR value for the word. That is,

$$\text{OR}(w) = \sum_i \text{OR}(w, c_i).$$

S- χ^2 is a variant of χ^2 . Positive values are indicative of membership of a word w in a category c , and negative values are indicative of non-membership. For a word w and a category c ,

$$\text{S-}\chi^2(w, c) = \frac{AD - BC}{N^2}.$$

We take the maximum S- χ^2 value over all categories c_i as the S- χ^2 value for the word. That is,

$$\text{S-}\chi^2(w) = \max_i \{\text{S-}\chi^2(w, c_i)\}.$$

Table 3: Summary statistics for the randomly generated subset datasets

Description	#/%	Statistic	OHSUMED	20 Newsgroups	Reuters-21578
Possible Categories	#		15	16	14
Categories/Dataset	#	max	3	3	5
	#	min	2	2	4
	#	mean	2.4	2.3	4.6
Documents/Dataset	#	max	1200	1200	1120
	#	min	800	800	921
	#	mean	960	920	1069
Words/Document	#	max	60	102	65
	#	min	53	52	58
	#	mean	57	76	61
Features/Dataset	#	max	8327	16923	6737
	#	min	6403	8005	4919
	#	mean	7239	11601	5959
Words in Single Category	%	max	66	77	54
	%	min	53	69	45
	%	mean	59	73	50
Categories/Word	#	max	3	3	5
	#	min	1	1	1
	#	mean	1.5	1.3	2.1

3.3 Performance Measures

To evaluate the utility of the various feature selection methods used, we use the F-measure, a measure that combines precision and recall, two commonly used measures of text categorization performance. Precision (P) measures the percentage of documents assigned to category c that are correctly assigned to category c , and recall (R) measures the percentage of documents that should have been assigned to category c that actually were assigned to category c . More formally,

$$P_i = \frac{TP_i}{TP_i + FP_i}$$

and

$$R_i = \frac{TP_i}{TP_i + FN_i},$$

where TP_i (i.e., true positives) is the number of documents assigned correctly to category c_i , FP_i (i.e., false positives) is the number of documents assigned to category c_i that should have been assigned to other categories, and FN_i (i.e., false negatives) is the number of documents assigned to other categories that should have been assigned to category c_i . The F-measure (F) is the harmonic average of precision and recall, and is defined as

$$F_i = \frac{2P_iR_i}{P_i + R_i},$$

where P_i and R_i are the precision and recall, respectively, for category c_i . After the F-measure is determined for each category c_i , the macro-average (i.e., the traditional arithmetic mean) of these values is determined, and this value becomes the overall F-measure. That is,

$$\text{average maximum } F = \frac{\sum_i F_i}{n},$$

where n is the number of categories.

3.4 Text Corpora

In previous studies on text categorization and feature selection, it is common for a new method or measure to be evaluated against frequently used text corpora such as OHSUMED (OHSUMED, 2005), 20 Newsgroups (Newsgroups, 1999), and Reuters-21578 (Reuters-21578, 1997). The OHSUMED text

corpus is a subset of the MEDLINE database containing 348,588 abstracts from 270 medical journals for the five years from 1987 to 1991. The 20 Newsgroups text corpus is a set of 20,000 Usenet articles. The Reuters-21578 text corpus is a set of economic news stories published in 1987.

In this work, we also use the OHSUMED, 20 Newsgroups, and Reuters-21578 text corpora, but we use them only as repositories from which to randomly generate many subset datasets having different characteristics. For example, from OHSUMED, we use only the first 20,000 abstracts from 1991, where each document is labelled with one of 23 cardiovascular disease categories. From these documents, ten unique subset datasets were randomly generated, each containing from 800 to 1200 documents. From 20 Newsgroups, where each article is labelled with one of 20 categories, ten unique subset datasets were randomly generated in the same way as for OHSUMED. And from Reuters-21578, where each document is labelled with one of 90 categories, we use the ModAptè split which contains 12,902 documents (9,603 in the training set and 3,299 in the test set). The documents in the training and test sets were combined into one document collection and those documents from categories that contained fewer than 100 documents were discarded. From the remaining documents, ten unique subset datasets were randomly generated, each containing from 921 to 1120 documents.

Summary statistics for the randomly generated subset datasets from the three text corpora are shown in Table 3. In Table 3, the #/% column describes either a count (i.e., #) or a percentage (i.e., %), the *Statistic* column describes the maximum, minimum, or arithmetic mean of the corresponding count or percentage, and the *OHSUMED*, *20 Newsgroups*, and *Reuters-21578* columns describe the values of the corresponding measures. For example, for the ten randomly generated Reuters-21578 datasets, the maximum, minimum, and arithmetic mean for the number of features per dataset (i.e., Features/Dataset), is 6737, 4919, and 5959, respectively.

3.5 Our Approach

Given some collection of text corpora and some collection of feature selection methods, the algorithm shown in Figure 1 describes the steps followed in our approach to generating the experimental results. In Figure 1, the algorithm consists of two phases.

In the first phase, the document pre-processing phase (lines 3 to 20), a word is compared against a list of common stop words, and if it is determined to be a stop word, it is discarded (lines 3 to 5). Punctuation

```

1: for each each text corpus do
2:   for each dataset randomly generated from the text corpus do
3:     for each document in the dataset do
4:       Remove stop-words, punctuation, and non-alphanumeric text
5:     end for
6:     for each remaining word in the dataset do
7:       Stem the word (a Porter stemmer was used)
8:       Store each unique stemmed word in the word list
9:     end for
10:    for each word in the word list do
11:      Determine TF (i.e., term frequency)
12:      Store TF in the corresponding element of the weight matrix
13:      Determine IDF (i.e., inverse document frequency)
14:      if IDF == zero then
15:        Remove the word from the word list
16:        Remove the corresponding TF from the weight matrix
17:      else
18:        Store normalized TF*IDF in the corresponding element of the weight matrix
19:      end if
20:    end for
21:    for each classifier do
22:      Create a Weka model using the word list and weight matrix
23:      Perform ten-fold cross-validation
24:      Store the F-Measure
25:      for each feature selection method do
26:        for each word in the word list do
27:          Score each word according to the feature selection method
28:          Store each unique score in the score list
29:        end for
30:        Sort the score list in ascending order
31:        Set the maximum F-measure to zero
32:        for each score in the score list from smallest to largest do
33:          Use the current score as the cutoff score
34:          for each word in the word list whose score <= the cutoff score do
35:            Remove the word from the word list
36:            Remove the corresponding TF*IDF from the weight matrix
37:          end for
38:          Create a Weka model using the word list and weight matrix
39:          Perform ten-fold cross-validation
40:          if the F-measure > the maximum F-measure then
41:            Set the maximum F-measure to F-measure
42:          end if
43:        end for
44:      end for
45:    end for
46:  end for
47:  Determine the average maximum F-measure over all the datasets
48: end for

```

Figure 1: The steps followed in our approach to generating the experimental results

and non-alphanumeric text is also discarded. The remaining words are stemmed and the unique words are stored in a word list (lines 6 to 9). Then an $m \times n$ weight matrix is built for the current dataset (lines 10 to 20), where m is the number of documents in the dataset and n is the number of words in the feature space. Each word is associated with a particular column in the matrix and the element at the intersection of a row and column is the normalized TF*IDF value for the word in the corresponding document. In addition to keeping track of the unique words encountered, the word list also specifies the column in the weight matrix that is associated with each word.

In the second phase, the exhaustive search phase (lines 21 to 45), Weka classification models are constructed for the current dataset to determine F-measure values using a five step process. In the first step, the base F-measure is determined by running the current classifier without any feature selection method (lines 22 to 24). That is, using the full feature space. In the second step, the current classifier is run on the the current dataset using each feature selection method (lines 25 to 44) for each feature selection method. To find the maximum possible F-measure, each word is scored using the current feature selection method and the unique scores are stored in a sorted score list (lines 26 to 30). In the third step, each score is used as a cutoff point to eliminate features from the feature space (lines 33 to 37). In the fourth step, the maximum possible F-measure is determined by running the current classifier on the reduced feature space (lines 38 to 42). Finally, in the

fifth step, the average of the maximum F-measures for the datasets is determined.

It is important to note that the search is exhaustive and to understand why an exhaustive search is required. For example, if there are 1,000 scores in the score list, the classifier must be run 1,000 times. It is not merely sufficient to run the classifier using the scores associated with the endpoints of some pre-determined intervals (e.g., every 100-th score) because that would result in determining the F-measure for only 10 scores, and the maximum F-measure could occur at some score not associated with an endpoint. Further, for each score, different words are eliminated from the feature space and the F-measures generated by each run of the classifier do not follow some regular curve, where the intermediate points can be interpolated. That is, the F-measure may increase or decrease as words are eliminated from the feature space.

4 Experimental Results

In this section, we present the results of our experimental evaluation of the CPD feature selection method. All of the experiments were run under Windows XP on an Intel Core 2 1.83 GHz processor with 3072 MB of memory. Due to the exhaustive search phase required to find the average maximum F-measure, the actual calendar duration of the experiments required several weeks to run to completion. The results are shown in Tables 4 through 6.

The relative performance of the seven feature selection methods is shown in Tables 4 and 5.

Table 4: Relative performance of feature selection using an SVM classifier

<i>Text Corpus</i>	<i>Feature Selection</i>	<i>Rank</i>	<i>Avg. Max. F-Measure</i>	<i>Change</i>	<i>Standard Deviation</i>	<i>Avg. Feature Space %</i>
OHSUMED	None	6	0.778	—	0.115	100.0
	CPD	1	0.900	+0.122	0.083	61.2
	χ^2	5	0.821	+0.043	0.124	24.9
	IG	2	0.867	+0.089	0.086	28.7
	DF	7	0.768	-0.010	0.111	46.9
	MI	2	0.867	+0.089	0.109	59.6
	OR	3	0.857	+0.079	0.082	16.2
	S- χ^2	4	0.823	+0.045	0.100	31.9
20 Newsgroups	None	7	0.956	—	0.037	100.0
	CPD	1	0.979	+0.023	0.018	76.5
	χ^2	6	0.957	+0.001	0.035	60.9
	IG	2	0.974	+0.018	0.021	46.5
	DF	8	0.948	-0.008	0.047	44.7
	MI	3	0.967	+0.011	0.030	78.9
	OR	4	0.964	+0.008	0.031	35.8
	S- χ^2	5	0.963	+0.007	0.029	48.2
Reuters-21578	None	8	0.761	—	0.074	100.0
	CPD	2	0.805	+0.044	0.071	64.7
	χ^2	5	0.778	+0.017	0.074	26.0
	IG	3	0.800	+0.039	0.064	9.5
	DF	6	0.773	+0.012	0.070	15.0
	MI	7	0.763	+0.002	0.073	99.7
	OR	1	0.823	+0.062	0.058	11.0
	S- χ^2	4	0.790	+0.029	0.067	10.1

Table 5: Relative performance of feature selection using a Naive Bayes classifier

<i>Text Corpus</i>	<i>Feature Selection</i>	<i>Rank</i>	<i>Avg. Max. F-Measure</i>	<i>Change</i>	<i>Standard Deviation</i>	<i>Avg. Feature Space %</i>
OHSUMED	None	8	0.754	—	0.109	100.0
	CPD	1	0.856	+0.102	0.088	66.3
	χ^2	5	0.774	+0.020	0.130	11.6
	IG	2	0.847	+0.093	0.085	13.0
	DF	7	0.760	+0.006	0.109	17.9
	MI	6	0.761	+0.007	0.120	92.8
	OR	3	0.846	+0.092	0.088	16.1
	S- χ^2	4	0.817	+0.063	0.095	11.9
20 Newsgroups	None	6	0.918	—	0.056	100.0
	CPD	1	0.947	+0.029	0.039	77.3
	χ^2	7	0.915	-0.003	0.054	15.3
	IG	1	0.947	+0.029	0.037	20.1
	DF	4	0.922	+0.004	0.055	23.0
	MI	5	0.921	+0.003	0.053	90.3
	OR	2	0.941	+0.023	0.039	24.6
	S- χ^2	3	0.938	+0.020	0.043	12.9
Reuters-21578	None	7	0.727	—	0.068	100.0
	CPD	3	0.773	+0.046	0.066	62.0
	χ^2	5	0.752	+0.025	0.066	18.9
	IG	2	0.775	+0.048	0.068	12.8
	DF	6	0.746	+0.019	0.070	11.5
	MI	8	0.725	-0.002	0.068	100.0
	OR	1	0.783	+0.056	0.066	30.8
	S- χ^2	4	0.763	+0.036	0.058	14.9

In Tables 4 and 5, the *Feature Selection* column describes the feature selection method used on the corresponding text corpus (the term “None” in this column describes the base case where no feature selection was used), the *Rank* column describes the standing of the F-measure for the corresponding feature selection method in relation to the F-measures for the other methods, the *Avg. Max. F-Measure* column describes the arithmetic mean of the ten largest F-measure values obtained from the ten randomly generated datasets, the *Change* column describes the difference between the average maximum F-measure value obtained when no feature selection is used and the average maximum F-measure value obtained for the corresponding feature selection method, the *Standard Deviation* column describes the standard deviation of the ten largest F-measure values used to determine the average maximum F-measure, and the *Avg. Feature Space %* column describes the average percentage of the feature spaces used corresponding to the ten largest F-measure values. For example, in Table 4, the average maximum F-measure corresponding to CPD when using OHSUMED is 0.900, representing an increase of +0.122 over the F-measure

obtained when no feature selection is used, and the average percentage of the feature space used is 61.2%. The highest ranking feature selection method according to the average maximum F-measure is indicated by bold font. Average maximum F-measure values shown in bold represent a statistically significant difference from the F-measure obtained when no feature selection is used. The paired Student’s t-test was used to determine statistical significance using a 95% level of significance (i.e., $\alpha = 0.05$) and a null hypothesis that the mean difference between paired observations is zero.

