Conferences in Research and Practice in
Information Technology

Volume 70

Data Mining and Analytics 2007

AUSTRALIAN
COMPUTER
SOCIETY

Data Mining and Analytics 2007

Proceedings of the
Sixth Australasian Data Mining Conference (AusDM'07),
Gold Coast, Australia, 3-4 December, 2007

Peter Christen, Paul J. Kennedy, Jiuyong Li,
Inna Kolyshkina and Graham J. Williams, Eds.

Volume 70 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 2007. Proceedings of the Sixth Australasian Data Mining Conference (AusDM'07), Gold Coast, Australia, 3-4 December, 2007

Conferences in Research and Practice in Information Technology, Volume 70.

Editors: **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 Information Technology
University of Technology, Sydney
Broadway, NSW, 2007, Australia
Email: paulk@it.uts.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

Inna Kolyshkina
Westpac Banking Corporation
Level 6, 60 Martin Place
Sydney NSW 2000, Australia
Email: ikolyshkina@westpac.com.au

Graham J. Williams
Australian Taxation Office
Narellan Street
Canberra ACT 2601, Australia
Email: Graham.Williams@togaware.com

Series Editors:
Vladimir Estivill-Castro, Griffith University, Queensland
John F. Roddick, Flinders University, South Australia
Simeon Simoff, University of Technology, Sydney, NSW
`crpit@infoeng.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 70.
ISSN 1445-1336.
ISBN 978-1-920682-51-4.

Printed, October 2007 by Griffith University Printing Services, Brisbane, QLD.
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

Association Rule and Frequent Pattern Mining

Financial and Policing/Security Data Mining

Algorithms

Data Mining Education

Preface

The Australasian Data Mining Conference series **AusDM**, started in 2002, is 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) and Cairns (2004) to conferences in Sydney (2005, 2006). This year we are delighted to be co-hosted with the Twentieth Australian Joint Conference on Artificial Intelligence on the Gold Coast, Queensland, and the Second International Workshop on Integrating AI and Data Mining. 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;
- Griffith University 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 e-Markets Research Group, for providing essential expertise for the event;
- the Australian Computer Society, for publishing the conference proceedings;
- StatSoft for their support;
- data mining postgraduate students from Queensland University of Technology for their local support.

This year the Steering Committee and IAPA have recognised the importance of education in data mining and we have included a special panel session devoted to Data Mining Education. Also this year, for the first time, we have presented a Best Paper Award (voted by the peer review) and a Best Presentation Award (voted by conference delegates). We are delighted to expand the social program this year and hope that conference attendees will enjoy this extra time to make new contacts and to trade "war stories".

The conference program committee reviewed 69 submissions. This was an almost 20% increase in the number of submissions from last year. From these submissions 26 were selected for publication and presentation. This was an acceptance rate of 38%. **AusDM** follows a rigid double 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 note that the cut-off threshold has been high (5 on a 7 point scale). This is testament to the high quality of submissions. 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. We are proud to include in these proceedings the papers from the Second International Workshop on Integrating AI and Data Mining. Papers published in both volumes by the Australian Computer Society are indexed and available for download.

Data mining and analytics today have advanced rapidly from the early days of pattern finding in commercial databases. They are now a core part of business intelligence and inform decision making in many areas of human endeavour including science, business, health care and security. Mining of unstructured text, semi-structured web information and multimedia data have continued to receive attention, as have professional challenges to using data mining in industry. Accepted submissions have been grouped into seven sessions reflecting these application areas. Three invited industry keynote sessions put the research into context.

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

Peter Christen
Paul J. Kennedy
Jiuyong Li
Richi Nayak
Simeon J. Simoff
Graham J. Williams

AusDM'07 General Chairs and Conference Chairs
October 2007

Organising Committee

Conference Chairs

Simeon J Simoff, University of Western Sydney / University of Technology, Sydney
Graham J Williams, Australian Taxation Office, Canberra

General Chairs

Peter Christen, Australian National University, Canberra
Paul Kennedy, University of Technology, Sydney
Jiuyong Li, University of South Australia, Adelaide
Richi Nayak, Queensland University of Technology, Brisbane

Organising Chair

Vladimir Estivill-Castro, Griffith University, Queensland

Publicity Chair

Longbing Cao, University of Technology, Sydney

Industry Chairs

Warwick Graco, Australian Taxation Office, Canberra
Eugene Dubossarsky, Ernst-Young, Sydney
John Galloway, NetMap Analytics, Sydney
Inna Kolyshkina, Westpac Banking Corporation, Sydney

Programme Committee

Rohan Baxter, Australian Taxation Office, Canberra, Australia
Helmut Berger, University of Technology, Sydney, Australia
Eibe Frank, University of Waikato, New Zealand
Mohamed Gaber, CSIRO ICT Centre, Tasmania, 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
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
Christine O'Keefe, CSIRO, Canberra, Australia
Kok-Leong Ong, Deakin University, Victoria, Australia
Mehmet Orgun, Macquarie University, Sydney, Australia
Tom Osborn, The Leading Edge, Sydney, Australia
Robert Pearson, Canberra, Australia
François Poulet, ESIEA, Laval, France
Ben Raymond, Australian Antarctic Division, Hobart, Australia
John Roddick, Flinders University, Adelaide, 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
Huaifeng Zhang, Centrelink, Canberra, Australia
Yanchang Zhao, University of Technology, Sydney, Australia

External Reviewers

Firas Wehbe
Aaron Ceglar
Wei Wang
Muhammad Cheema

AusDM Sponsors

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

http://www.togaware.com

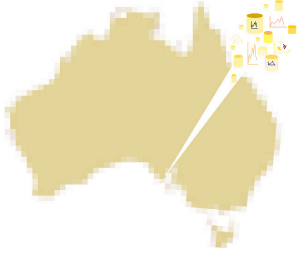
Faculty of Information Technology
http://www.uts.edu.au, http://www.it.uts.edu.au

The e-Markets Research Group
http://www.e-markets.org.au

http://www.iapa.org.au

http://www.netmapanalytics.com

http://www.statsoft.com



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

Conference Programme

Monday, 3 December, 2007

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

09:05 - 10:05 **INDUSTRY KEYNOTE**.

10:05 - 10:30 Coffee break

10:30 - 12:30 **Session 1: Industry Data Mining**

- 10:30 - 11:00 ANALYTICS FOR AUDIT AND BUSINESS CONTROLS IN CORPORATE TRAVEL & ENTERTAINMENT,
  Vijay Iyengar, Ioana Boier, Karen Kelley, Raymond Curatolo
- 11:00 - 11:30 CUSTOMER ANALYTICS PROJECTS: ADDRESSING EXISTING PROBLEMS WITH A PROCESS THAT LEADS TO SUCCESS,
  Inna Kolyshkina, Simeon Simoff
- 11:30 - 12:00 PREDICTIVE MODEL OF INSOLVENCY RISK FOR AUSTRALIAN CORPORATIONS,
  Rohan Baxter, Mark Gawler, Russell Ang
- 12:00 - 12:30 ESTABLISHING A LINEAGE FOR MEDICAL KNOWLEDGE DISCOVERY,
  Anna Shillabeer, John Roddick

10:30 - 12:30 **Session 2: Text Mining**

- 10:30 - 11:00 MEASURING DATA-DRIVEN ONTOLOGY CHANGES USING TEXT MINING,
  Majigsuren Enkhsaikhan, Wilson Wong, Wei Liu, Mark Reynolds
- 11:00 - 11:30 DETERMINING TERMHOOD FOR LEARNING DOMAIN ONTOLOGIES USING DOMAIN PREVALENCE AND TENDENCY,
  Wilson Wong, Wei Liu, Mohammed Bennamoun
- 11:30 - 12:00 DETERMINING TERMHOOD FOR LEARNING DOMAIN ONTOLOGIES IN A PROBABILISTIC FRAMEWORK,
  Wilson Wong, Wei Liu, Mohammed Bennamoun
- 12:00 - 12:30 USING CORPUS ANALYSIS TO INFORM RESEARCH INTO OPINION DETECTION IN BLOGS,
  Deanna Osman, John Yearwood, Peter Vamplew

12:30 - 14:00 Lunch

14:00 - 14:30 **Session 3: Best Paper**

- 14:00 - 14:30 DISCOVERING FREQUENT SETS FROM DATA STREAMS WITH CPU CONSTRAINT,
  Xuan Hong Dang, Wee-Keong Ng, Kok-Leong Ong, Vincent C S Lee