In Table 4, when using OHSUMED, all of the feature selection methods showed a statistically significant increase in the F-measure from when no feature selection is used, except for DF, which showed a statistically significant decrease. When using 20 Newsgroups, only CPD, IG, and S- χ^2 showed a statistically significant increase. DF showed a decrease, but it was not statistically significant. And when using Reuters-21578, all of the feature selection methods showed a statistically significant increase, except for MI. When using OHSUMED and 20 Newsgroups, CPD showed the largest increase, while on Reuters-

21578, OR showed the largest increase followed by CPD, which ranked second.

In Table 5, when using OHSUMED and 20 Newsgroups, only CPD, IG, DF, OR, and $S\text{-}\chi^2$ showed a statistically significant increase. When using 20 Newsgroups, χ^2 showed a decrease, but it was not statistically significant. When using Reuters-21578, all of the feature selection methods showed a statistically significant increase, except for MI, which showed a statistically significant decrease. CPD showed the largest increase when using OHSUMED, and tied with IG when using 20 Newsgroups. When using Reuters-21578, OR showed the largest increase followed closely by IG and CPD, which ranked second and a very close third, respectively.

In comparison to the other feature selection methods, CPD appears to perform competitively according to the F-measure value, consistently showing statistically significant increases and having the largest F-measure in four out of six text categorization tasks. However, to this point, the relative performance of the individual feature selection methods has only been statistically verified against the base case where no feature selection method is used. A comparison of the relative performance of CPD to that of the other feature selection methods is shown in Table 6.

Table 6: Comparison of CPD to the other feature selection methods

Text Corpus	Feature Selection	SVM	Naive Bayes
OHSUMED	χ^2	+	+
	IG	+	+
	DF	+	+
	MI	+	+
	OR	+	+
	$S\text{-}\chi^2$	+	+
20 Newsgroups	χ^2	+	+
	IG	+	o
	DF	+	+
	MI	+	+
	OR	+	+
	$S\text{-}\chi^2$	+	+
Reuters-21578	χ^2	+	+
	IG	o	o
	DF	+	+
	MI	+	+
	OR	-	o
	$S\text{-}\chi^2$	+	+

In Table 6, the *Feature Selection* column describes the feature selection methods to which CPD is being compared, and the *SVM* and *Naive Bayes* columns describe whether the F-measure for CPD is statistically significantly different from that of the other feature selection methods when using the SVM and Naive Bayes classifiers, respectively. The plus sign (i.e., +), circle (i.e., o), and minus sign (i.e., -) in these columns represent a statistically significant increase, no statistically significant difference, and a statistically significant decrease, respectively, in the F-measure value for CPD from that of the corresponding feature selection method. For example, we saw previously in Table 4 that the F-measures for CPD and IG are 0.900 and 0.867, respectively. In Table 6, the F-measure for CPD is shown to be statistically significantly different from that for IG. Specifically, the F-measure value for CPD represents a statistically significant increase from that for IG. Again, the paired Student's t-test was used to determine statistical significance using a 95% level of significance (i.e., $\alpha = 0.05$).

In Table 6, when using the SVM classifier, the F-measure values for CPD represent a statistically significant increase from all the other feature selection methods, regardless of the text corpus used, except for IG and OR when using Reuters-21578, where there is no statistically significant difference

and a statistically significant decrease, respectively. Similar results are shown when using the Naive Bayes classifier, except for IG when using 20 Newsgroups and IG and OR when using Reuters-21578, where there is no statistically significant difference.

5 Conclusion and Future Work

We introduced and evaluated a new feature selection method for text categorization tasks called CPD (categorical proportional difference). Experimental results showed that CPD outperformed other frequently studied feature selection methods in four out of six text categorization tasks using the OHSUMED, 20 Newsgroups, and Reuters-21578 text corpora.

Future work will focus on expanding the scope of the experiments to include additional classifiers and to utilize larger datasets with a greater number of features and categories. In addition, we will also attempt to identify distinct statistical differences between corpora to better understand the performance of a particular combination of classifier and feature selection method. For example, CPD was not the best performer on the Reuters-21578 dataset. Summary statistics for this dataset showed that it had a much lower percentage of words occurring in a single category. This suggests that measuring when a word and category occur together, and when neither the word nor category occur, values not considered by CPD, may be necessary for maximizing classifier performance. Finally, we will investigate whether statistical properties of a dataset can be used to predict the affect of feature selection on the dataset. For example, preliminary results have suggested that ratios based upon the number of overlapping and non-overlapping words across categories, and the distribution of words across categories show a surprising correlation to the size of the increase in accuracy that can be expected.

References

- Forman, G. (2003), 'An extensive empirical study of feature selection metrics for text classification', *Journal of Machine Learning Research* **3**, 1289–1305.
- Forman, G. (2008), Feature selection for text classification, in H. Liu and H. Motoda, eds, 'Computational Methods of Feature Selection', Chapman and Hall / CRC, pp. 257–276.
- Galavotti, L., Sabastiani, F. and Simi, M. (2000), Experiments on the use of feature selection and negative evidence in automated text categorization, in 'Proceedings of the 4th European Conference on Research and Advanced Technology for Digital Libraries (ECDL'00)', Lisbon, Portugal, pp. 59–68.
- Han, E.-H., Karypis, G. and Kumar, V. (2001), Text categorization using weight adjusted k-nearest neighbor classification, in 'Proceedings of the 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'01)', Hong Kong, China, pp. 53–65.
- Joachims, T. (1998), Text categorization with support vector machines: Learning with many relevant features, in 'Proceedings of the 10th European Conference on Machine Learning (ECML'98)', Chemnitz, Germany, pp. 137–142.
- Kim, S.-B., Han, K.-S., Rim, H.-C. and Myaeng, S. (2006), 'Some effective techniques for naive bayes

- text classification', *IEEE Transactions on Knowledge and Data Engineering* **18**(11), 1457–1466.
- Mladenic, D. and Grobelnik, M. (1999), Feature selection for unbalanced class distribution and naive bayes, in 'Proceedings of the 16th International Conference on Machine Learning (ICML'99)', Bled, Slovenia, pp. 258–267.
- Montanes, E., Diaz, I., Ranilla, J., Combarro, E. and Fernandez, J. (2005), 'Scoring and selecting terms for text categorization', *IEEE Intelligent Systems* **20**(3), 40–47.
- Newsgroups (1999), <http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html>.
- Ng, H., Goh, W. and Low, K. (1997), Feature selection, perceptron learning, and a usability case study for text categorization, in 'Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'97)', Philadelphia, Pennsylvania, pp. 67–73.
- OHSUMED (2005), <http://davis.wpi.edu/~xmdv/datasets/ohsumed.html>.
- Reuters-21578 (1997), <http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html>.
- Sebastiani, F. (2002), 'Machine learning in automated text categorization', *ACM Computing Surveys* **34**(1), 1–47.
- Witten, I. and Frank, E. (2005), *Data Mining: practical machine learning tools and techniques (2nd Edition)*, Morgan Kaufmann.
- Yang, Y. and Pedersen, J. (1997), A comparative study on feature selection in text categorization, in 'Proceedings of the 14th International Conference on Machine Learning (ICML'97)', Nashville, U.S.A., pp. 412–420.

Structure-Based Document Model with Discrete Wavelet Transforms and Its Application to Document Classification

Supphachai Thaicharoen¹Tom Altman¹Krzysztof J. Cios²

¹ Department of Computer Science and Engineering,
University of Colorado Denver, Campus Box 109,
PO Box 173364, Denver, CO 80217-3364, U.S.A.

Email: supphachai.thaicharoen@email.cudenver.edu, tom.altman@ucdenver.edu

² Virginia Commonwealth University,
Richmond, VA 23238, U.S.A.;
IITiS PAN, Poland.
Email: kcios@vcu.edu

Abstract

Term signal is an existing text representation that depicts a term as a vector of frequencies of occurrences in a number of user-defined partitions of a document. Although term signal augments the traditional vector space model with patterns of term occurrences, its document division is not coherent with the actual logical structure of a document. In this paper, we propose a novel document model, termed Structure-Based Document Model with Discrete Wavelet Transforms (SDMDWT), that exploits the structural information of documents and mathematical transforms for document representation. The proposed SDMDWT model enhances the existing term signal concept by additionally taking into consideration document's structural information during document division. We evaluated the proposed model on two different domains of standard data sets, WebKB 4-Universities and TREC Genomics 2005, using Support Vector Machines binary classification. The experimental results show that using our SDMDWT model for document representation demonstrates promising improvements of classification performances over existing document models.

1 Introduction

Document representation is one of the important tasks in text mining particularly for document classification and clustering. Its traditional approach is based on the "bag of words" or vector space model (VSM) where a document is represented by a vector of weights of unique terms selected from a data set. Weights are typically computed from frequency of term occurrences either within a document (term frequency) or across a data set (document frequency), or both.

In addition to the vector space representation, Park et al. proposed a concept of term signal that takes into account both frequency information and patterns of term occurrences in a document (Park et al. 2004, 2002a,b, Park, Palaniswami & Ramamohanarao 2005, Park & Ramamohanarao 2004, Park, Ramamohanarao & Palaniswami 2005). Term signal is a representation that describes frequency of a term

in physical locations of a document. It augments the traditional vector space model with patterns of term occurrences. With term signal, a document is first divided into a number of partitions based on a sequence and the number of terms in the document. Then, a term is represented as a vector of frequencies of term occurrences in those partitions. Finally, a document, consisting of a number of chosen terms, is represented as a vector of term signals. Park et al. additionally used a number of mathematical transforms such as Cosine Transforms, Fourier Transforms and Discrete Wavelet Transforms on their document representation, and computed document ranking based on query terms. Pryczek and Szczepaniak applied this term signal concept to document classification using Fourier Transforms (Pryczek & Szczepaniak 2006). Using the term signal with mathematical transforms for document representation was shown to be better than the traditional vector space representation for information retrieval in Park et al.'s and for document classification in Pryczek and Szczepaniak's studies. In this paper, we used document representation model based on this term signal concept with Discrete Wavelet Transforms as one of baseline models, and referred to this model as the *Spectral Space Model with Discrete Wavelet Transforms* (SPSMDWT).

Another method for enhancing document model is by additionally including structural information of documents into document representation. With the increase of publicly available full-text databases such as PubMed Central¹ and the widespread uses of semi-structure documents such as XML and HTML pages, the document structural approach became increasingly studied. Hakenberg et al. exploited structural information of full-text biomedical articles by assigning different weights to term occurrences in different sections, which resulted in performance improvements on document classification from the baseline classifier (Hakenberg et al. 2005). Denoyer and Gallinari proposed a Bayesian network model for semi-structure document classification that can be used to handle structural information and different types of document content (Denoyer & Gallinari 2004). In their study, a document is viewed as a tree where each node is corresponding to a document component, and links between two nodes represent dependencies or relations between document components. Model parameters are learned from training documents for each class, and a Bayesian network model is then built for each document. Based on experimental results, their proposed generative models were superior to the models that did not take into account the structural information of documents. Bratko and

Copyright ©2008, Australian Computer Society, Inc. This paper appeared at the Seventh Australasian Data Mining Conference (AusDM 2008), Glenelg, Australia. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 87, John F. Roddick, Jiuyong Li, Peter Christen and Paul Kennedy, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

¹<http://www.pubmedcentral.nih.gov/>

Filipić investigated a number of document models, named tagging, splitting and stacking, that utilize document structural information for document categorization (Bratko & Filipić 2006). For the tagging approach, the same words that occur in different sections of a document are treated as different words. For the splitting approach, texts in different document sections are modeled and evaluated separately, and the results are combined to give the final prediction. Finally, the stacking approach is similar to the splitting approach in that texts in different sections of a document are modeled and evaluated separately. The difference is that the final prediction is generated by a meta classifier that is built from the prediction results of classifications on different sections. Bratko and Filipić found that the stacking approach gave the best result.

In this paper, we propose a novel document representation model, termed Structure-Based Document Model with Discrete Wavelet Transforms (SDMDWT). The proposed SDMDWT model is built upon the concept of term signal by additionally taking into consideration document structure. With the SDMDWT model, a whole data set is first analyzed and the overall structure of its documents is captured. Then, original documents are pre-processed and converted into intermediate semi-structured documents in order to facilitate further processing. Finally, structured-based document model is constructed for each document and the Discrete Wavelet Transforms are applied. Our choice of using Discrete Wavelet Transforms for our model rather than other mathematical transforms is based on the latest work by Park et al. (Park, Ramamohanarao & Palaniswami 2005).

We evaluated our method on two different domains of standard data sets, WebKB 4-Universities and TREC Genomics 2005, using Support Vector Machines binary classification. Support Vector Machines (SVM) have been shown to work well with high-dimensional data and to be suitable to document classification (Joachims 1998). We utilized the VSM and SPSMDWT as baseline document models. The experimental results show that our SDMDWT model outperforms VSM and SPSMDWT on both standard data sets based on F-measure, micro-averaged F-measure and macro-averaged F-measure.

This paper is organized as follows. We present the technical background in Section 2, which covers the term signal concept, term and signal weighting schemes and Discrete Wavelet Transforms. Then, we describe our proposed document model, the data sets used in this paper and the pre-processing framework including a feature selection approach in Section 3. Next, we explain our evaluation method and provide experimental results in Section 4. Finally, we discuss experimental results and conclude this paper in Section 5.

2 Technical background

In this section, we describe background methods that are useful to understand the rationale behind our approach. We begin with the concept of term signal that our proposed method is built on, then weighting schemes for weighting terms and term signals and finally wavelet transforms that are used to transform term signals from the frequency domain to the wavelet domain.

2.1 Term signal

A term signal, introduced by Park et al. (Park et al. 2004, 2002a,b, Park, Palaniswami & Ramamohanarao

2005, Park & Ramamohanarao 2004, Park, Ramamohanarao & Palaniswami 2005), is a vector representation of terms that describes frequencies of term occurrences in particular partitions within a document. To construct a term signal, a document is first divided into a user-defined B number of sections. Then, a term signal t in a document d can be represented as a vector of physical sections by Equation 1.

$$s(t, d) = [f_{t,1,d}, f_{t,2,d}, \dots, f_{t,B,d}], \quad (1)$$

where $f_{t,b,d}$ is frequency of term t in section b of document d for $0 < b \leq B$. $f_{t,b,d}$ can also be considered as the b^{th} signal component of a term signal $s(t, d)$. For example, suppose that a document consisting of a sequence of 32 words is divided into 8 partitions. There will be 4 words per partition or bin. The document d can be graphically represented, see Figure. 1.

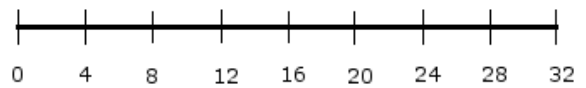


Figure 1: Document d with 8 partitions.

Accordingly, if a term t occurs once in each of the 2^{nd} , 3^{rd} , 10^{th} and 24^{th} positions in a document d , its term signal $s(t, d)$ can be represented by Equation 2 and depicted by Figure. 2.



Figure 2: Term signal t in document d .

$$s(t, d) = [2, 0, 1, 0, 0, 1, 0, 0]. \quad (2)$$

As shown in Figure 2, term t occurs two times in the 1^{st} bin, one time in the 3^{rd} bin and one time in the 6^{th} bin, respectively.

2.2 Weighting schemes

Weighting scheme is an assignment of numerical weights to terms in a vector space model. Weight of a term can be computed using a number of parameters such as term frequency, document frequency, document length, number of documents in a data set, etc. One of the most commonly used term-weighting schemes for document classification is $TF \cdot IDF$, which stands for term frequency multiplied by the inverse of document frequency. It can be formulated by Equation 3.

$$TF \cdot IDF = TF \times \lg(N/DF), \quad (3)$$

where TF is term frequency or frequency of term occurrences within a document, N is the total number of documents in a data set and DF is document frequency or frequency of term occurrences across a data set. The underlying assumption of $TF \cdot IDF$ weighting scheme is that terms that occur in many documents, high DF , are common and do not represent documents well. In contrast, terms that occur very often in a document, high TF , are considered

important features of the document. $TF \cdot IDF$ compensates between term frequency and document frequency.

For document representation using term signal, variations of $TF \cdot IDF$ weighting schemes can be used for weighting a term signal as described in (Park et al. 2002b). One of the variations, $PTF \cdot IDF$, is formulated by Equation 4.

$$PTF \cdot IDF = (1 + \lg(f_{t,d})) \left(\frac{f_{t,b,d}}{f_{t,d}} \right) \times \lg(N/DF), \quad (4)$$

where $f_{t,d}$ is frequency of occurrences of term t in document d and $f_{t,b,d}$ is frequency of occurrences of term t in partition b of document d .

2.3 Discrete Wavelet Transforms

A wavelet is a mathematical function in time/space domain, which can be expressed by Equation 5.

$$\psi_{s,l}(t) = 2^{s/2} \psi(2^s t - l), \quad (5)$$

where s is a dilation or scaling parameter, l is a translation or time-/space-location parameter and $s, l \in \mathbb{Z}$. For any function $f(t) \in L^2(\mathbb{R})$ and for which $\psi_{s,l}(t)$ forms an orthonormal basis for the space of signals of interest (in this case, $f(t)$), a wavelet transform of $f(t)$ can be computed by Equation 6.

$$\Psi(s, l) = \langle f(t), \psi_{s,l}(t) \rangle = \int_{-\infty}^{\infty} f(t) \psi_{s,l}^*(t) dt, \quad (6)$$

where $\psi_{s,l}^*(t)$ is a complex conjugate of $\psi_{s,l}(t)$. Wavelet transform can be described by a concept of multi-resolution analysis. Multi-resolution analysis is the decomposition of a signal into sub-signals of different resolutions or scales. In each step of multi-resolution analysis, a signal is decomposed into two sub-signals, approximation and detail. The approximation sub-signal is expressed by a linear combination of scaling functions $\varphi(t)$, and the detail sub-signal is represented by a linear combination of wavelet functions $\psi(t)$.

The scaling function is a finite energy function in $L^2(\mathbb{R})$ and is defined by Equation 7.

$$\varphi_{s,l}(t) = 2^{s/2} \varphi(2^s t - l), \text{ where } s, l \in \mathbb{Z}. \quad (7)$$

In multi-resolution analysis, the subspace spanned by the scaling function must satisfy the following properties.

$$V_s \subset V_{s+1} \text{ for all } s \in \mathbb{Z},$$

$$V_{-\infty} = \{0\} \text{ and } V_{\infty} = L^2$$

and

$$\{0\} \leftarrow \dots \subset V_{-1} \subset V_0 \subset V_1 \subset \dots \rightarrow L^2.$$

The wavelet function is also a finite energy function in $L^2(\mathbb{R})$ and is defined by Equation 8.

$$\psi_{s,l}(t) = 2^{s/2} \psi(2^s t - l), \text{ where } s, l \in \mathbb{Z}. \quad (8)$$

The subspace spanned by the wavelet function, W_s is an orthogonal complement of V_s in V_{s+1} . In other words, $V_s \perp W_s$ and $V_{s+1} = V_s \oplus W_s$, which leads to the following properties.

$$L^2 = \dots \oplus W_{-1} \oplus W_0 \oplus W_1 \oplus \dots$$

and

$$W_{-\infty} \oplus \dots \oplus W_{-1} = V_0.$$

Let V_s be a subspace spanned by scaling functions, W_s be a subspace spanned by wavelet functions, $V_0 \subset V_1 \subset \dots \subset L^2$ and $V_s = V_{s-1} \oplus W_{s-1}$, any function $f(t) \in L^2(\mathbb{R})$, where $L^2(\mathbb{R}) = V_0 \oplus W_0 \oplus W_1 \oplus W_2 \oplus \dots$, can be mathematically expressed by Equation 9.

$$f(t) = \sum_{l=-\infty}^{\infty} a_l \varphi_l(t) + \sum_{s=0}^{\infty} \sum_{l=-\infty}^{\infty} d_{s,l} \psi_{s,l}(t). \quad (9)$$

The first term is mapped to the approximation sub-signal, and the second term is referred as the detail sub-signal. The coefficients a_l and $d_{s,l}$ are called discrete wavelet transforms, which can be computed by Equations 10 and 11.

$$a_l = \langle f(t), \varphi_{s,l}(t) \rangle = \int f(t) \varphi_{s,l}^*(t). \quad (10)$$

$$d_{s,l} = \langle f(t), \psi_{s,l}(t) \rangle = \int f(t) \psi_{s,l}^*(t). \quad (11)$$

Discrete wavelet transforms can be efficiently calculated by using the filter bank tree-structured algorithm. The filter bank tree of discrete wavelet transforms of a signal $f(t)$ can be recursively expressed by Equation 12.

$$\begin{aligned} f & \xrightarrow{DWT^1} A^1 + D^1 \\ & \xrightarrow{DWT^2} A^2 + D^2 + D^1 \\ & \xrightarrow{DWT^3} A^3 + D^3 + D^2 + D^1 \\ & \dots \\ & \xrightarrow{DWT^s} A^s + D^s + D^{s-1} + \dots + D^1, \end{aligned} \quad (12)$$

where A^s represents an approximation sub-signal and D^s corresponds to a detail sub-signal in the s^{th} level of transforms of discrete wavelet transforms of the signal $f(t)$, respectively.

As described by Equations 10 and 11, discrete wavelet transform coefficients are computed by inner products between the signal itself and the scaling/wavelet functions. For example, the 1-levels of Haar scaling and Haar wavelet signals are defined by Equations 13 and 14.

$$V_1^1 = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0, 0, \dots, 0 \right), \quad (13)$$

$$V_2^1 = \left(0, 0, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0, 0, \dots, 0 \right),$$

$$V_3^1 = \left(0, 0, 0, 0, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0, 0, \dots, 0 \right),$$

...

$$V_{N/2}^1 = \left(0, 0, \dots, 0, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right).$$

$$W_1^1 = \left(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, 0, 0, \dots, 0 \right), \quad (14)$$

$$\begin{aligned}
W_2^1 &= (0, 0, \frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}, 0, 0, \dots, 0), \\
W_3^1 &= (0, 0, 0, 0, \frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}, 0, 0, \dots, 0), \\
&\dots \\
W_{N/2}^1 &= (0, 0, \dots, 0, \frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}).
\end{aligned}$$

Hence, for a discrete signal $f(t) = (4, 6, 10, 12, 8, 6, 5, 5)$, its 1-level Haar transform can be computed by Equations 15 and 16, and is conclusively represented by Equation 17.

$$\begin{aligned}
A_1^1 &= \langle f(t), V_1^1 \rangle = 5\sqrt{2}, \\
A_2^1 &= \langle f(t), V_2^1 \rangle = 11\sqrt{2}, \\
A_3^1 &= \langle f(t), V_3^1 \rangle = 7\sqrt{2}, \\
A_4^1 &= \langle f(t), V_4^1 \rangle = 5\sqrt{2}.
\end{aligned} \tag{15}$$

$$\begin{aligned}
D_1^1 &= \langle f(t), W_1^1 \rangle = -\sqrt{2}, \\
D_2^1 &= \langle f(t), W_2^1 \rangle = -\sqrt{2}, \\
D_3^1 &= \langle f(t), W_3^1 \rangle = \sqrt{2}, \\
D_4^1 &= \langle f(t), W_4^1 \rangle = 0.
\end{aligned} \tag{16}$$

$$f(t) \xrightarrow{H^1} (5\sqrt{2}, 11\sqrt{2}, 7\sqrt{2}, 5\sqrt{2} | -\sqrt{2}, -\sqrt{2}, \sqrt{2}, 0). \tag{17}$$

The numerical example above was excerpted from (Walker 2008). Interested readers can further study wavelets from several useful resources such as (Burrus et al. 1998), (Goswami & Chan 1999), (Mix & Olejniczak 2003) and (Walker 2008).

3 Data and methods

3.1 Data

WebKB 4-Universities. The WebKB 4-Universities data set² is a collection of 8282 web pages collected from computer science departments of several universities by the World Wide Knowledge Base project of the Carnegie Mellon text learning group in January 1997. The web pages in the data set are manually classified into the following 7 categories, student, faculty, staff, course, project, department and other. Each category contains web pages gathered from 4 main universities, Texas, Washington, Wisconsin and Cornell, and the remaining pages collected from other universities. Four most populous categories, student, faculty, course and project, are used in this paper, which accounts for 4199 web pages.

TREC Genomics 2005. The TREC Genomics 2005 data set is a corpus of full-text documents in SGML format. The data set is a collection of mouse genome articles from three journals, Journal of Biological Chemistry (JBC), Journal of Cell Biology (JCB), and Proceedings of the National Academy of Science (PNAS), over a two-year (2002-2003) period. There are four major types of articles in the data set – Alleles of mutant phenotypes, Embryologic gene expression, Gene Ontology and Tumor biology, which are corresponding to the following four class labels, A, E, G and T, respectively. TREC Genomics 2005 data set consists of 5837 training documents and

6403 testing documents. Among the 5837 training documents, 338 documents are related to Alleles (A), 81 documents to Gene Expression (E), 462 documents to Gene Ontology (G) and 36 documents to Tumor (T). In 6403 testing documents, 332 documents are assigned to class A, 105 documents to class E, 518 documents to class G and 20 documents to class T. The remaining documents do not have any class labels associated with them.

3.2 The proposed SDMDWT model and its preprocessing framework

The SDMDWT document model is an enhancement of the term signal proposed by Park et al. (Park et al. 2004, 2002a,b, Park, Palaniswami & Ramamohanarao 2005, Park & Ramamohanarao 2004, Park, Ramamohanarao & Palaniswami 2005) such that each bin is mapped to a document component in a document based on the captured document structure rather than is derived from computation. We call our term signal based on document structure the *structure-based term signal*. The structure-based term signal is defined in Definition 1.