14:30 - 15:00 **IAPA Discussion**

15:00 - 15:30 Coffee break

15:30 - 17:30 **Session 4: Unsupervised Learning**

- 15:30 - 16:00 EXPLORATORY MULTILEVEL HOT SPOT ANALYSIS: AUSTRALIAN TAXATION OFFICE CASE STUDY,
  Denny, Graham J. Williams, Peter Christen
- 16:00 - 16:30 THE APPLICATION OF DATA MINING TECHNIQUES TO CHARACTERIZE AGRICULTURAL SOIL PROFILES,
  Leisa Armstrong, Dean Diepeveen, Rowan Maddern
- 16:30 - 17:00 USEFUL CLUSTERING OUTCOMES FROM MEANINGFUL TIME SERIES CLUSTERING,
  Jason Chen
- 17:00 - 17:30 A TWO-STEP CLASSIFICATION APPROACH TO UNSUPERVISED RECORD LINKAGE,
  Peter Christen

15:30 - 17:00 **Session 5: Association Rules and Frequent Patterns Mining**

15:30 - 16:00 SEMGRAM - INTEGRATING SEMANTIC GRAPHS INTO ASSOCIATION RULE MINING,
John Roddick, Peter Fule

16:00 - 16:30 ARE ZERO-SUPPRESSED BINARY DECISION DIAGRAMS GOOD FOR MINING FREQUENT PATTERNS IN HIGH DIMENSIONAL DATASETS?,
Elsa Loekito, James Bailey

16:30 - 17:00 PCITMINER- PREFIX-BASED CLOSED INDUCED TREE MINER FOR FINDING CLOSED INDUCED FREQUENT SUBTREES,
Sangeetha Kutty, Richi Nayak, Yuefeng Li

Tuesday, 4 December, 2007

09:00 - 10:00 **INDUSTRY KEYNOTE**

10:05 - 10:30 Coffee break

10:30 - 12:30 **Session 6: Financial and Policing / Security Data Mining**

10:30 - 11:00 NEWS AWARE VOLATILITY FORECASTING: IS THE CONTENT OF NEWS IMPORTANT?,
Calum Robertson, Shlomo Geva, Rodney Wolff

11:00 - 11:30 EFFECTIVENESS OF USING QUANTIFIED INTERMARKET INFLUENCE FOR PREDICTING TRADING SIGNALS OF STOCK MARKETS,
Chandima Tilakaratne, Musa Mammadov, Sidney Morris

11:30 - 12:00 ADAPTIVE SPIKE DETECTION FOR RESILIENT DATA STREAM MINING,
Clifton Phua, Kate Smith-Miles, Vincent Lee, Ross Gayler

12:00 - 12:30 MINING FOR OFFENDER GROUP DETECTION AND STORY OF A POLICE OPERATION,
Fatih Ozgul, Julian Bondy, Hakan Aksoy

10:30 - 12:30 **Session 7: Algorithms**

10:30 - 11:00 PREFERENCE NETWORKS: PROBABILISTIC MODELS FOR RECOMMENDATION SYSTEMS,
Tran The Truyen, Dinh Quoc Phung, Svetha Venkatesh

11:00 - 11:30 CLASSIFICATION FOR ACCURACY AND INSIGHT: A WEIGHTED SUM APPROACH,
Anthony Quinn, Andrew Stranieri, John Yearwood

11:30 - 12:00 A NEW EFFICIENT PRIVACY-PRESERVING SCALAR PRODUCT PROTOCOL,
Artak Amirbekyan, Vladimir Estivill-Castro

12:00 - 12:30 AN E-MARKET FRAMEWORK TO DETERMINE THE STRENGTH OF BUSINESS RELATIONSHIPS BETWEEN INTELLIGENT AGENTS,
Khandaker Shahidul Islam

12:30 - 14:00 Lunch

14:00 - 15:00 **INDUSTRY KEYNOTE**.

15:00 - 15:30 Coffee break

15:30 - 16:55 **Session 8: Data Mining Education**

15:30 - 16:00 REFLECTION ON DEVELOPMENT AND DELIVERY OF A DATA MINING UNIT,
Bozena Stewart

16:00 - 16:30 EVALUATION OF A GRADUATE LEVEL DATA MINING COURSE WITH INDUSTRY PARTICIPANTS
Peter Christen

16:30 - 16:55 DATA MINING EDUCATION FORUM

16:55 - 17:00 **Conference Close and Presentation of Best Presentation Award**

Contributed Papers

Analytics for Audit and Business Controls in Corporate Travel & Entertainment

Vijay Iyengar, Ioana Boier
IBM Thomas J. Watson Research Center
19 Skyline Drive, Hawthorne, NY 10532, USA
(vsi, ioana)@us.ibm.com

Karen Kelley, Raymond Curatolo
IBM Global Technology Services
150 Kettletown Road, Southbury, CT 06488, USA
(langan, rac)@us.ibm.com

Abstract

Travel and Entertainment (T&E) expenses are under increasing scrutiny as one of the largest controllable indirect expenses in a firm. This involves internal audits and analysis by business controls personnel to identify fraud and misuse and to take appropriate corrective actions. We have developed a set of statistical models to identify suspicious behavior for further investigation. Our Behavioral Shift Models (BSM) leverage domain knowledge in the form of simple, generic templates that represent classes of fraud and abuse. The emphasis is on robustly detecting repeated, out-of-the-norm behaviors as opposed to single instance occurrences. In this paper, we describe the application of these models and characterize their detection capabilities empirically. We also present validated results and insights generated by our approach when applied to production data from multiple firms for several T&E scenarios.

Keywords: Audits, business controls, fraud and abuse.

1 Introduction

Travel and Entertainment (T&E) expenses are considered one of the largest controllable indirect expenses in a firm. The recent emphasis on business integrity and compliance in conjunction with a tight business environment and constant attention to the bottom line have led to a renewed focus on the implementation of effective management and controls for T&E. This entails multiple dimensions, including improvement of internal controls and related business processes, expense monitoring and timely auditing, and improved vendor procurement and management.

The problem we address in this paper is that of analyzing transaction data logged through a T&E system for the purpose of effective audit and business controls. The data consists of expense and approval records, but completely lacks historical information on the outcome of any subsequent actions. Unfortunately, it is rather common practice in the audit & business controls domain to process candidates deemed worthy of attention without documenting the results of the investigation within the original T&E environment. This poses an interesting set of challenges for the analysis of the data and considerably reduces the number of options among the techniques developed to date (see Section 2). The outcome of the analysis may be directed towards audit or business controls and may be relevant at different granularity levels (in what follows, the elements of each such level, e.g., individuals, organizational subunits such as accounting centers and divisions are referred to as *entities*).

The audit and business control functions serve different purposes. *Audit* refers to checks that are performed to ascertain the validity and reliability of the T&E information. For example, an audit could examine a specific subset of travel expenses claimed by an employee. Such an audit could uncover fraud, error in the claims process (e.g., incorrect expense type used to categorize a claim), or misuse (e.g., bypassing the expense approval process by inappropriately splitting transactions into ones of smaller, less conspicuous amounts). *Business controls* encompass activities that examine and analyze data from expense claims and expense approval processes for excessive violations of relevant corporate policies and guidelines. For example, excessive approval of violations of business class travel policy by an organization within a firm could trigger an investigation and potential action to improve compliance with business travel policy. Currently, the internal audit and business controller roles are being emphasized in many organizations due to an increased focus on corporate business integrity. A decision to audit is not simply viewed in terms of balancing the cost of the audit against the costs due to the abuse. An audit is frequently pursued if there is adequate evidence and sometimes the investigation exposes the "tip-of-the-iceberg" where the same entities are involved in violations in other domains beyond T&E.

Detection of candidates for auditing and/or business control actions is a critical and challenging task. A typical approach relies on the deep knowledge of domain experts (auditors and business controls personnel) to aim at specific scenarios that reveal potential mechanisms for fraud, errors and misuse of policy. Clearly, such an approach makes for a highly non-uniform process of identifying candidates depending on the method and expertise of the individual domain expert. In the context of a T&E software system, this approach entails capturing and updating all possible scenarios describing mechanisms for fraud and misuse as they are discovered. At the opposite end of the spectrum, an ambitious approach is to try to detect entities for further

investigation without explicit domain knowledge about fraud and abuse mechanisms. This approach is relatively new and has been less explored in the literature or in the commercial space.

In our work we have adopted a middle path, by developing a set of Behavioral Shift Models (BSM), i.e., statistical models that identify suspicious behavior while relying only partially on domain knowledge. The latter is used solely to define a set of simple, generic templates that represent classes of fraud and abuse that may be of interest. The parameters of our statistical models for any given template are learned from the data. We contend that our models provide a balance between the amount of detailed domain knowledge required and the robustness of the insights generated (e.g., few false positives).

In this paper, we present two models that cover two classes of scenarios. The first model is applicable to cases that involve positive real-valued variables (e.g., categorized expense amounts, time durations such as payment delays). As an example, consider tip expenses of individual employees incurred during business travel. These expenses tend to be paid in cash and may not require receipts for reimbursement. A typical analysis scenario would seek to detect those employees with significantly high tip claims. Section 3 describes our first model, its empirical characterization, and results from scenarios in this class (referred to as *Expense Amount Scenarios*) using production T&E data from multiple firms. The second model is applicable to scenarios involving count data (referred to as *Event Count Scenarios*) for events like business rule exceptions. For example, organizations typically have well-defined business rules regarding the class of air travel allowed for business trips. They also have a business process for approving exceptions to this rule. From a business controls perspective, it is important to monitor and assess whether an organizational unit is lax in its business controls by excessively approving this type of exception. Section 4 describes the details of our second model and the corresponding results. The remainder of the paper is dedicated to the background for this work (Section 2), discussion (Section 5) and our conclusions (Section 6).

2 Background

Our work touches upon a multitude of aspects, some generic and some domain-specific. Broad topics like outlier identification, statistical inference, and hypothesis testing are examples in the former category. Analysis of transaction data in T&E systems for purposes such as reporting, monitoring, and compliance are representative of the latter. In this section we attempt to narrow down this rich field starting from the problem we are trying to solve and its desired (if not required) outcomes as described in the previous section.

Outlier detection pertains to the detection of anomalous observations (outliers) in data sets. The abnormality is typically defined with respect to other samples within the same data set. A broad spectrum of techniques has been developed for different applications on a varied theoretical backdrop that includes Statistics, Machine Learning, Neural Networks, etc. A comprehensive overview of outlier detection is provided by Hodge and Austin (2004). Using the taxonomy proposed in that work as our reference, we note that methods that fall in the Type 1 (i.e., unsupervised clustering) or Type 2 (i.e., supervised classification) categories do not offer suitable solutions to our problem: an a priori proximity metric to be used for clustering would be difficult to conjecture and labeled data is unavailable. The closest to our approach are the Type 3 methods (i.e., semi-supervised detection) that model normality and use it to pinpoint abnormal cases.

There has been extensive research done on outlier detection methods to identify observations (or points) in n-dimensional space that deviate from other observations. For example, statistical and data mining methods for this task are compared by Williams et al (2002). Such methods do not address the problem of analyzing repeat behavior that is of interest in our domain. We are interested in identifying entities with outlying behavior and the data contains varying numbers of behavioral observations for each entity.

A comprehensive review of statistical fraud detection is provided by Bolton and Hand (2002). Their exposition on unsupervised methods is clearly relevant to the problem addressed in this paper. The notion of using a statistical profile of the normal behavior has been used in earlier works. The computer intrusion detection work by Denning (1987) is an example that uses this approach. One of the statistical models used by Denning for representing the normal profile consists of the summarization using the mean and standard deviation. Any single observation is tested and scored for deviation from this normal characterization. The more recent work on peer group analysis by Bolton and Hand (2001) incorporates a key refinement by using local models in the form of peer groups that define normal behavior for any entity being analyzed for deviant behavior. However, in both examples the normal profile is used to score the deviation of a single observation. As mentioned earlier, our problem requires analyzing multiple observations for each entity to determine entities with repeated and significant outlying behavior.

Formulations developed in the area of Scan Statistics (Kulldorff 1997, Glaz et al 2001, Huang et al 2007) are well-suited to the problem at hand. The approach is to use hypothesis testing (Lehmann 1986) using the Likelihood Ratio Test (LRT) to scan for clusters of abnormality that stand out within the entire space of data considered. The expanding body of work in this area includes development of models suited for various underlying distributions and applications to various domains. Our work could be viewed as an adaptation of this approach to the problem of identifying suspicious behavior for audit and business controls purposes.

Our resulting solution is novel in the T&E domain for at least two essential reasons: (a) it uses a robust scoring mechanism that considers the magnitude of abnormality without requiring specification of boundaries between normal and abnormal; (b) it emphasizes the repetition of abnormal behavior as an important metric in

characterizing outliers. In T&E both aspects are crucial. For example, expenditure limits are set through policies and exceptions to these trigger alarms. However, there is considerable room for fraud under these limits which may not always be caught through additional thresholds (see Section 3). Capturing repetitiveness is also of essence: it corresponds to the amount of evidence to justify an audit and the cost of the corresponding follow-up investigation and it may reveal integrity gaps that may point to other problems. The importance of gathering sufficient audit evidence has been highlighted in other financial areas by Beasley et al (2001).

The importance of financial controls and policy adherence in the T&E domain is emphasized by the National Business Travel Association (NBTA). NBTA is a leading forum in the business travel domain and a source for information about the domain, including commercial service and product providers in this space. Commercial packages typically provide reporting functions that summarize and sort the data based on domain knowledge of the important metrics in this space. However, to the best of our knowledge our model and method represents the first analysis in this domain that evaluates each entity based on the magnitude of deviation from normal and the repetitiveness of the behavior after appropriate normalization.

3 Expense Amount Scenarios

In this section we describe our method as it applies to scenarios that involve positive, real-valued variables. Consider the tip expenses scenario introduced in Section 1. A typical analysis scenario would seek to detect those employees with significantly high tip claims. There are two important aspects we consider: (a) repetitiveness – we are interested in candidate entities (individual employees in this case) that exhibit a profile / pattern of repeated excessive tipping (in contrast with methods that focus on finding isolated outlier tip amounts); and (b) significance: to properly quantify excessiveness we incorporate domain knowledge that helps us normalize the range of our variables. For tips, the amount must be normalized by the location where the tip expense was incurred. The template for this scenario would specify the expenses to be analyzed (tip expenses), the covariate structure for normalization (location where expense was incurred), and the target entities (employees).

3.1 The Model

To analyse expense amount scenarios, we apply hypothesis testing using the LRT formulation. We compare the distribution of values for a given entity ξ with that of the baseline B of values computed from all the entities:

(H0: null hypothesis) $\mathbf{E}[\xi] = \mathbf{E}[B]$

(H1: alternate hypothesis) $\mathbf{E}[\xi] > \mathbf{E}[B]$

where $\mathbf{E}[]$ denotes the expectation (mean) operator. As previously explained, the values considered in the LRT must be normalized by taking into consideration all the relevant factors (determined through domain knowledge).

Through empirical analysis of many expense amount scenarios across datasets from multiple firms, we found that the exponential model developed in a recent work by Huang, Kulldorff and Gregario (2007) on a spatial scan statistic for survival data provides an excellent characterization for the majority of T&E baselines after proper normalization. In addition, we observed that in practice it has good power for a broader class of distributions (e.g., Gamma) which is in line with the observations of Huang et al (2007).

For each scenario we specify the entity space (e.g., employees, department, divisions, and business units) being targeted by the analysis. We also specify the covariates according to domain knowledge. Referring back to our tip example, the entities are individual employees and the location where the tip expense was incurred is the only covariate. The target variable is the amount of tip expense claimed. Categorical covariates are handled directly by determining a normalization factor F for each combination of covariate values. Typically, this factor F is the mean (or max) value for that combination of covariate values over all the entities. Normalization of an individual value simply becomes the ratio of the raw value and F. For example, each tip expense can be normalized by dividing it by the mean tip value for the location where the expense was incurred. Consider an entity ξ with M normalized values which sum up to S. Let the total number of normalized values over all the entities be N and their sum be T. The test statistic $\mathbf{Y}(\xi)$ for the exponential model is given by:

$$\mathbf{Y}(\xi) = M \times \mathbf{log}\left(\frac{M}{S}\right) + (N-M) \times \mathbf{log}\left(\frac{N-M}{T-S}\right) - N \times \mathbf{log}\left(\frac{N}{T}\right).$$

Following the methodology used with most scan statistics (Kulldorff 1997, Huang et al 2007), a p-value is computed by performing a number Z of Monte Carlo experiments. In each experiment, the entity values are determined by sampling from the baseline and the maximum test statistic achieved by any entity is computed. We use sampling with replacement instead of the permutation approach used by Huang et al (2007) since in our domain a small number of extreme values are deleted from the baseline. The p-value for the entity ξ is computed using the formula $(L+1)/Z$, where L is the number of Monte Carlo experiments with a maximum test statistic exceeding $\mathbf{Y}(\xi)$. Entities with p-values that reject the null hypothesis at the prescribed α level are considered candidates for further investigation and ranked in decreasing order of their test statistic values Y.