Definition 1. *Structure-based term signal.*

Structure-based term signal is a vector of frequencies of term occurrences in different components of a document, where components are derived from the actual document structure. The structure-based term signal of a term t in a document d is defined by Equation 18.

$$st(t, d) = [f_{t,c_1,d}, f_{t,c_2,d}, \dots, f_{t,c_n,d}], \tag{18}$$

where $f_{t,c_1,d}, f_{t,c_2,d}, \dots, f_{t,c_n,d}$ are corresponding to the frequencies of term t in document components c_1, c_2, \dots, c_n of document d , respectively.

The key differences between our proposed structure-based term signal (Equation 18) and the existing term signal (Equation 1) are (i) the number of components of our model is derived from the actual document structure such as the number of sections, but that of the existing term signal is defined by users and (ii) the length of each component in our model is based on the actual length of document components. However, the length of each component in the existing term signal model is computed from the total number of terms in a document divided by the user-defined number of components.

Definition 2. *Structure-based document model.*

According to the structure-based term signal in Definition 1, a document can be represented by a vector of structure-based term signals, which is defined by Equation 19.

$$d = [st(t_1, d), st(t_2, d), \dots, st(t_n, d)], \tag{19}$$

where t_1, t_2, \dots, t_n are term features selected in the feature selection step, and $st(t_1, d), st(t_2, d), \dots, st(t_n, d)$ are the structure-based term signals of terms t_1, t_2, \dots, t_n in document d , respectively.

The SDMDWT preprocessing framework. The preprocessing framework for constructing SDMDWT model is summarized as follows.

1. *Capture the document structure of a data set:* The first step for constructing the proposed SDMDWT model is to analyze documents in the data set and to capture the common characteristics of their document structure. As mentioned by Bratko and Filipič (Bratko & Filipič 2006),

²<http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/>

how the structure of a document is captured depends on particular semantics of document structure and its perceived relevance to the text mining task that is document classification in this paper. For WebKB 4-Universities data set, we divided each document into the following 2 components, heading text (H) and non-heading text (N). We used the `<h#>` and `</h#>` HTML tags for distinguishing between the two components. The heading text is text enclosed by the `<h#>` and `</h#>` HTML tags, and the non-heading text is text that is not. The “#” inside each `<h#>` tag represents the level number of the `<h>` tag. We also included text enclosed by `<title>` and `</title>` into the heading text component. For TREC Genomics 2005 biomedical data set, we partitioned each document into the following 7 components based on the actual logical organization of documents, *Title* (T), *Abstract* (A), *Introduction* (I), *Method* (M), *Result* (R), *Conclusion* (C) and *Other* (O). Note that these document divisions are by no means definite. They depend on the data set used and how the document structure is captured. For example, for full-text biomedical documents, *Title* and *Abstract* may be combined into one component instead of being independent components.

2. *Collect variants of component labels and construct a mapping table:*

Although documents in the same domain tend to have similar structure, it is uncommon that their component labels are different. For example, in the WebKB 4-Universities HTML pages, the HTML heading `<h>` tags have several levels such as `<h1>`, `<h2>`, `<h3>`, `<h4>`, etc. Moreover, in TREC Genomics 2005 data set, the *Method* components of various papers are labeled as “Methods”, “Patients and methods”, “Experimental procedures”, etc. Therefore, after collecting all variants of component labels in the data set, a mapping table was constructed for mapping variants of component labels to their corresponding user-defined component names. Table 1 gives an example of name variants of the *Method* section in TREC Genomics 2005 data set.

3. *Pre-process documents and perform feature selection:*

For WebKB 4-Universities data set, we utilized a combination of JTiny³ and Java regular expression to clean up and parse the original HTML documents into the semi-structured XML documents. We used simple words as features and did not perform word stemming and stop-word removal on this data set. For TREC Genomics 2005 data set, we implemented an SGML Java parser to parse original SGML documents into the semi-structured XML documents. We utilized Lingpipe⁴ to break texts into sentences and used Genia Tagger⁵ to perform part-of-speech tagging and to detect terms, phrases and biological entities. We used phrases as features and removed those that are in the PubMed stop word list⁶. Word stemming using Porter stemmer (Porter 1997) was also carried out. For both data sets, we ranked terms using Information Gain based on (Yang & Pedersen 1997), which is formulated by Equation 20, and then selected the

Table 1: An example of the mapping tables.

Name variations of the <i>Method</i> component	
	methods
	method
	experimental procedures
	experimental procedure
	experiemntal procedures
	experimantal procedures
	experimental procecedures
	experimental procodures
	experimental approach
	experimental approaches
	experimental results and interpretation
	materials and methods
	material and methods
	materials
	mateials and methods
	matelials and methods
	materials methods
	metrials and methods
	methods and materials
	methods and methods
	patients and methods
	subjects and methods
	computational methods
	model and methods
	research design and methods
	media and materials

top n terms to be used in experiments. Note that we cleaned and parsed texts in the original documents into XML documents with organized components as intermediate representation in order to facilitate the construction of structure-based term signals and document models in the next steps.

$$\begin{aligned}
 IG(t) = & - \sum_{i=1}^m P_r(c_i) \log P_r(c_i) \\
 & + P_r(t) \sum_{i=1}^m P_r(c_i|t) \log P_r(c_i|t) \\
 & + P_r(\bar{t}) \sum_{i=1}^m P_r(c_i|\bar{t}) \log P_r(c_i|\bar{t}),
 \end{aligned} \quad (20)$$

where $P_r(c_i)$ is the probability of class c_i , $P_r(t)$ is the probability of term t , $P_r(c_i|t)$ is the probability of a document that contains term t and has class c_i , and finally $P_r(c_i|\bar{t})$ is the probability of a document that does not contain term t and has class c_i .

4. *Represent each term using the proposed structure-based term signal:*

After pre-processing documents and selecting features, we constructed the structured-based term signal for each selected term. For WebKB 4-Universities data set, since each WWW page is divided into 2 components, the heading text component (H) and non-heading text component (N). As a result, each selected term could be constructed by Equation 21.

$$st(t, d) = [f_{t,CH,d}, f_{t,CN,d}], \quad (21)$$

For TREC Genomics 2005 data set, each selected term was represented by Equation 22. Since

³<http://jtidy.sourceforge.net>

⁴<http://alias-i.com/lingpipe>

⁵<http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger/>

⁶<http://www.ncbi.nlm.nih.gov/books/bv.fcgi?highlight=\&stopwords&rid=helppubmed.table.pubmedhelp.T43>

Discrete Wavelet Transforms requires a signal length to be a power of two, we added an additional zero-value component to the structure-based term signal.

$$st(t, d) = [f_{t,CT,d}, f_{t,CA,d}, f_{t,CI,d}, f_{t,CM,d}, f_{t,CR,d}, f_{t,CC,d}, f_{t,CO,d}, 0], \quad (22)$$

5. *Apply pre-weighting and Discrete Wavelet Transforms to each structured-based term signal:*
In this step, we applied the pre-weighting in Equation 4 and Haar Discrete Wavelet Transforms explained in the technical background section to each structured-based term signal in each document.
6. *Construct the structured-based document model:*
Finally, for both data sets, each document was constructed as a vector of structured-based term signals as defined by Equation 19.

4 Experiments and results

WebKB 4-Universities. We evaluated our SDMDWT model using an open source LIBSVM (Chang & Lin 2001) in Weka (Witten & Frank 2005), with C-SVC, linear kernel and 0.01 tolerance as SVM parameter values. We generated a sub data set for each class with one-against-all strategy. For example, if documents in the category “student” is specified as positive documents, all other documents in the data set will be labeled as negative documents. For each sub data set, we performed 10-cross validation with binary classification and then collected results.

TREC Genomics 2005. We evaluated our SDMDWT model using an open source LIBSVM library (Chang & Lin 2001) with C-SVC, linear kernel and 0.001 tolerance as parameter values. We also used one-against-all strategy. For documents that belong to more than one class, if one of their classes is the positive class under consideration, then we assign positive labels to them. For each class, we performed a binary classification with train/test sub data sets, and then collected results.

For both data sets, we utilized F-measure, micro-averaged F-measure and macro-averaged F-measure as performance measures.

F-measure or F1-measure is a combination of Precision and Recall, defined by Equation 23.

$$F\text{-measure} = \frac{2 * Precision * Recall}{Precision + Recall}. \quad (23)$$

Precision and Recall can be calculated by Equations 24 and 25.

$$Precision = \frac{TP}{TP + FP}, \quad (24)$$

$$Recall = \frac{TP}{TP + FN}, \quad (25)$$

where TP, TN, FN and FP are the number of true positives, true negatives, false negatives and false positives, respectively.

The micro-average and macro-average are performance averages across multiple categories. As described in (Yang 1999), the micro-averaged performance is viewed as a per-document average because it gives equal weight to every document. To compute a micro-averaged performance, a global contingency

Table 2: WebKB 4-Universities: Micro-averaged F-measure.

F^a/M^b	SDMDWT	SPSMDWT	VSM
500	0.928794	0.924501	0.887868
1000	0.926999	0.923317	0.894601
1500	0.929770	0.927415	0.900745
2000	0.928939	0.924868	0.900231
2500	0.929213	0.921566	0.896142
5000	0.924592	0.919918	0.892918
7500	0.922485	0.921965	0.884373

^aNumber of features

^bDocument model

Table 3: WebKB 4-Universities: Macro-averaged F-measure.

F^a/M^b	SDMDWT	SPSMDWT	VSM
500	0.923765	0.917390	0.867331
1000	0.922871	0.916421	0.874013
1500	0.926470	0.921714	0.884561
2000	0.923818	0.918427	0.883605
2500	0.923653	0.914590	0.880300
5000	0.918141	0.914226	0.873226
7500	0.915144	0.914865	0.863115

^aNumber of features

^bDocument model

table is constructed, whose cell value is the sum of the corresponding cell in each contingency table of each class. For example, the number of true positives in the global contingency table is the sum of the number of true positives from all contingency tables of all classes. Then, a micro-averaged performance such as micro-averaged Precision or micro-averaged Recall is computed from this global contingency table. In this paper, we use the micro-averaged F-measure, which can be computed by Equation 26.

$$\text{Micro-averaged F-measure} = \frac{2 * \text{Micro-averaged Precision} * \text{Micro-averaged Recall}}{\text{Micro-averaged Precision} + \text{Micro-averaged Recall}}. \quad (26)$$

The macro-averaged performance is considered per-category average because it gives equal weight to every class. It can be computed by the sum of performance from each class divided by the total number of classes. The macro-averaged F-measure can be computed by Equation 27.

$$\text{Macro-averaged F-measure} = \frac{\sum_{i=1}^c F\text{-measure}_i}{c}, \quad (27)$$

where $F\text{-measure}_i$ is the F-measure of class i , and c is the number of classes.

4.1 WebKB 4-Universities

According to Tables 2 and 3, we can conclude that our SDMDWT model is better than SPSMDWT that is the document model based on the original term signal concept, and it distinguishably outperforms VSM for all different numbers of features based on micro-averaged and macro-averaged F-measures.

In addition, in the Faculty and Project categories, our SDMDWT model is clearly superior to SPSMDWT and VSM models based on F-measure. The performance comparisons of document models for these two classes are shown in Figures 3 and 4, accordingly.

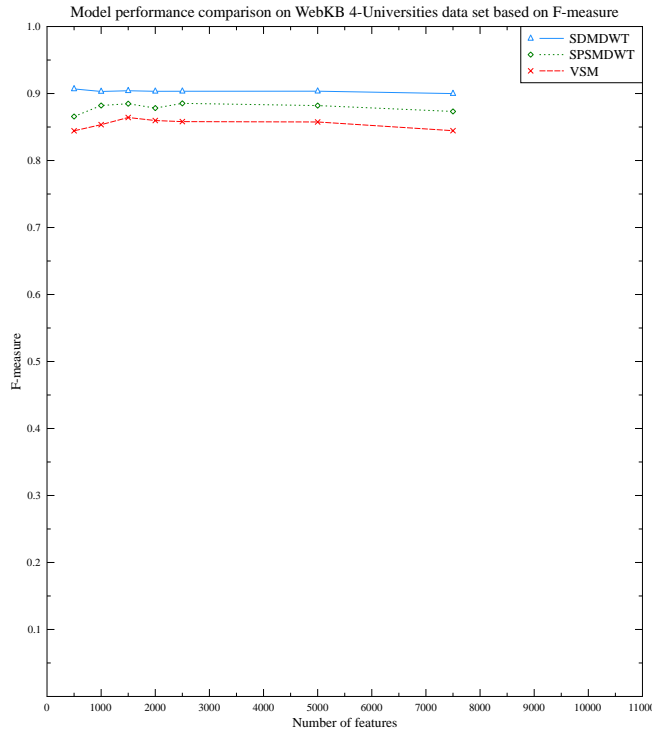


Figure 3: WebKB 4-Universities: Performance comparison based on F-measure when the Faculty category is positive category.