3.2 Empirical Characterization

We will analyse the power of our model empirically for a range of Gamma distributions that are based on our characterization of production T&E data. Our experimental procedure for this empirical characterization is sketched in Figure 1. Specifically, we report on one set of experiments each of which simulates 1000 entities. Each entity has varying number of data items representing the expense claims submitted. The number of data items for an entity is modeled by a Gamma

distribution (Γ_1) with shape parameter 1.0 and scale parameter 16 (i.e., mean = 16). The data values are characterized by various two-parameter Gamma distributions (Γ_2). In each case, without loss of generality, we choose the value distributions to have a mean 1.0 (the test statistic is invariant under multiplicative scaling). The following values were considered for the Gamma shape parameter: {0.25, 0.5, 1.0, 2.0, 3.0, 4.0, 5.0, and 6.0}. In each experiment, a single target entity is chosen to have its values increased by a percentage δ that is varied in the range [10, 400]. Note that the test statistic **Y** is not sensitive to the distribution of the increase across individual values for the target entity since it is based only on the sum of all its values.

This empirical analysis provides a characterization for the ability of our method to detect a target entity with inflated values at a given α level for p-values. Similarly, we also determine the number of non-target entities that are detected at the given α level and we use the two resulting characterizations to quantify the performance of our model in terms of false negatives and false positives in this idealized experimental setup.

Input: N_ξ number of entities in the population

N_b number of baselines

N_{exper} number of simulation experiments for each baseline

Γ_1{shape, scale}for the distribution of the number of expense items across entities

Γ_2{shape, scale}for the distribution of expense amounts across entities the population (mean = 1)

Output: statistics regarding successful identification of engineered increases

Algorithm:

Generate N_ξ random numbers according to Γ_1 distribution; these represent the number of expense items for each entity

for each baseline 1.. N_b **do**

  generate data for this baseline according to Γ_2 distribution; this data represents the amount of each expense for every entity

  for each δ amount of percentage increase **do**

    for each experiment 1..N_{exper} **do**

      select an entity ξ^* to be engineered

      apply a δ% increase to each expense amount in ξ^*

      run BSM model and compute p-values and detection statistics

      record entities with p-values below chosen α level & detection statistics

    end for

  end for

end for

Figure 1. Experimental procedure to evaluate the detection capabilities of BSM.

For illustration, we use $\alpha = 0.01$. First, we consider the experiments done with the Gamma shape parameter of 1.0 for the value distribution. We use a classification model (Duda, Hart and Stork 2001) to discriminate the class of experimental cases in which the target entity was detected by our method from those in which it was not. Figure 2 shows both these classes with the x-axis representing the number of data elements (ϕ) in the target entity and the y-axis representing the total excess added to the target's values (λ). The points marked with a "+" are instances of the class where the target entity was detected at the given α level. A linear classifier was generated using a training set composed of 25% of the data using an SVM formulation (Christianini and Shawe-Taylor 2000). The accuracy on the test set (remaining 75% of the data) was 95% in this case indicating that this linear discriminator is a reasonable characterization of the target detection achieved by our method. The equation for the linear discriminant is $\lambda = 0.513 \times \varphi + 13.26$ and it characterizes the amount of excess that is detectable by this model. For example, a target entity with 40 data entries is detected only if, on average, its values are increased by 84% of the mean value. In the limit, as the number of values in the target entity increases, the excess has to be 51% of the mean for it to be detected. The relatively high value for the excess needed in this case is due in part to the skewed nature of the exponential distribution (i.e., Gamma shape parameter of 1.0). The relatively long tail for values even under "normal" circumstances results in the need to have sizeable excess before it is deemed significant. Our experience with production T&E data summarized in the next subsection shows that even with this conservative performance our model detects many interesting candidates for further investigation.

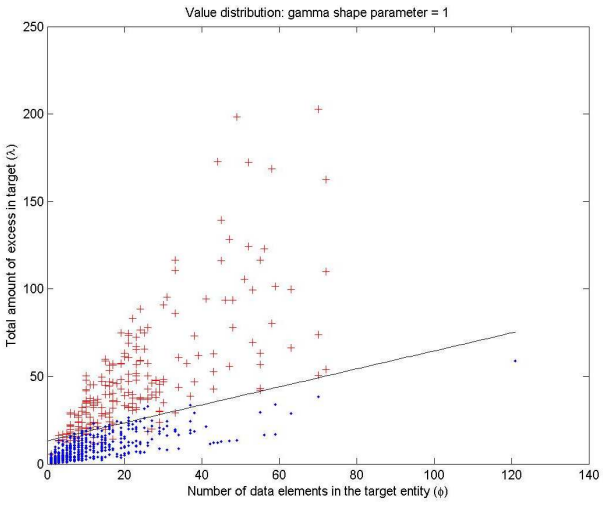


Figure 2. Detection of target entity for Gamma shape parameter of 1.0

Repeating the analysis by choosing other Gamma distributions for the baseline (i.e., choosing the Gamma shape parameter), we can characterize the detection ability of our model using the linear classifier learned for each Gamma distribution. The linear discriminants for the various Gamma distributions are shown in Figures 3 and 4. The SVM accuracy is higher than 93% in all the cases confirming the validity of these characterizations. The skewed value distributions when the shape parameter

is ≤ 1.0 result in larger excesses being required for detection (Figure 3). The tighter distributions that result as the Gamma shape parameter is increased lead to detection with much smaller excess (Figure 4). For example, when the shape parameter is 6.0, a target entity with 40 data entries is detected if, on average, its values are increased by as little as 25%.

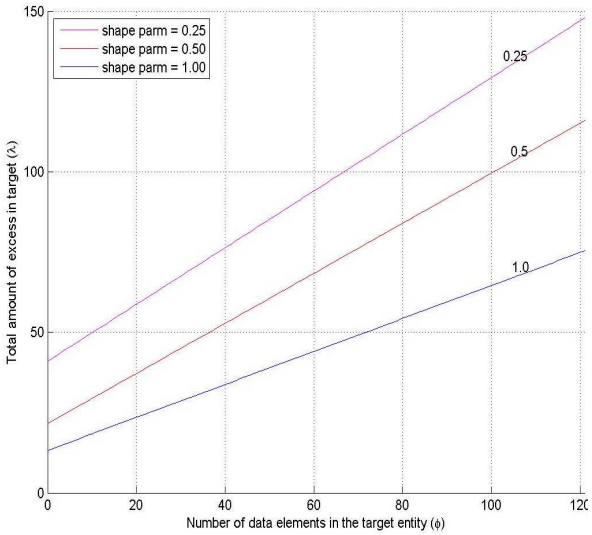


Figure 3. Linear classifier results for Gamma shape parameter values 0.25, 0.5, and 1.0.

The previous characterization indicates the ability of the model to detect the target for a wide range of Gamma distributions for the baseline values though the magnitude of excess needed is quite high when the baseline distribution itself has a long tail.

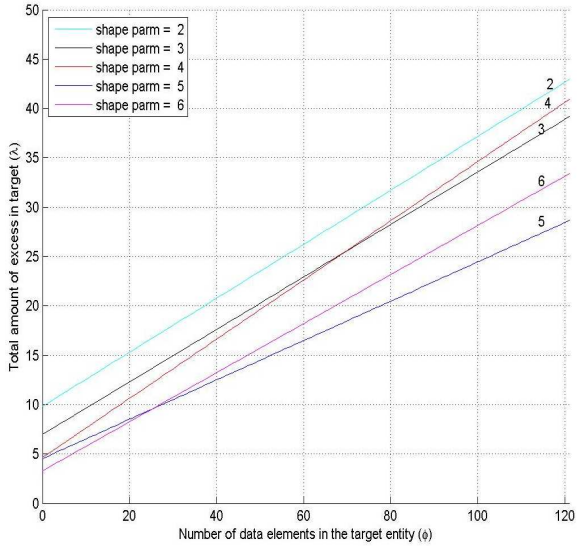


Figure 4. Linear classifier results for Gamma shape parameter values 2.0, 3.0, 4.0, 5.0, and 6.0.

Next, we consider the detection of non-target entities which gives us an indication of the false positive rate. At $\alpha = 0.01$, non-target entities were not detected in any of the experiments. The tradeoff between sensitivity and false positives can be illustrated if compare these results with those for $\alpha = 0.05$. The number of experiments (expressed as a percentage of the total number) in which non-target entities were detected is given for both α levels (0.01 and 0.05) in Table 1. At $\alpha = 0.05$ non-target entities are detected when the baseline Gamma distribution has shape parameter values of 2.0 and 3.0. In each instance, when a non-target entity was detected only one such entity was detected. The detection sensitivities for the two α levels can be compared by considering the corresponding equations for the linear discriminants. For the value Gamma shape parameter value of 1.0 considered earlier, the equation for the linear discriminant is $\lambda = 0.342 \times \varphi + 13.00$ at $\alpha = 0.05$ indicating the increase in sensitivity compared to the equation $\lambda = 0.513 \times \varphi + 13.26$ we had for $\alpha = 0.01$. The user can control this tradeoff between detection sensitivity and false positive rate by the choice of the α level.

The empirical characterization with the idealized Gamma distribution for the baselines indicates the magnitude of excess that is detectable by our method while keeping the false alarms rate in check. We show the utility of our method in the next subsection with results obtained by applying our method to production T&E data.

Gamma Shape Parameter	α -level =0.01		α -level =0.05	
	Target detected	Target not detected	Target detected	Target not detected
0.25, 0.5, 1.0	0	0	0	0
2.0	0	0	6.4%	6.7%
3.0	0	0	9.0%	10.3%
4.0, 5.0, 6.0	0	0	0	0

Table 1. Percentage of instances with non-target entity detection

3.3 Application to Production T&E Data

We applied our model to production T&E data from multiple firms in an enterprise expense reporting environment (GERS) and we reviewed the results of various scenarios with audit and business control professionals. The reviews were of a qualitative, not quantitative nature, i.e., they did not provide a quantitative assessment of false positive and false negative rates, but they did confirm the usefulness of our model. The top significant candidates detected by our technique in each scenario were found by the auditors to be interesting targets for further investigation. Interestingly, most of the candidates identified were not previously known to the domain experts as suspicious cases. In addition, we also did a few controlled experiments in which known cases were added to the data to confirm that BSM correctly detected them as

candidates for further investigation. In this section we present some of our analysis results. All these examples are based on data for one year. Other time periods of interest include calendar month and quarter. In all our analyses p-values were estimated using Z = 9999 Monte Carlo experiments.

3.3.1 Receipt limits scenario

This scenario focuses on employee behavior with respect to business rules that set limits for travel expenses. Specifically, we consider a rule that states that only the actual expenses should be claimed and that the limits should not be viewed as an entitlement. Under this rule, we explore expenses incurred that do not require receipts to be submitted since they are below the corresponding specified limits. We seek to detect individuals who are likely violating this business rule and, in particular, we are looking for those who are trying to exploit the receipt limits by claiming expenses just below them (i.e., "flying just under the radar" behavior). Specifically, we will consider expense types that require a receipt above $25. Note that the converted US$ value will be presented in this paper even when the expense was incurred in another currency. We will also focus the analysis on expenses paid with cash (not by corporate credit card) over a one year time period. No covariates are used in this analysis. We present the results from two different firms {A, B} for this scenario.

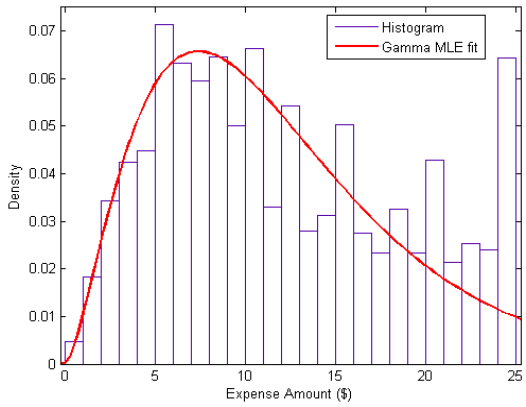


Figure 5. Histogram of expenses in firm A subject to $25 receipt limit and the corresponding Gamma fit

In the first firm A, the receipt limit of $25 is applied to expense categories like employee meals, business meals, ground transportation, parking, tips and tolls. Our analysis was performed on 660K expenses of these types that were below the $25 receipt limit. These expenses were claimed by 27K employees. The histogram of these expenses is shown in Figure 5. The maximum likelihood estimate for a Gamma distribution fit to these expenses has parameters {shape = 2.63, scale = 4.52} and the corresponding probability density function is also plotted in Figure 5. The histogram shows an increase in the counts near the maximum value of $25. It is important to note that the p-value computation described in Section 3.1 samples actual expense amounts and hence factors in the increased counts at the limit that occur across the firm. While this phenomenon of increased counts at the limit across the population is intuitive, a disproportionate increase in counts near the limit for any particular employee would be worthy of detection. Figure 6 shows the corresponding expenses for the top three employees in firm A identified by BSM in this scenario. The disproportionate concentration of expenses near the limit value of $25 is clear for these three employees.

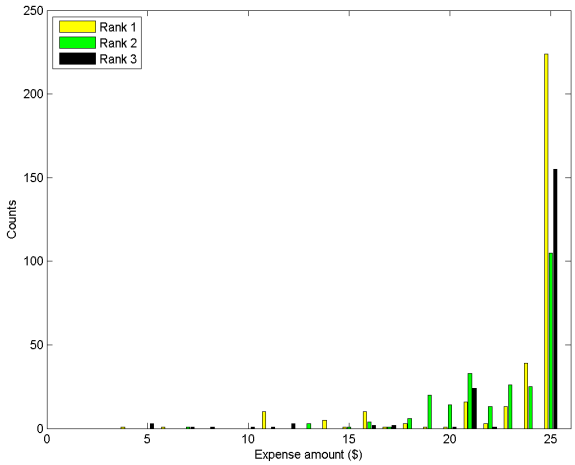


Figure 6. Histogram of expenses for the top three employees (in firm A) identified by BSM

Considering the second firm B, the receipt limit of $25 is applied to expense categories like employee meals, hotel, ground transportation, tolls/parking, tips and laundry. The data analyzed corresponds to a subset of the employees in the firm B. The analysis considered 110K expenses that were submitted by 3.6K employees. The histogram of these expenses is shown in Figure 7. The maximum likelihood estimate for a Gamma distribution fit to these expenses has parameters {shape = 1.76, scale = 4.98} and the corresponding probability density function is also plotted in Figure 7.

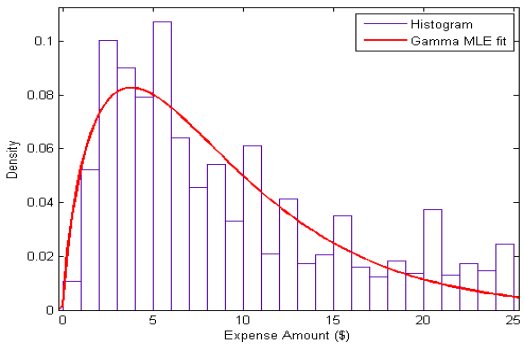


Figure 7. Histogram of expenses in firm B subject to $25 receipt limit and the corresponding Gamma fit

Figure 8 shows the corresponding expenses for the top three employees in firm B identified by BSM in this scenario. Again, the disproportionate excess concentration near the limit of $25 for these employees is clearly worthy of further investigation. Interestingly, the top employee in this case also has a concentration of $1 expenses (all corresponding to tips). We have observed a variety of expense amount patterns for the entities

identified by BSM. These would not be easily detected by simple filters considering disproportionate behavior in fixed expense amount windows below the limit.