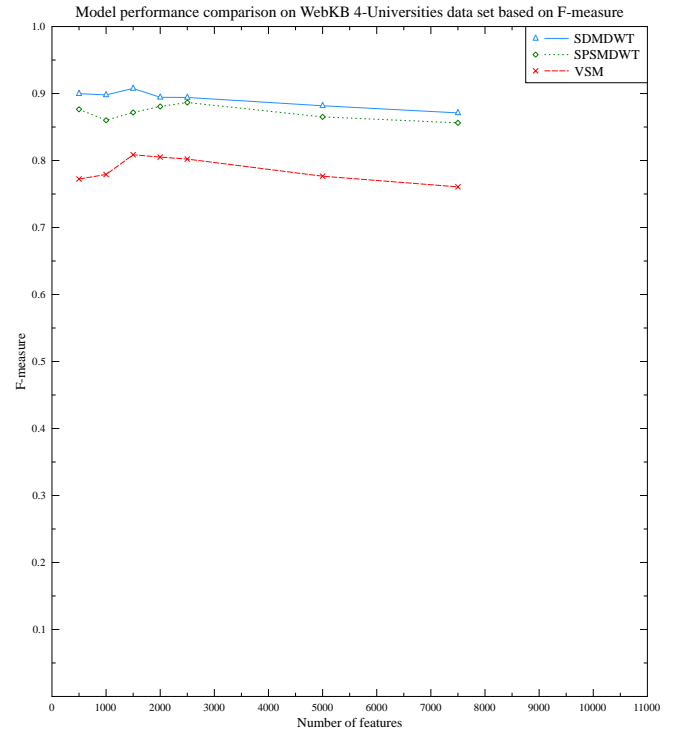


Figure 4: WebKB 4-Universities: Performance comparison based on F-measure when the Project category is positive category.

Table 4: TREC Genomics 2005: Micro-averaged F-measure.

F^a/M^b	SDMDWT	SPSMDWT	VSM
7500	0.244156	0.168052	0.223663
8500	0.213144	0.156863	0.180879
10000	0.198377	0.138127	0.100000

^aNumber of features

^bDocument model

4.2 TREC Genomics 2005

Based on Tables 4 and 5, we can conclude that our SDMDWT model is superior to SPSMDWT and VSM models based on the micro-averaged and macro-averaged F-measures.

Moreover, our SDMDWT model gives distinct performance improvements based on F-measure compared to SPSMDWT and VSM model where class Alleles (A) is considered the positive class, which is shown by Figure 5.

5 Conclusions

In this paper, we proposed a novel document representation model, termed Structure-Based Document Model with Discrete Wavelet Transforms (SDMDWT), that exploits structural information of doc-

uments and Discrete Wavelet Transforms for document representation. The proposed model is built on the existing term signal concept that represents a pattern of term occurrences in different partitions of a document as a vector. The main difference between our SDMDWT and SPSMDWT models lies on the fact that our SDMDWT model takes into consideration structural information when partitioning a document. Accordingly, a document division of our SDMDWT model is more coherent with the actual logical document structure than that of SPSMDWT model. This inclusion of structural information into document representation allows further improvement of text mining tasks where document structure is concerned such as weighting document components differently. The pre-processing framework of the proposed SDMDWT document model can be divided into the following steps: (i) capturing document structure, (ii) collecting all various names of document component headings and constructing a mapping table, (iii) pre-processing documents and performing feature selection, (iv) representing each selected term using the structure-based term signal, (v) applying pre-weighting and Discrete Wavelet Transforms to each structure-based term signal and (vi) constructing the structure-based document models.

According to the experimental results on both TREC Genomics 2005 and WebKB 4-Universities data sets, using our SDMDWT model for document representation gives better classification performances (F-measure, micro-averaged F-measure and macro-averaged F-measure) than using the traditional vector space model (VSM) and the document model that is based on the original term signal concept (SPSMDWT). The clear performance improvements based on F-measure occur on the Faculty and Project categories of WebKB 4-Universities dataset and on the Alleles (A) category of TREC Genomics 2005 dataset, which are shown by Figures 3, 4 and 5, respectively. Therefore, we can conclude that struc-

Table 5: TREC Genomics 2005: Macro-averaged F-measure.

F^a/M^b	SDMDWT	SPSMDWT	VSM
7500	0.141999	0.104290	0.129638
8500	0.126299	0.098373	0.110393
10000	0.119382	0.087978	0.065065

^aNumber of features

^bDocument model

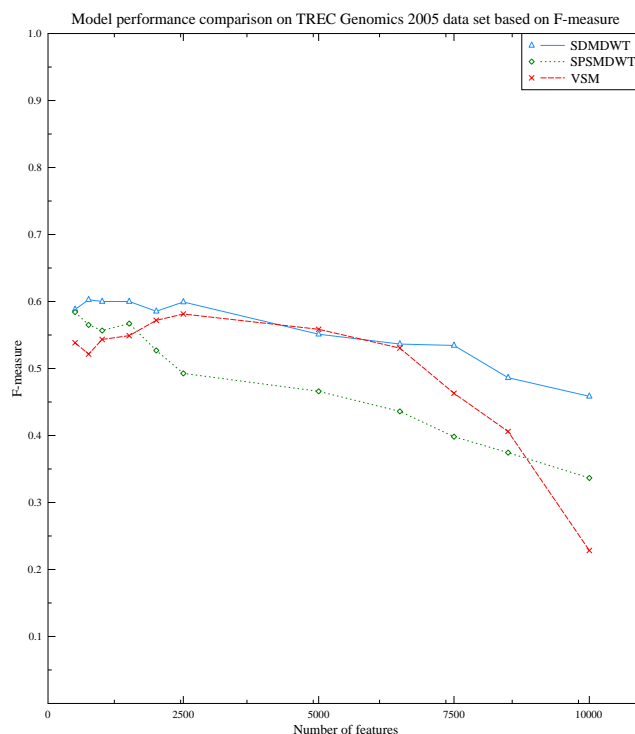


Figure 5: TREC 2005: Performance comparison based on F-measure when the Alleles (A) category is positive category.

tural information of documents can be incorporated into document representation to improve performance of document classification.

Lessons learned from this research study are that although several text documents in scientific or WWW domains are presented in the semi-structured formats such as XML, SGML and HTML, to be able to exploit structural information, a manual analysis is still required for capturing the common structural characteristics of documents. In addition, even with the same document structure, components in different documents are generally labeled with different names such as “Conclusions”, “Concluding remarks”, “Summary”, etc. As a result, to facilitate pre-processing, full-text documents (particularly in scientific domain) should be standardized on the labels and number of main components and should be presented in a semi-structured format such as XML with component labels used as element names.

For future work, our proposed technique and idea could possibly be applied to other types of text mining tasks for performance improvement such as document clustering and feature selection that additionally take into account document structure.

6 Acknowledgments

The authors thank all researchers at the Center for Computational Pharmacology at the University of Colorado Denver. Particularly, William Baumgartner for a number of useful discussions at the beginning of this work, and Dr. Lawrence Hunter for providing the data and other resources for this research. We also thank Sam Wheeler, System Administrator and Lab Manager at the College of Engineering and Applied Science, University of Colorado Denver, for his technical assistance.

References

- Bratko, A. & Filipič, B. (2006), ‘Exploiting structural information for semi-structured document categorization’, *Inf. Process. Manage.* **42**(3), 679–694.
- Burrus, C. S., Gopinath, R. A. & Guo, H. (1998), *Introduction to Wavelets and Wavelet Transforms: A Primer*, Prentice Hall.
- Chang, C.-C. & Lin, C.-J. (2001), *LIBSVM: a library for support vector machines*. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- Denoyer, L. & Gallinari, P. (2004), ‘Bayesian network model for semi-structured document classification’, *Inf. Process. Manage.* **40**(5), 807–827.
- Goswami, J. C. & Chan, A. K. (1999), *Fundamentals of Wavelets: Theory, Algorithms, and Applications*, John Wiley & Sons, Inc.
- Hakenberg, J., Rutsch, J. & Leser, U. (2005), Tuning text classification for hereditary diseases with section weighting, in ‘Proc International Symposium on Semantic Mining in Biomedicine, SMBM’, Hinxton, UK, pp. 34–37.
- Joachims, T. (1998), Text categorization with support vector machines: Learning with many relevant features, in ‘ECML ’98: Proceedings of the 10th European Conference on Machine Learning’, Springer-Verlag, London, UK, pp. 137–142.
- Mix, D. F. & Olejniczak, K. J. (2003), *Elements of wavelets for engineers and scientists*, Wiley-Interscience.
- Park, L. A. F., Palaniswami, M. & Ramamohanarao, K. (2002a), A new implementation technique for fast spectral based document retrieval systems, in ‘ICDM ’02: Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM’02)’, IEEE Computer Society, Washington, DC, USA, p. 346.
- Park, L. A. F., Palaniswami, M. & Ramamohanarao, K. (2002b), A novel web text mining method using the discrete cosine transform, in ‘PKDD ’02: Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery’, Springer-Verlag, London, UK, pp. 385–396.
- Park, L. A. F., Palaniswami, M. & Ramamohanarao, K. (2005), ‘A novel document ranking method using the discrete cosine transform’, *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(1), 130–135.
- Member-Laurence A. F. Park and Sr. Member-Marimuthu Palaniswami and Member-Kotagiri Ramamohanarao.
- Park, L. A. F. & Ramamohanarao, K. (2004), Hybrid pre-query term expansion using latent semantic analysis, in ‘ICDM ’04: Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM’04)’, IEEE Computer Society, Washington, DC, USA, pp. 178–185.
- Park, L. A. F., Ramamohanarao, K. & Palaniswami, M. (2005), ‘A novel document retrieval method using the discrete wavelet transform’, *ACM Trans. Inf. Syst.* **23**(3), 267–298.
- Park, L. A., Ramamohanarao, K. & Palaniswami, M. (2004), ‘Fourier domain scoring: A novel document ranking method’, *IEEE Transactions on Knowledge and Data Engineering* **16**(5), 529–539.

- Porter, M. F. (1997), 'An algorithm for suffix stripping', pp. 313–316.
- Pryczek, M. & Szczepaniak, P. S. (2006), 'On textual documents classification using fourier domain scoring', *wi* **0**, 773–777.
- Walker, J. S. (2008), *A Primer on Wavelets and Their Scientific Applications*, second edn, Chapman & Hall/CRC.
- Witten, I. H. & Frank, E. (2005), *Data Mining: Practical machine learning tools and techniques*, 2nd edn, Morgan Kaufmann, San Francisco.
- Yang, Y. (1999), 'An evaluation of statistical approaches to text categorization', *Inf. Retr.* **1**(1-2), 69–90.
- Yang, Y. & Pedersen, J. O. (1997), A comparative study on feature selection in text categorization, in 'ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning', Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 412–420.

Combining Structure and Content Similarities for XML Document Clustering

Tien Tran

Richi Nayak

Peter Bruza

Faculty of Information Technology
Queensland University of Technology
GPO Box 2434, Brisbane QLD 4001, Australia

Emails: {t4.tran, r.nayak, p.bruza}@qut.edu.au

Abstract

This paper proposes a clustering approach that explores both the content and the structure of XML documents for determining similarity among them. Assuming that the content and the structure of XML documents play different roles and importance depending on the use and purpose of a dataset, the content and structure information of the documents are handled using two different similarity measuring methods. The similarity values produced from these two methods are then combined with weightings to measure the overall document similarity. The effect of structure similarity and content similarity on the clustering solution is thoroughly analysed. The experiments prove that clustering of the text-centric XML documents based on the content-only information produces a better solution in a homogeneous environment, documents that derived from one structural definition; however, in a heterogeneous environment, documents that derived from two or more structural definitions, clustering of the text-centric XML documents produces a better result when the structure and the content similarities of the documents are combined with different strengths.

Keywords: XML, clustering, latent semantic kernel, vector space model.

1 Introduction

Over the past years, electronic documents in several formats, such as XML, HTML and XHTML, have been proposed to represent the textual content of the documents in a structural manner. For data representation and exchange, formatting in XML has emerged as a standard (Bray et al. 2004). With the continuous growth of the XML documents, data management issues, such as retrieval and storage of the large number of documents, have also arisen (Nayak et al. 2002). Clustering of these documents is one way of handling this issue. XML clustering is a task which can be applied to organize the massive amounts of XML documents into groups without the prior knowledge (Han & Kamber 2001); each group containing the documents that share similar characteristics. Clusters can be derived based on the content or based on the structural information of the XML documents. For example, clustering of XML documents based on the content is for dealing with XML datasets in a homogeneous environment, documents that use the same

structure to represent different topics or themes e.g. IEEE transactions. This type of clustering application is useful in information retrieval and document engineering. On the other hand, clustering of XML documents based on the structure is for dealing with XML datasets in a heterogeneous environment, documents that use different structures to represent the same information, such as, a purchase order has different representations according to its originator where its information may represent differently. This type of clustering application is useful in database indexing, data-warehouse, data integration and document engineering.

A number of XML clustering approaches has been proposed in recent years; however, there is still very little work on the clustering of semi-structure documents that effectively combines the content and the structure information of the XML documents for clustering, especially for XML datasets in the homogeneous environment. Assuming that the content and the structure of the XML documents play different roles and importance according to the use and purpose of an application, we propose an approach to cluster text-centric XML datasets, datasets in which the content is the most important feature in determining the document similarity, by calculating each of the content similarity and the structure similarity of a document separately, and then combining them with appropriate strengths, defined by the user, for document similarity. The structure similarity is determined by the commonality and co-occurrence of paths between document structures. A latent semantic kernel (Cristianini et al. 2002) is used to determine the semantic association within document contents.