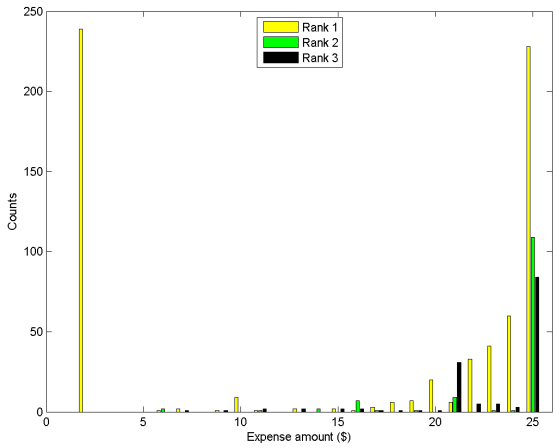


Figure 8. Histogram of expenses for the top three employees (in firm B) identified by BSM

3.3.2 Procurement analysis scenario

An important feature of our approach is the ability to do the analysis focusing on targets at different levels of granularity. Scenario 2 will utilize this capability by analyzing vendors, specifically hotel chains, to identify those that are significantly more expensive even after normalization for location is done. The analysis was done by considering the hotel room rates paid during business travel over a period of one year. The location (country, city) of the hotel was considered as a covariate for normalization. The average hotel room rate paid in each location was used as the normalization factor *F* (i.e., a normalized expense of 1 implies that the corresponding location's average room rate was paid). The analysis considered 523K expenses for hotel room nights that were paid to roughly 300 hotel vendors. The top ranked hotel vendor identified by BSM had significant usage (39K room nights) and a total excess charge of 9.3% after normalization by location. The second ranked hotel vendor identified by BSM had much less usage (around 2K) but a significantly higher percentage excess of 28.4% compared to the location based normalization factors.

The baseline of normalized room rates considering all the vendors and locations can be visualized using the cumulative distribution function (cdf) shown in Figure 9. The maximum likelihood estimate for a Gamma distribution fit for this baseline has parameters {shape = 7.78, scale = 0.129}. Figure 9 also shows the cumulative distribution function for the top two hotel vendors discussed above. Note that the ranking by BSM takes into account both the repetitiveness and the magnitude of excess compared to normal but the visualization in Figure 9 only depicts the latter.

In addition to the filtering of significant entities and their ranking, the BSM approach lends itself to providing diagnostic information that can help the user gain further insight on the identified entities. We have found it to be very useful to break down an identified entity's excessive deviation from the normal baseline by the covariate segments. For example, a single location is responsible for almost all of the excess exhibited by the second ranked entity. An excess of around 30% was charged by this entity at this location based on the data from all the relevant hotel vendors. This kind of diagnostic information can help focus the further investigation and corrective action.

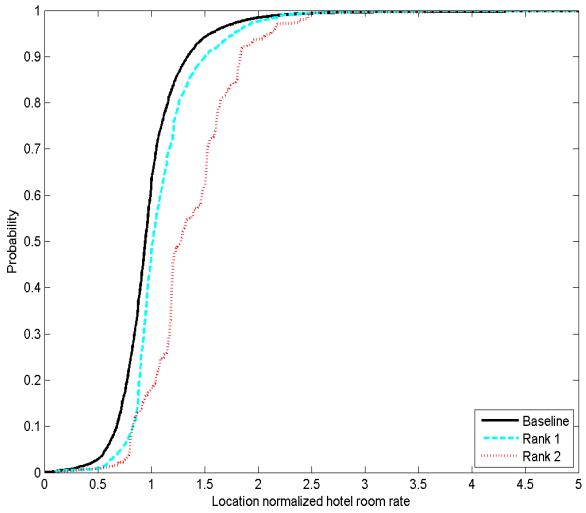


Figure 9. Cumulative distribution function for normalized room rate (baseline and BSM ranked top two hotel vendors)

3.3.3 Submission delay scenario

This scenario illustrates the application of BSM to other types of positive real valued quantities besides dollar amounts, for example, delays in submitting expense claims for approval. Organizations typically have a business rule specifying the maximum allowed time for submission after the expense was incurred. However, the guidelines typically suggest making an effort towards prompt submissions. Habitual delays in submission might indicate issues worthy of investigation even if the maximum delay limits are not always violated. This scenario identifies employees with repetitive excessive claim submission delays. The analysis was performed for firm B and focused on expenses charged to the corporate credit card.

The analysis considered 414K individual expense submissions from 4K employees over a period of a year. The average delay in claiming expenses was around 10 days for the baseline considering all 4K employees. The claim submission delay distribution was characterized by a Gamma distribution with maximum likelihood parameters {shape = 1.25, scale = 7.82}. The results for the top three employees ranked by BSM for having repetitive excessive delays are given in Table 2 and clearly show the repetitive deviation from the norm.

BSM Rank	Number of claims	Average submission delay
1	94	122
2	74	128
3	426	38

Table 2. Expense claim submission delays for the top three employees in firm B identified by BSM

4 Event Count Scenarios

In this section we consider scenarios involving discrete occurrences of events such as approvals of exceptions to specific business rules. A typical template would aim to determine if an entity has excessive (or insufficient) counts for a specified event type given the counts for the opportunities for the events. For example, consider the scenario from Section 1 to identify organizational units with excessive approval of exceptions to the prescribed class of air travel. Clearly, for this scenario we would need to consider, for each organizational unit, both the number of air travel expenses claimed and the number of air travel class exceptions that were approved. It might seem intuitive to consider some attribute of the air travel as a covariate, e.g., international versus domestic travel. However, our experience is that business controls professionals do not make accommodations for such attributes (beyond any use of such attributes in the corresponding business rule) when assessing if an organizational unit is being lax. There are other scenarios where the use of covariates is more appropriate. One such example is the approval of exceptions to the business rule that defines when receipts have to be submitted for T&E expense claims. There could be different reasons provided for why, on occasion, a receipt is missing (e.g., receipt lost, receipt not available). The rates of occurrence and approval of missing receipt exceptions clearly varies by expense type. For example, it is typically the case that missing receipt exception rates for hotel room expenses are low. On the other hand, missing receipt exception rates for ground transportation expenses like cab fares are much higher. Therefore, the expense type is an appropriate covariate when we are trying to detect organizational units with excessive approvals of missing receipts exceptions.

4.1 The Model

Our approach to detect entities with excessive (or insufficient) counts for specific events is similar to the one for expense amount scenarios in the use of the LRT. The LRT based on a Poisson model is well suited to model event counts that are proportional to known opportunities with possible categorical covariates. The LRT using the Poisson model has been used extensively in various surveillance applications (especially in public health) following the work on the spatial scan statistic by Kulldorff 1997. Indirect standardization was proposed in that work as one approach to handle categorical covariates. Let $O(\xi, F)$ and $V(\xi, F)$ represent the count of opportunities and the count of target event occurrences for entity ξ for the combination F of categorical values for the covariates, respectively. The expected number of target event occurrences $X(\xi)$ for an entity ξ is calculated using indirect standardization as:

$$X(\xi) = \sum_{F} \left\{ \left(\frac{\sum_{\xi'} V(\xi', F)}{\sum_{\xi''} O(\xi'', F)} \right) \times O(\xi, F) \right\}.$$

Following Kulldorff 1997, the test statistic $W(\xi)$ for Poisson model is given by

$$W(\xi) = Y(\xi) \times log\left(\frac{Y(\xi)}{X(\xi)} \right) + \left(U - Y(\xi)\right) \times log\left(\frac{U - Y(\xi)}{U - X(\xi)} \right),$$

where $Y(\xi)$ represents the aggregate number of target event occurrences for entity ξ over all combinations of categorical covariate values and U represents the total number of occurrences of target events over all the entities and the covariate value combinations.

The p-value is computed by performing a number Z of Monte Carlo experiments, where, in each experiment the target event counts for an entity ξ are determined by sampling from a Poisson distribution with mean equal to the expected count $X(\xi)$. As before, the p-value for the entity ξ is computed using the formula $(L+1)/Z$, where L is the number of Monte Carlo experiments with a maximum test statistic exceeding $W(\xi)$. Entities with p-values that reject the null hypothesis at the prescribed α level are candidates for further investigation and are ranked in decreasing order of their test statistic values W. The behavior of the LRT model using the Poisson model has been well-studied given its wide usage in domains like public health and epidemiology. In the next subsection we present results on production T&E data that demonstrate its applicability to this domain.

4.2 Application to Production T&E Data

As described earlier in Section 3.3, we applied our model for event count scenarios to production T&E data from multiple firms in an enterprise expense reporting environment (GERS) and reviewed the results of various scenarios with audit and business control professionals. In this section we will present results from two of these scenarios. The chosen scenarios will also illustrate the ability of our approach to do the analysis at different organizational levels. This is important feature since business controls are typically exercised by monitoring expenses for organizational units that are more suitable for expense management and policy guidance.

4.2.1 Hotel limit exceptions scenario

This scenario is related to the business rule that specifies upper limits by location on hotel room rates and requires management approval of exceptions to this rule. The goal of the analysis is to identify organizational units that are approving exceptions to this rule excessively. The analysis was done for firm B targeting 15 organizational units. In the time period of the year considered, there

were 4.6K exception approvals (events) for 43K underlying hotel expenses (opportunities) implying a baseline event rate of 10.7%. The top three organizational units identified by BSM as having significantly excessive ($\alpha = 0.01$) exception approvals are listed in Table 3. Clearly, the counts of approval events and opportunities indicate patterns of excessive approvals in these three organizational units that warrant further investigation.

BSM Rank	Number of hotel expenses	Number of hotel limit exception approvals (expected number)	Poisson test statistic W
1	777	235 (83.2)	99.75
2	609	144 (65.2)	35.96
3	1371	247 (146.8)	29.43

Table 3. Results for the top three organizational units identified by BSM as having excessive hotel limit exception approvals

4.2.2 Missing receipt exceptions scenario

This scenario addresses the business rule that requires submission of receipts based on the expense category and amount. The goal of the analysis is to identify approvers who are approving exceptions to this rule excessively. As discussed earlier, the rates of occurrence and approval of missing receipt exceptions across the firm clearly varies from one expense category to another. Therefore, the expense category is an appropriate covariate for this analysis.

BSM Rank	Number of exception opportunities	Number of exception approvals (expected number from indirect standardization)	Poisson test statistic W
1	403	245 (22.4)	363.5
2	1255	375 (72.5)	314.7
3	624	234 (25.7)	309.1

Table 4. Results for the top three approvers identified by BSM as having excessive missing receipt approvals

The analysis was done for firm A considering the exception approvals over a one year period. The analysis considered 18K exception approvals by 12K approvers that resulted from 159K opportunities for this exception. Table 4 shows the results for the top three approvers identified by BSM as having excessive approval rates after normalization by expense categories. Table 4 clearly indicates the repeated approvals by these approvers and its excessiveness when compared to expected numbers based on behavior across all approvers. Examining the diagnostic information for the top ranked approver in Table 4 led to the actionable insight that the dominant expense categories for the corresponding exceptions were employee lunch and dinner and also that one employee was the main contributor.

5 Discussion

The diversity of the application areas for fraud detection has been pointed out by Bolton and Hand (2002). Bolton and Hand also stress that operational and data characteristics of the application domain determine suitable fraud detection methods and tools. Analysis of expense claims for audit and business controls purposes is an application domain with specific characteristics and requirements. The models and methods presented in this paper address the following needs in this domain:

- Conservative analysis that identifies entities for further investigation when significant evidence is available.
- Entities analyzed at various levels of granularity in the firm based on the scenario and the corrective action that will follow.
- Analysis that can handle the data and operational characteristics like lack of labeled data, significant tails in the value distributions, impact of business rules (e.g., limits), and the need for normalization considering one or more covariates.
- Provide detailed evidence for the entities identified to help audit and business controls professionals determine if an investigation is warranted and to bootstrap it if the investigation is pursued.

Our simple and intuitive template structure has been used to create over 50 specific scenarios for the analysis of T&E data in an enterprise expense reporting environment. Our scenarios utilize only the structured data logged in the expense claim process. Unstructured data for the entities identified like explanations for triggering exceptions are presented as part of the evidence used for further investigation. Including the unstructured data in the automated analysis is unlikely to be useful due to its unreliable nature (inconsistent and possibly inaccurate or even misrepresented information).

Future work also includes utilizing the BSM scoring of entities based on their outlying behavior to impact the controls and management actions for selected entities within the travel expense management system.

The BSM model has also been applied to other domains like procurement (one such scenario was illustrated in Section 3.3.2). Our ongoing work in other domains suggests that BSM can be a valuable part of a toolkit for identifying entities with outlying behavior in various domains.

6 Conclusion

We have described a set of Behavioral Shift Models developed in the context of Travel and Entertainment (T&E) expense management for efficient auditing and business controls. Our models combine recent advances in unsupervised statistical analyses with T&E domain knowledge to profile and rank entities in a firm based on the deviation of their travel spending behavior from that of the general population. The focus is on repeated suspicious behavior as opposed to a one-time outlying case, in line with the domain practice of conservative filtering that takes into account the amount of evidence available. We have modeled two broad classes of data: one for continuous, real-valued variables and one for discrete, Poisson-type variables covering a large number of scenarios in the T&E domain. We characterized the discriminating power of our method using a systematic simulation approach that evaluates the detection capability of the BSM for different data distributions with different amounts of engineered deviations from the population norm. Lastly, we have presented several example scenarios with validated results of our analyses of T&E data from several firms.

7 Acknowledgements

We would like to thank the audit and business controls professionals who shared their deep knowledge of this domain with us and helped validate our results.

8 References

Beasley, M.S., Carcello, J.V., and Hermanson, D.R. (2001): Top 10 Audit Deficiencies, *Journal of Accountancy*, *American Institute of Certified Public Accountants*.

Bolton, R. J. and Hand, D.J. (2002): Statistical Fraud Detection: A Review (with discussion), *Statistical Science*, 17(3), 235-255.

Bolton, R. J. and Hand, D.J. (2001): Unsupervised Profiling Methods for Fraud Detection, *Credit Scoring and Credit Control VII*, Edinburgh, UK.

Christianini, N., and Shawe-Taylor, J. (2000): *Support Vector Machines*, Cambridge University Press, Cambridge.

Denning, D. E. (1987): An Intrusion-Detection Model, *IEEE Transactions on Software Engineering*, Vol. SE-13(2).

Duda, R.O., Hart P.E., and Stork, D.S. (2001): *Pattern Classification*, John Wiley & Sons.

GERS: IBM Global Expense Reporting Solutions, http://www-935.ibm.com/services/us/index.wss/offering/igs/a1009035

Glaz, J., Naus, J., and Wallenstein, S. (2001): *Scan Statistics*, Springer-Verlag, New York.

Hodge, V., J., and Austin, J. (2004): A Survey of Outlier Detection Methodologies. *AI Review*, 22 , 2004, pp. 85-126.

Huang, L., Kulldorff, M., and Gregorio, D. (2007): A Spatial Scan Statistic for Survival Data, *Biometrics* 63 (1), pp. 109-118.

Kulldorff, M (1997): A Spatial Scan Statistic, *Communications in Statistics: Theory and Methods*, 26:1481-1496.

Lehmann, L.E. (1986): *Testing Statistical Hypothesis*, Springer-Verlag, New York.

National Business Travel Association (NBTA): http://www.nbta.org/About/TheValueofManagedTravel.

Williams, G., Baxter, R., He, H., Hawkins, S., and Gu, L. (2002): A Comparative Study of RNN for Outlier Detection in Data Mining, International Conference of Data Mining & CSIRO Technical Report CMIS02/102.

Customer Analytics Projects: Addressing Existing Problems with a Process that Leads to Success

Inna Kolyshkina
Institute of Analytics Professionals of Australia
20, 9-19 Nickson Street Surry Hills NSW 2010
inna@iapa.org.au

Simeon Simoff
University of Western Sydney
NSW 1797, Australia
s.simoff@uws.edu.au

Abstract

This article explicitly outlines an approach designed to allow optimal utilisation of Analytics in the industry setting. The paper focuses on the key stages of the Analytics process that have not been identified in previous Analytics methodologies and draws on industry, consulting and research experience to show that correct design of the project trajectory can allow the industry to fully realise the benefits that Analytics has to offer. As the case studies provided demonstrate, it is often the skipping of key stages, especially the preliminary analysis stage, that are currently responsible for preventing success of an Analytics project. It has been shown how, using the outlined approach, project can achieve maximum effectiveness and business buy-in.

Keywords: Customer Analytics, Industry Analytics, Data Mining Effectiveness, Analytics Projects Management, Analytics Industry Case Studies.

1 Introduction.

The vast amount and complexity of the information we have to deal with on a daily basis is a challenge for contemporary business decision makers. From a business perspective analytics can be defined as a subset of what has come to be called business intelligence—a set of technologies and processes that use information and data to understand and analyze business performance (Davenport and Harris, 2007).

"In today's competitive, globalized market where not much more than the making of good decisions separates competitors from each other, analytics is the emerging new business adviser, a guide that utilises the business data to generate guidance regarding making better business decisions." (Davenport and Harris, 2007). The science of analytical reasoning provides the reasoning framework upon which strategic and tactical analytics technologies are built.