The empirical analysis reveals that clustering of the text-centric XML datasets based on the content-only information produces a better solution in a homogeneous environment; however, in a heterogeneous environment, clustering of the text-centric XML datasets produces a better result when the structure and the content similarities of the documents are combined with different strengths. Our contributions are as follows: (1) Using Latent Semantic Kernel (LSK) for measuring the semantic associations of the textual content of XML documents, and; (2) Exploiting the semantic of the document contents and the commonality of the document structure for XML clustering.

1.1 Related Work

There has been a myriad of clustering approaches proposed in recent years. Some of these approaches (Kurgan et al. 2002, Shen & Wang 2003) discard the structural information of the XML documents and the similarity learning is based on the content-only information. However, a good clustering process should not discard the use of the structure since XML is popu-

larly known for its representation and storing of the structural content that can be easily processed by systems such as the databases.

Clustering approaches are varied according to the representation of the XML data such as tree-based, path-based, graph-based, etc. The method of calculating document similarity varies accordingly. The similarity matrix generated by these approaches usually becomes an input to a traditional clustering method such as the hierarchical agglomerative algorithm or the k-means algorithm (Han & Kamber 2001). Several approaches (Nierman & Jagadish 2002, Dalamagas et al. 2004) have been proposed to represent the XML documents as tree-based and use the tree edit distance to measure the similarity between the documents using the document structure. Lian et al. approach (Lian et al. 2004) represents the XML document as graph-based and measures the common set of nodes and edges appearing between the documents. To retain the structure information from the XML documents, some approaches (Jeong & Keun 2004, Leung et al. 2005, Jeong & Keun 2005) use the sequential pattern mining to extract the frequent paths from XML documents and then use them for clustering. XClust (Lee et al. 2002) introduces a complex computational technique to map the element similarity between the schemas by considering the semantics, immediate descendent and leaf-context information. Its purpose is to be used as the preprocessing stage for applications such as data integration.

The approaches which previously discussed consider only the structure information. Content mining has been well explored in area such as information retrieval where the content of the document can be represented as a vector space model (Salton & McGill 1983). Methods such as tf*idf weight (Salton & McGill 1983), feature reduction methods such as principal component analysis (Liu et al. 2004) and latent semantic analysis (Landauer et al. 1998) have been widely used to measure the similarity between a document to a query (Kim et al. 2005, Yang et al. 2005). The latent semantic analysis (Landauer et al. 1998) constructs a semantic space wherein terms and documents that are closely associated are placed near one another. This space reflects major associative patterns in the data and ignores less important patterns.

Recognizing the importance of the content with the structure of the XML documents, a number of approaches (Shen & Wang 2003, Kc et al. 2006, Yang et al. 2005) have been proposed to incorporate the content and the structure of the XML documents for clustering. Shen and Wang (2003) approach breaks the XML documents into a number of macro-path sequences where each macro-path contains the properties of an element such as its name, attributes, data types and textual content. A matrix similarity of the XML documents is then generated based on the macro-path similarity technique. The clustering of XML documents is performed based on the similarity matrix with the support of approximate tree inclusion and isomorphic tree similarity. Kc et al. (2006) uses the self-organizing maps (Kohonen 1990) for learning the structure of the XML documents. However when it attempts to use the self-organizing maps for including both the content and the structure of XML documents, it performs poorer than the structure-only clustering solutions on the INEX datasets. This shows that for certain datasets, using the structure and the content information together in the clustering process degrades the performance of the clustering solutions (Denoyer et al. 2006). Taking this into consideration when dealing with different datasets, our approach measures the content and the structure similarities separately, and then combines them with dif-

ferent strengths. This gives a relative importance to the structure and to the content according to the type of the datasets.

2 Overview of the Proposed Clustering Approach

Figure 1 illustrates the overview of the proposed clustering approach. The XML dataset is pre-processed to extract the content and the structural information. The content of the XML documents, here, refers to the textual data, and the structure is referring to the elements (or tags) which are used to structure the content. The content of a document is represented by a collection of unique terms after stop-word removal and stemming (Porter 1980). Stop-word is term that considered not to be important such as “is”, “or”, “a”, etc. Only the keywords of the content are used for the content similarity measure. Whereas, the structural information of the XML documents is represented as paths, containing element names in hierarchical order, which are used for structure similarity measure.

Both the content and the structure information are represented using the vector space model (Salton & McGill 1983). As the proposed approach addresses the problem of combining the structure and the content similarities for text-centric dataset, sophisticated structure measure is not required since the text-centric dataset is conformed to the same structural definition and various instances of the dataset do not vary much in their structure representations. The content is measured separately from the structure using a different method. The document similarity is measured by combining the structure similarity value and the content similarity value. The output of the document matching is a pair-wise document similarity matrix which contains the document similarity between each pair of XML documents in the dataset. This matrix is then used to cluster the dataset. The next section describes how the document similarity is measured in more detail.

3 Document Similarity Measure

The document similarity between two XML documents, d_x and d_y , is defined as:

$$\begin{aligned} docSim(d_x, d_y) = & (contSim(d_x, d_y) \times \lambda) \\ & + (structSim(d_x, d_y) \times (1 - \lambda)). \end{aligned} \quad (1)$$

The document similarity is a combination of the content similarity value and the structure similarity value. The λ , ranging from 0 to 1, is defined by the user to adjust the importance of the content similarity ($contSim$) and structure similarity ($structSim$). A pair-wise document similarity matrix is generated by computing the similarity between each pair of XML documents in the dataset using the document similarity measure as defined in equation 1. A clustering method such as k-means or hierarchical agglomerative can be applied to find clusters in the pair-wise document similarity matrix.

3.1 Structure Similarity Measure

The structure of an XML document relates to how the content in the XML document is structured. Information such as element names, data types, constraints, parents, ancestors, children, etc. can be used to discover the structural similarity between XML documents. To simplify the structure matching

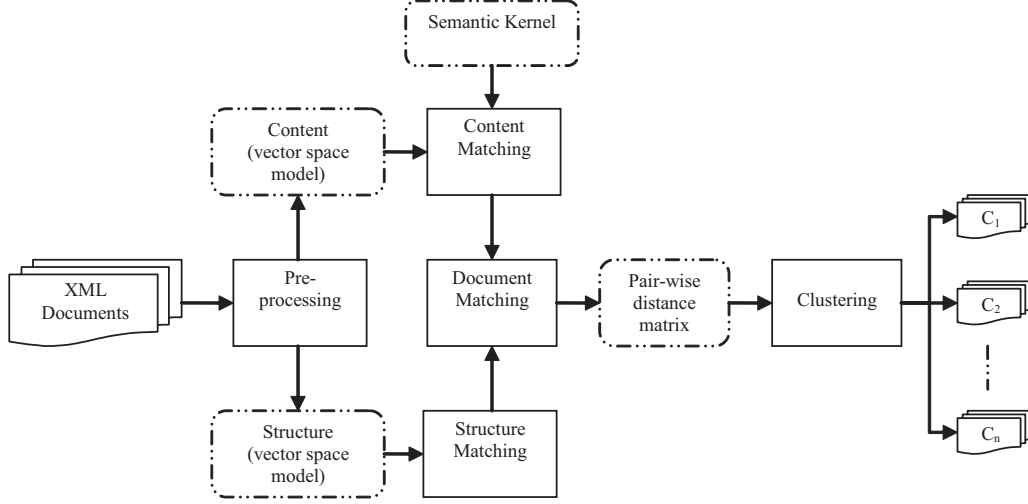
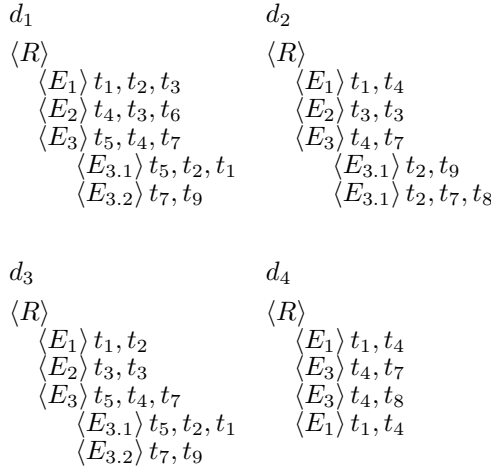


Figure 1: Overview of the proposed clustering approach.

Figure 2: Dataset D containing 4 XML documents.

process, only the element names, the most important property of the elements, are used for structure matching. The structure of an XML document is represented as a tree-based in which it is broken down into a collection of distinct paths. These paths are used to measure the structural distance between XML documents. Given a dataset of XML documents $\{d_1, d_2, \dots, d_n\}$, denoted by D , a set of distinct paths $\{p_1, p_2, \dots, p_f\}$, denoted by P , are extracted from D .

Definition 1 (Path). A path, p_i , contains element names from the root element to the leaf element. The leaf element is an element that contains the textual content.

Definition 2 (Structure Modeling). The structure of a document, d_i , is modelled as a vector $\{p_{i,1}, p_{i,2}, \dots, p_{i,f}\}$, where each element of the vector represents the frequency of a path in P that appears in the document.

Definition 3 (Structure Matching). Given two documents, d_x and d_y , and their corresponding vectors, $\{p_{x,1}, p_{x,2}, \dots, p_{x,f}\}$ and $\{p_{y,1}, p_{y,2}, \dots, p_{y,f}\}$ respectively. The distance between the two documents

Table 1: A matrix Y representing the structure information of the dataset.

path/doc	d_1	d_2	d_3	d_4
R/E_1	1	1	1	2
R/E_2	1	1	1	0
$R/E_3/E_{3.1}$	1	2	1	0
$R/E_3/E_{3.2}$	1	0	1	0
R/E_3	1	1	1	2

is computed using the Euclidean distance.

$$structSim(d_x, d_y) = \sqrt{\sum_{i=1}^f (p_{x,i} - p_{y,i})^2}. \quad (2)$$

The $structSim$ is normalised between 0 and 1.

Example. Let us assume a collection, D , containing 4 XML documents $\{d_1, d_2, d_3, d_4\}$, as shown in figure 2; element names in the documents are shown as embraced within brackets, $\langle R \rangle$ is the root element and $\langle E_i \rangle$ is the internal element or leaf element. The content of a document is denoted by T . The structure of a document is extracted and represented as a vector. The structures of all the documents in the dataset can be put together as a path-document matrix, $Y_{f \times n}$, where f is the number of distinct paths in P and n is the number of documents in D , as shown in table 1. Each cell in matrix Y is the frequency of a distinct path appearing in a document.

3.2 Content Similarity Measure

The semantic association among the document contents is measured using a latent semantic kernel (Cristianini et al. 2002). Consider the example documents in figure 2, a set of distinct terms $\{t_1, t_2, \dots, t_m\}$, denoted by T , is extracted from the dataset D . A term-document matrix, $X_{m \times n}$, where m is the number of terms in T and n is the number of documents in dataset D , is constructed as shown in table 2.

The singular value decomposition (SVD) decomposes the term-document matrix, $X_{m \times n}$, into three matrices (equation 3), where U and V have orthonormal columns values of left and right singular vectors

Table 2: A matrix X representing the content information of the dataset.

term/doc	d_1	d_2	d_3	d_4
t_1	2	1	2	2
t_2	2	2	2	0
t_3	2	2	2	0
t_4	2	2	1	4
t_5	2	0	2	0
t_6	1	0	0	0
t_7	2	2	2	1
t_8	0	1	0	1
t_9	1	1	1	0

respectively and S is a diagonal matrix of singular values ordered in decreasing magnitude.

$$X = USV^T. \quad (3)$$

SVD can optimally approximate matrix X with a smaller sample of matrices by selecting k largest singular values and setting the rest of the values to zero. Matrix U_k of size $m \times k$ and matrix V_k of size $n \times k$ may be redefined along with $k \times k$ singular value matrix S_k (equation 4). This can approximate the matrix X in a k -dimensional document space.

$$\hat{X}_{m \times n} = U_k S_k V_k^T. \quad (4)$$

Matrix \hat{X} is known to be the matrix of rank k which is closest in the least squares sense to X . Matrix U_k becomes the latent semantic kernel that can be used to measure the semantic associations between two document contents.

Definition 4 (Terms). A term, t_i , is a keyword that appears in the textual content of the elements in the XML document after stop-word removal and stemming (Porter 1980).

Definition 5 (Content Modeling). The content of a document, d_i , is modelled as a vector $\{t_{i,1}, t_{i,2}, \dots, t_{i,m}\}$, where each element of the vector represents the frequency of a term in T that appears in the document.

Definition 6 (Content Matching). Given two vectors, d_x and d_y , the semantic similarity of the documents content is measured as:

$$\text{contSim}(d_x, d_y) = \frac{d_x^T P P^T d_y}{|P^T d_x| |P^T d_y|}. \quad (5)$$

where matrix P is matrix U_k , and P is used as a mapping function to transform the two documents, d_x and d_y , into concept space to determine the semantic association of document contents.

4 Empirical Evaluation

4.1 Dataset

The IEEE and Wikipedia datasets, available from the INEX 2006 Document Mining Challenge (Denoyer et al. 2006), are used to evaluate the proposed clustering approach. The clusters are labelled according to the content theme or topic which makes the content similarity measure more important than the structure similarity measure.

The IEEE dataset is derived from the same structural definition therefore all documents contain the same set of element names. Likewise, the Wikipedia dataset is not conformed to any particular structural

Table 3: Datasets.

Datasets	#Documents	#True Categories
Wikipedia	3000	60
IEEE	6054	18
Heterogeneous dataset	3900	78

definition but documents also contain the same set of element names amongst the dataset. As a result no semantic learning is necessary on the element names. Table 3 shows the detail of the datasets. A subset of the Wikipedia dataset is used in the experiments. Wikipedia and IEEE datasets are homogeneous dataset, meaning, the documents in the dataset are conformed to only one structural definition; whereas, the heterogeneous dataset is a mixture of both the Wikipedia and IEEE documents where they are conformed to two different structural definitions.

4.2 Evaluation Methods

Two evaluation methods are used to measure the accuracy of the clustering solution; micro-average F1 and macro-average F1. Given a particular category, consider the number of positive documents which are clustered as positive (PP), the number of false negative documents which are clustered as positive (NP), and the number of false positive documents which are clustered as negative (PN), precision and recall are defined as follows:

$$\text{Precision}(P) = \frac{PP}{PP + NP}. \quad (6)$$

$$\text{Recall}(R) = \frac{PP}{PP + PN}. \quad (7)$$

The F1 measure for this particular category can be defined as:

$$F1 = \frac{2PR}{P + R}. \quad (8)$$

Micro-average F1 is calculated by summing up the PP , the NP , and the PN values from all the categories; F1 value is then calculated based on these values. Macro-average F1, on the other hand, is derived from averaging the F1 values over all the categories. The best clustering solution for an input data set is the one where micro- and macro-average F1 measures are close to 1. The Micro-average F1 value is easier to achieve than the macro-average F1 value.

4.3 Experimental Design

In the experiments, a subset, ranging from 1000 to 1300 documents, of each dataset is used for the construction of the latent semantic kernels. Only a subset is used because applying the singular vector decomposition method (SVD) on a large term-document matrix is expensive in terms of computational time and memory requirements, and sometimes infeasible. During the selection of the subset, it is ensured that the kernel is build on a large number of terms that appear in the dataset. Documents that contain large number of frequent terms are selected for the kernel construction. In the experiments, the clustering solution is analysed using different k values for selecting the kernels. Results, as shown in table 4 on the heterogeneous dataset, show that the k dimension of 200 and 400 is good to infer semantic association among the dataset contents. These values have been used

Table 4: The effect of k values on the clustering solution for the heterogeneous dataset.

k	Micro-average F1	Macro-average F1
100	0.299	0.240
200	0.346	0.290
400	0.308	0.247
600	0.283	0.222
800	0.280	0.223

for the evaluation and approaches comparison in this paper. Three different kernels are created for three different datasets as shown in table 3. A hierarchical clustering method (Karypis 2007) is used to cluster the pair-wise document similarity matrix produced from our clustering approach. The hierarchical clustering method performs by first dividing the dataset, in this case the pair-wise document similarity matrix, into two groups, and then one of these two groups is chosen to be bisected further. The process is repeated until the number of bisections in the process equals to the number of clusters defined by the user.

Experiments are conducted to evaluate the effect of the structure similarity and the content similarity on the clustering solutions. Results of the proposed clustering approach on the IEEE dataset are compared with two other approaches (Doucet & Lehtonen 2006, Kc et al. 2006). The first one is the Doucet et al. (2006) approach which uses the vector space model for representing the XML document features, and then k-means to cluster the documents. The other one is the Kc et al. (2006) approach which uses the self-organization maps to combine the structure and content information for document clustering.

4.4 Results and Analysis

The effect of the weighting parameter λ . The structure and content similarities are adjusted with the weighting parameter λ . Figures 3, 4, and 5 show the effect of the weighting importance of the content similarity and structure similarity on the Wikipedia, IEEE and the heterogeneous datasets, respectively. Each figure shows the performance of micro-average and macro-average F1 values with various combinations of weighting parameters that is monitored by λ in equation 1. The graphs in the figures start with λ set to 0, where the importance of the content similarity is set to 0 and the importance of the structural similarity is set to a 1. The F1 values are then recorded each time with an increment of 0.1 in λ , decreasing the structural weight parameters by 0.1 and increasing the content weight parameters by 0.1. In general, the F1 measures become better with each increment in the content weight parameter. When the content weight is set to a higher value, the results are better in comparison to the results when the content weight is set to a lower value. This shows that the content information on these datasets plays an important role on the performance of the clustering solution. This is the expected results as documents are categorized according to the content that they share. Based on the results in figures 3 and 4, it can be ascertained that the structure of the data does not play much importance in the clustering of the datasets in homogeneous environment. However, in heterogeneous environment, results, as shown in figure 5, show that when the structure weight is assigned with a 0.1 or 0.2, the results are slightly better than the result with the content-only information. This emphasizes that by combining the structure and content measures with different strengths produces a better clustering solution for text-centric XML documents from het-

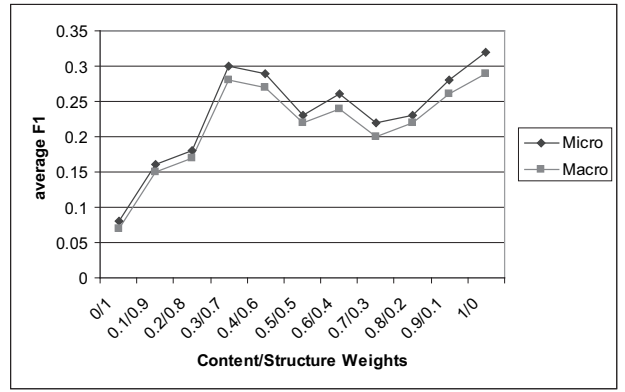


Figure 3: The effect of the structure and content similarities on the clustering solution of the Wikipedia dataset.

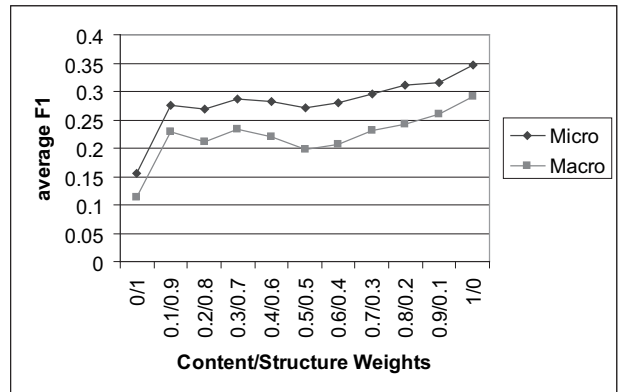


Figure 4: The effect of the structure and content similarities on the clustering solution of the IEEE dataset.

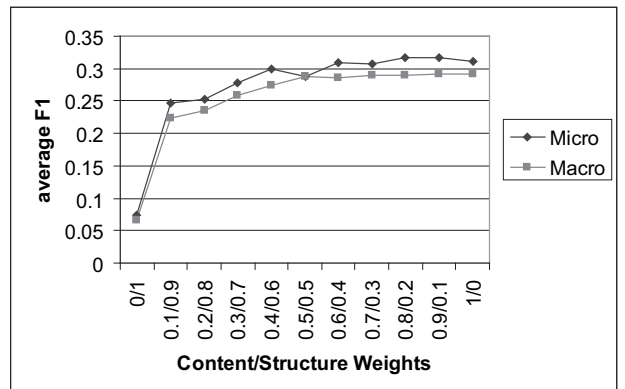


Figure 5: The effect of the structure and content similarities on the clustering solution of the heterogeneous dataset.

In this paper, we employ the structure similarity based on the frequency of the paths represented in the vector space model. We have also employed other representations and measures to exploit the structure information of XML documents in clustering as shown in table 5. The path vector space model approach is the one which has been used in this paper. The path-based approach (Nayak & Tran 2007) measures the structure similarity between documents using the path representation. The paths between documents are measured by considering

Table 5: The structure-only clustering solutions for Wikipedia dataset.

Approach	Micro-average F1	Macro-average F1
Path Vector Space Model	0.08	0.07
Path-based (Nayak & Tran 2007)	0.12	0.04
Tree-based (Kutty et al. 2007)	0.10	0.02

the hierarchical order of the elements in the paths. Whereas, the tree-based approach (Kutty et al. 2007) is to measure the structure similarity based on tree representation where the sibling information of the elements is also exploited for document similarity. All three approaches give very close results, as given in table 5, showing that the structure similarity on the Wikipedia dataset does not improve much with different representations. This shows that the proposed way of determining the structural similarity, in this paper, is sufficient enough for the clustering of the text-centric datasets. The path vector space model is chosen to be used in this proposed approach because it is faster to compute than the other two representation approaches.

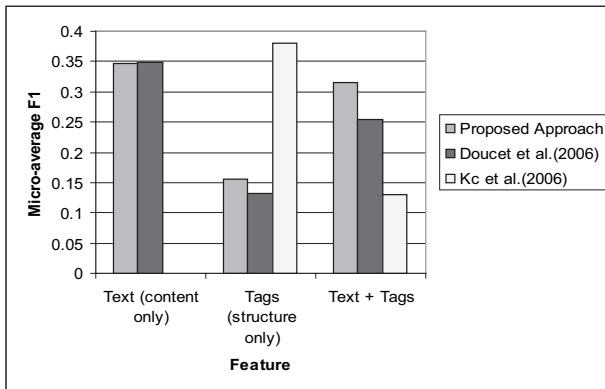


Figure 6: The micro-average F1 of the proposed approach, Doucet et al.(2006), and Kc et al.(2006)

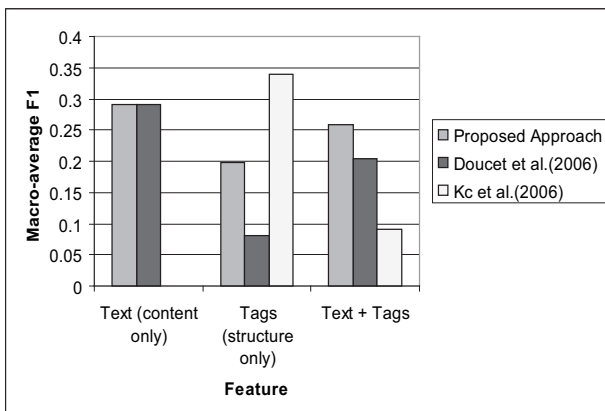


Figure 7: The macro-average F1 of the proposed approach, Doucet et al.(2006), and Kc et al.(2006)

Approaches comparison on the IEEE dataset. Figures 6 and 7 show the comparison of the proposed approach with the other approaches on micro-average F1 and macro-average F1 results, respectively. For

the content-only information (content similarity), our approach and Doucet et al. (2006) produce similar results. Even though, our kernel is built on a subset of document features in the dataset however the performance of the kernel is not worse than the vector space models based on the whole dataset features. Kc et al. (2006) uses the self-organization maps that outperforms both the vector space model based methods, Doucet et al. method (2006) and our approach, for using the structure-only information (structure similarity). However, when the structure and the content information are used, Kc et al.(2006) method performs the worse.

In summary, the self-organization maps method (Kc et al. 2006) is much better than the vector space model approach employed in Doucet et al. (2006) and co-occurrence counting of paths used in our approach for learning the structure of XML documents. On the other hand, the content of XML documents are better represented and grouped if it is represented as a vector space model or using the latent semantic kernel in our approach. When both the structure and content information of the XML documents are used for clustering, The proposed clustering approach outperforms the other two approaches because it can adjust the weighting importance on the content similarity and the structure similarity depending on the nature of the dataset.

5 Conclusions

This paper introduces a clustering approach based on two separate measures to explore the structure and content similarities of XML documents. In this paper we propose to adapt the latent semantic kernel to learn the semantic associations of XML document contents for content similarity. The result of the content similarity are combined with the structure similarity of the documents by assigning the two similarity measures with different weightings. This paper produces a systematic study of the effect of the structure and content similarities of the XML documents in the clustering process that has not been done previously in our knowledge. The method is thoroughly analysed and compared with other methods.

Empirical analysis ascertains the following information. In heterogeneous environment, the inclusion of structural similarity with the content similarity can produce a better result. The performance of the proposed approach is better when the dataset is in a heterogeneous environment rather than in a homogeneous environment. This is due to combining both the structure and content similarities using different measures and different weights. This shows the applicability of the proposed approach as this is usually the case in real practice. While grouping the data sets based on theme categories such as the Wikipedia and IEEE datasets, the clustering performance degrades when the structure of the documents is included in the clustering process. The content of the Wikipedia and IEEE datasets plays a major role in determining the clustering solutions, whereas the structure plays a small role.

The structure mining employed by this paper is a trivial method of measuring the structure of XML documents in a heterogeneous environment since hierarchical structural information of the document structure is not fully captured. However, the previous work has shown that the order of nodes is not important in clustering of text-centric XML datasets. Also the focus of this paper is to include the content and structure similarities in the clustering process. The experiments ascertained that the structure similarity and

the content similarity can contribute to the overall clustering solution when documents belong to different structural definitions.

References

- Bray, T., Paoli, J., Sperberg-McQueen, C., Maler, E. & Yergeau, F. (2004), 'Extensible markup language (xml) 1.0 (third edition) w3c recommendation'.
URL: <http://www.w3.org/TR/2004/REC-XML-20040204/>
- Cristianini, N., Shawe-Taylor, J. & Lodhi, H. (2002), 'Latent semantic kernels', *Journal of Intelligent Information Systems (JJIS)* **18**(2).
- Dalamagas, T., Cheng, T., Winkel, K. & Sellis, T. K. (2004), Clustering xml documents by structure, in 'SETN'.
- Denoyer, L., Gallinari, P. & Vercoastre, A.-M. (2006), Report on the xml mining track at inx 2005 and inx 2006, in 'INEX 2006', Dagstuhl Castle, Germany, pp. 432–443.
- Doucet, A. & Lehtonen, M. (2006), Unsupervised classification of text-centric xml document collections, in '5th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX', pp. 497–509.
- Han, J. & Kamber, M. (2001), *Data Mining: Concepts and Techniques*, San Diego, USA: Morgan Kaufmann.
- Jeong, H. H. & Keun, H. r. (2004), A new xml clustering for structural retrieval, in '23rd International Conference on Conceptual Modeling', Shanghai, China.
- Jeong, H. H. & Keun, H. R. (2005), Clustering and retrieval of xml documents by structure, in 'ICCSA', Singapore.
- Karypis, G. (2007), 'Cluto - software for clustering high-dimensional datasets — karypis lab'.
URL: <http://glaros.dtc.umn.edu/gkhome/views/cluto>
- Kc, M., Hagenbuchner, M., Tsoi, A., Scarselli, F., Sperduti, A. & Gori, M. (2006), Xml document mining using contextual self-organizing maps for structures, in '5th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX', Dagstuhl Castle, Germany, pp. 510–509.
- Kim, Y.-S., Cho, W.-J., Lee, J.-Y., Oh & Yu-Jin (2005), An intelligent grading system using heterogeneous linguistic resources, in 'IDEAL 2005', p. 102108.
- Kohonen, T. (1990), 'Self-organisation and associative memory', *Springer, 3rd edition*.
- Kurgan, L., Swiercz, W. & Cios, K. J. (2002), Semantic mapping of xml tags using inductive machine learning, in '11th International Conference on Information and Knowledge Management', Virginia, USA.
- Kutty, S., Tran, T., Nayak, R. & Li, Y. (2007), Clustering xml documents using closed frequency subtrees - a structure-only based approach, in '6th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2007', Dagstuhl Castle, Germany.
- Landauer, T. K., Foltz, P. W. & Laham, D. (1998), 'An introduction to latent semantic analysis', *Discourse Processes* (25), 259–284.
- Lee, L. M., Yang, L. H., Hsu, W. & Yang, X. (2002), Xclust: Clustering xml schemas for effective integration, in '11th ACM International Conference on Information and Knowledge Management (CIKM'02)', Virginia.
- Leung, H.-p., Chung, F.-l., Chan, S. & Luk, R. (2005), Xml document clustering using common xpath, in 'International Workshop on Challenges in Web Information Retrieval and Integration (WIRI '05)', pp. 91–96. TY - CONF.
- Lian, W., Cheung, D. W., Maoulis, N. & Yiu, S.-M. (2004), 'An efficient and scalable algorithm for clustering xml documents by structure', *IEEE TKDE* **16**(1), 82–96.
- Liu, J., Wang, J., Hsu, W. & Herbert, K. (2004), 'Xml clustering by principal component analysis', *Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on* pp. 658–662.
- Nayak, R. & Tran, T. (2007), 'A progressive clustering algorithm to group the xml data by structural and semantic similarity', *IJPRAI* **21**(3), 1–23.
- Nayak, R., Witt, R. & Tonev, A. (2002), Data mining and xml documents, in 'The 2002 International Workshop on the Web and Database (WebDB 2002)'.
- Nierman, A. & Jagadish, H. V. (2002), Evaluating structural similarity in xml documents, in '5th International Conference on Computational Science (ICCS'05)', Wisconsin, USA.
- Porter, M. (1980), 'An algorithm for suffix stripping', *Program* **14**(3), 130–137.
- Salton, G. & McGill, M. J. (1983), 'Introduction to modern information retrieval', *McGraw-Hill*.
- Shen, Y. & Wang, B. (2003), Clustering schemaless xml document, in '11th international conference on Cooperative Information System'.
- Yang, J., Cheung, W. & Chen, X. (2005), Learning the kernel matrix for xml document clustering, in 'e-Technology, e-Commerce and e-Service'.

Author Index

- Adams, Brett, 187
 Ahmed, Chowdhury Farhan, 79
 Al-Oqaily, Ahmad, 111
 Allwright, Alan, 41
 Altman, Tom, 209
 AlZoubi, Omar, 123
 Arbelaitz, Olatz, 163, 171
- Bringay, Sandra, 95
 Bruza, Peter, 219
- Calvo, Rafael A., 123
 Catchpoole, Daniel R., 111, 133
 Choi, Jong Pill, 141
 Chong, Wen Haw, 21
 Christen, Peter, iii, 51
 Cios, Krzysztof J., 209
- Dash, Manoranjan, 179
 Dattasharma, Abhi, 153
- Edwards, Brett, 193
- G, Sridhar, 153
 Gayler, Ross, 51
 Ghous, Hamid, 133
 Gurrutxaga, Ibai, 163, 171
- Hilderman, Robert, 201
 Honnappa, Harsha, 61
- Jeacocke, David, 105
 Jeong, Byeong-Soo, 79
- Kennedy, Paul J., iii, 111, 133
 Khan, Umer, 141
 Kim, Minkoo, 141
 Koh, Yun Sing, 87
 Koprinska, Irena, 123
- Laurent, Anne, 95
 Lee, Young-Koo, 79
 Li, Jiuyong, iii, 73
- M^aPérez, Jesús, 163, 171
 Martín, José I., 163, 171
 Muguerza, Javier, 163, 171
 Murray, D. Wayne, 105
- Nayak, Richi, 193, 219
- Pears, Russel, 87
 Perona, Iñigo, 163, 171
 Phung, Dinh, 187
 Poon, Josiah, 27
 Powers, David M.W., 3
 Price, Richard, 17
- Roddick, John F., iii, 41
- Saneifar, Hassan, 95
 Shalom, S.A. Arul, 179
 Shan, Yin, 105
 Shin, Hyunjung, 141
 Simeon, Mondelle, 201
 Simoff, Simeon J., 111, 133
 Sun, Xiaoxun, 73
 Sutinen, Alison, 105
- Tanbeer, Syed Khairuzzaman, 79
 Teisseire, Maguelonne, 95
 Thaicharoen, Supphachai, 209
 Tran, Kha, 187
 Tran, Tien, 219
 Tripathi, Praveen Kumar, 153
 Tue, Minh, 179
- Venkatesh, Svetha, 187
- Wang, Hua, 73
 Webb, Geoff, 15
 Weng, Cheng G., 27
- Xie, Zhipeng, 33
- Zatorsky, Michael, 193

Recent Volumes in the CRPIT Series

ISSN 1445-1336

Listed below are some of the latest volumes published in the ACS Series *Conferences in Research and Practice in Information Technology*. The full text of most papers (in either PDF or Postscript format) is available at the series website <http://crpit.com>.

Volume 67 - Conceptual Modelling 2007

Edited by John F. Roddick, *Flinders University* and Annika Hinze, *University of Waikato, New Zealand*. January, 2007. 978-1-920682-48-4.

Contains the proceedings of the Fourth Asia-Pacific Conference on Conceptual Modelling (APCCM2007), Ballarat, Victoria, Australia, January 2007.

Volume 68 - ACSW Frontiers 2007

Edited by Ljiljana Brankovic, *University of Newcastle*, Paul Coddington, *University of Adelaide*, John F. Roddick, *Flinders University*, Chris Steketee, *University of South Australia*, Jim Warren, *the University of Auckland*, and Andrew Wendelborn, *University of Adelaide*. January, 2007. 978-1-920682-49-1.

Contains the proceedings of the ACSW Workshops - The Australasian Information Security Workshop: Privacy Enhancing Systems (AISW), the Australasian Symposium on Grid Computing and Research (AUSGRID), and the Australasian Workshop on Health Knowledge Management and Discovery (HKMD), Ballarat, Victoria, Australia, January 2007.

Volume 69 - Safety Critical Systems and Software 2006

Edited by Tony Cant, *Defence Science and Technology Organisation, Australia*. February, 2007. 978-1-920682-50-7.

Contains the proceedings of the 11th Australian Conference on Safety Critical Systems and Software, August 2006, Melbourne, Australia.

Volume 70 - Data Mining and Analytics 2007

Edited by Peter Christen, Paul Kennedy, Jiuyong Li, Inna Kolyshkina and Graham Williams. December, 2007. 978-1-920682-51-4.

Contains the proceedings of the 6th Australasian Data Mining Conference (AusDM 2007), Gold Coast, Australia. December 2007.

Volume 72 - Advances in Ontologies 2006

Edited by Mehmet Orgun, *Macquarie University* and Thomas Meyer, *National ICT Australia, Sydney*. December, 2006. 978-1-920682-53-8.

Contains the proceedings of the Australasian Ontology Workshop (AOW 2006), Hobart, Australia, December 2006.

Volume 73 - Intelligent Systems for Bioinformatics 2006

Edited by Mikael Boden and Timothy Bailey, *University of Queensland*. December, 2006. 978-1-920682-54-5.

Contains the proceedings of the AI 2006 Workshop on Intelligent Systems for Bioinformatics (WISB-2006), Hobart, Australia, December 2006.

Volume 74 - Computer Science 2008

Edited by Gillian Dobbie, *University of Auckland, New Zealand* and Bernard Mans, *Macquarie University*. January, 2008. 978-1-920682-55-2.

Contains the proceedings of the Thirty-First Australasian Computer Science Conference (ACSC2008), Wollongong, NSW, Australia, January 2008.

Volume 75 - Database Technologies 2008

Edited by Alan Fekete, *University of Sydney* and Xuemin Lin, *University of New South Wales*. January, 2008. 978-1-920682-56-9.

Contains the proceedings of the Nineteenth Australasian Database Conference (ADC2008), Wollongong, NSW, Australia, January 2008.

Volume 76 - User Interfaces 2008

Edited by Beryl Plimmer and Gerald Weber, *University of Auckland*. January, 2008. 978-1-920682-57-6.

Contains the proceedings of the Ninth Australasian User Interface Conference (AUI2008), Wollongong, NSW, Australia, January 2008.

Volume 77 - Theory of Computing 2008

Edited by James Harland, *RMIT University* and Prabhu Manyem, *University of Ballarat*. January, 2008. 978-1-920682-58-3.

Contains the proceedings of the Fourteenth Computing: The Australasian Theory Symposium (CATS2008), Wollongong, NSW, Australia, January 2008.

Volume 78 - Computing Education 2008

Edited by Simon, *University of Newcastle* and Margaret Hamilton, *RMIT University*. January, 2008. 978-1-920682-59-0.

Contains the proceedings of the Tenth Australasian Computing Education Conference (ACE2008), Wollongong, NSW, Australia, January 2008.

Volume 79 - Conceptual Modelling 2008

Edited by Annika Hinze, *University of Waikato, New Zealand* and Markus Kirchberg, *Massey University, New Zealand*. January, 2008. 978-1-920682-60-6.

Contains the proceedings of the Fifth Asia-Pacific Conference on Conceptual Modelling (APCCM2008), Wollongong, NSW, Australia, January 2008.

Volume 80 - Health Data and Knowledge Management 2008

Edited by James R. Warren, Ping Yu, John Yearwood and Jon D. Patrick. January, 2008. 978-1-920682-61-3.

Contains the proceedings of the Australasian Workshop on Health Data and Knowledge Management (HDKM 2008), Wollongong, NSW, Australia, January 2008.

Volume 81 - Information Security 2008

Edited by Ljiljana Brankovic, *University of Newcastle* and Mirka Miller, *University of Ballarat*. January, 2008. 978-1-920682-62-0.

Contains the proceedings of the Australasian Information Security Conference (AISC 2008), Wollongong, NSW, Australia, January 2008.

Volume 82 - Grid Computing and e-Research

Edited by Wayne Kelly and Paul Roe, *QUT*. January, 2008. 978-1-920682-63-7.

Contains the proceedings of the Australasian Workshop on Grid Computing and e-Research (AusGrid 2008), Wollongong, NSW, Australia, January 2008.

Volume 83 - Challenges in Conceptual Modelling

Edited by John Grundy, *University of Auckland, New Zealand*, Sven Hartmann, *Massey University, New Zealand*, Alberto H.F. Laender, *UFMG, Brazil*, Leszek Maciaszek, *Macquarie University, Australia* and John F. Roddick, *Flinders University, Australia*. December, 2007. 978-1-920682-64-4.

Contains the tutorials, posters, panels and industrial contributions to the 26th International Conference on Conceptual Modeling - ER 2007.

Volume 84 - Artificial Intelligence and Data Mining 2007

Edited by Kok-Leong Ong, *Deakin University, Australia*, Wenyan Li, *University of Texas at Dallas, USA* and Junbin Gao, *Charles Sturt University, Australia*. December, 2007. 978-1-920682-65-1.

Contains the proceedings of the 2nd International Workshop on Integrating AI and Data Mining (AIDM 2007), Gold Coast, Australia. December 2007.

Volume 86 - Safety Critical Systems and Software 2007

Edited by Tony Cant, *Defence Science and Technology Organisation, Australia*. December, 2007. 978-1-920682-67-5.

Contains the proceedings of the 12th Australian Conference on Safety Critical Systems and Software, August 2006, Adelaide, Australia.