"At a time when companies in many industries offer similar products and use comparable technology, many of the previous bases for competition are no longer viable. In a global environment, physical location is frequently not a source of advantage, and protectionist regulation is increasingly rare. Proprietary technologies can often be rapidly copied and attempts to achieve breakthrough innovation in products or services often fail. What's left as a basis for competition is execution and smart decision making. An organisational commitment to and developed capability of Analytics is enabling market-leading companies to succeed in the rapidly evolving arena of global competition." (Davenport and Harris, 2007)

The information and data mining software systems facilitate the analytical reasoning process, providing humans with means to deal with the enormous amount of data and information generated in various areas of human endeavour. Since its inception in the late 80s, data and information mining technologies have reached the level of embedded technology, coming as part of modern data management and analysis suites. However, technology is only one of the necessary conditions for achieving competitive advantage. The issue of Analytics not being fully utilised in organisations due to lack of a clearly defined analytics process has been recognised for some time. In the late '90's – early 2000's a number of methodologies was developed to address this issue. Among them CRISP-DM (Chapman, et al., 2000) is perhaps the best known and broadly used iterative data mining methodology. However such methodologies are focussed primarily on the technical aspects of the data mining process with little attention to the business aspects of the overall Analytics process (Pyle, 2004). For instance, "Business Understanding" is part of CRISP-DM, however, little is provided about how that actually can be done. There is an embedded assumption that the business analysts will somehow communicate with the data miners and in this communication the data mining models will be related to business key performance indicators (KPIs). However, industry leaders have pointed out the existence of a communication gap between data

mining experts and business domain experts (Fayyad, 2004). This gap, together with some related issues, has been explored in Van Rooyen's (2004) critical evaluation of the project management utility of CRISP-DM and Data Mining Projects Methodology (DMPM) of the SAS Institute, in a business decision-support environment.

Recently there have been attempts to improve existing Analytics methodologies and allow them to become more effective and reliable in providing useful insights in business contexts. A notable contribution to the field is Van Rooyen's (2005) Strategic Analytics Methodology (SAM). In the last few years there has been an improvement in the use of Analytics in business settings. However, it is important to stress that the potential value of Analytics has not been fully realised or utilised in business settings as yet.

This paper presents an Analytics process approach that aims to maximise the value-add of Analytics projects. The process draws on the extensive knowledge and experience gained from industry, consulting, research and education activities in Analytics. The paper focuses specifically on the key process stages that have not been identified or given sufficient attention in previous Analytics methodologies. Further the paper is organised as follows. Section 2 considers the reasons why analytics projects may fail. Section 3 discusses solutions to the problems discussed in Section 2. Section 4 illustrates and reflects on the practical application of the stage model of the Analytics process.

2 Why do Analytics Projects under-deliver?

Analytics is a new area and as such, is currently hindered by two main issues:

- the lack of business acumen and experience on the part of the analysts in business contexts, as well as;
- limitations of the awareness of the potential value of Analytics in the business world.

Although, these issues are temporary in nature, they influence the outcomes of Analytics efforts. As a result it is not unusual for an Analytics project to under-deliver. This is widely recognised in the industry. Citing Dragoon (2006), "many segmentation efforts today are an exercise in futility... Organizations often wind up with segments that drain resources, instead of with segments that lead to more effective ways of running the business or meeting customers' needs." Academic research has also recognised these problems (Van Royeen, 2005).

The specific reasons behind the value-add potential of Analytics not being fully realised have been discussed since late '90s – early 2000's. At that time, the reasons suggested by industry experts (Fayyad, 2004) and Pyle (see Figure 1 - Rule 2, 4, 5, 7, 8), were mainly surrounding ineffective communication between analysts and business institutions due to business institutions' lack of knowledge about Analytics' capabilities. The reasons also included such factors as analysts' focussing on the technical solutions instead on the business solutions that are oriented at the impact on the bottom line.

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

Figure 1: Pyle's nine rules for Analytics project failure (adapted from (Pyle, 2004))

The need to resolve this problem led to the development of approaches that would ensure the maximum value-add of Analytics. For example, in 1999-2000 two major data mining project methodologies were specifically designed to more effectively help in business settings, these were DataMining Projects Methodology (DMPM) (SAS Institute, 2000), and CRISP-DM (Chapman et al., 2000).

However, while these approaches have been adopted by a substantial part of the Analytics community the capacity of Analytics projects has still not been fully realised. That these approaches have not fully covered the important features of an Analytics project is evidenced by the recent move to develop a new version of CRISP-DM methodology (CRISP-DM 2.0).

A detailed discussion and evaluation of the above approaches is presented in (Van Rooyen, 2004). Van Rooyen's statement that "despite the potential for the output of data mining to support business decision making, data mining practitioners are still finding resistance to the uptake of data mining by the business community" (Van Rooyen, 2004, p. 86.) is confirmed by industry and consulting experience expressed at a number of Analytics conferences and other forums in Australia and internationally.

Part of the answer is in the fact that there are many examples of projects where Analytics has under-delivered. Further, we explore the nature of the factors that have undermined the success of Analytics projects, based on a number of case studies involving Analytics projects from across the industry, ranging from transportation to banking.

2.1 Lack of congruency between business issues and analytics targets.

A widely recognised reason of Analytics projects' failure is the lack of congruency between the issues seen as central by the business and the issues actually targeted and addressed by the Analytics projects. The reason behind this may be two-fold:

- It may be due to a lack of clarity of the problem to be addressed on the part of the business, or;
- It can also stem from a lack of clarity and specificity in understanding of the key issues on the part of the analysts.

Currently, this scenario is so common that the existing version of CRISP-DM includes a specifically designed step to allow the abandonment of the project for this very reason (see Van Rooyen, 2004).

A typical example from the industry that clearly characterises the above described issue, is that of a financial institution which had the objective of using Analytics to improve their mortgage customer retention strategy. In terms of the project objectives it was not clearly specified whether the project should be aimed at decreasing current customer churn within a certain period of time, and, if so, what this period of time is (for example within next 12 months), or at refining bank's acquisition strategy so that customers applying for mortgage are better screened. The nature of the business goal would then require a unique analysis approach, which would lead to different outcomes and different implementations recommendations.

The lack of clear communication and understanding of the business' main concerns and requirements in this project led to confusion and subsequently the project was delayed. This could have easily been avoided had there been a greater focus early on identification and clarification of project goals. This could have been carried out via a process of thoroughly interviewing the main stakeholders. However, lacking this step, the ill-defined goals fundamentally undermined the project from its inception.

It is important to note that industry and especially consultancy are becoming increasingly more effective in addressing this issue. Consultancies draw on their experience of clearly establishing client needs and providing fitting solutions. Also, industries where Analytics units are practically in the position of internal consultants to the business find that these units more often provide clear lines of communication for the early formulation of specific project objectives that match business requirements.

2.2 Lack of Analytics project management expertise.

Another major reason for failure of Analytics projects is the lack of Analytics project management expertise, and, in particular, the habit of skipping important early stages in the Analytics project development process. These lead to such issues as the use of unreliable or incomplete data, lack of robustness, overlooking of important drivers of the business issues, and incorrect choice of Analytics tools.

Also, a related issue is project scope creep which sometimes leads to withdrawal of funding and to project termination.

All of these issues result not from the deficiency of Analytics as such, but from improper utilisation of this useful instrument. In other words, Analytics becomes a potentially valuable resource that has not been tapped into correctly. Such outcomes not only hinder organisation's ability to compete, they unjustifiably undermine the credibility use of Analytics for improving business decision.

We use several case studies to illustrate different aspects of the problem.

2.2.1 Incomplete insights due to oversight of important specific drivers of the business issue that the project aimed to address.

A regional airline with separate branches in two states wished to increase its profit by choosing the right staff KPI's. The Analytics project approach involved gathering the established KPI's for combined operations, and correlating these with profit. However, the analysis was unsuccessful. The KPI's selected (such as number of tickets sold) did not sufficiently drive profit. Upon review it was identified that the unclear outcomes of the analyses were due to unaccounted factors such as differences in the nature of the services provided by the two branches and significant environmental changes (emergence of a competitor within the industry and increases in fuel cost). In hindsight it is clear that having a preliminary step in the analysis process would have been fundamentally advantageous for the project. In other words the project was severely limited due to a lack of early clarification of the significant industry specific factors surrounding the data.

2.2.2 Using outdated Analytics techniques leads to limited insights.

A major private health insurer wanted to address an increase in customer churn and rising numbers of expensive claims. More specifically, the company needed a way to identify potentially expensive claims to allow more efficient management and early intervention. Analytics was used to achieve these goals in the following way. Domain experts' opinions were sought and used to identify the 5-6 factors which may influence churn. Cross tabulations and logistic regression modelling was used to see which customers are most at risk of churning based on the identified factors. Age was used as the main predictor of claim cost. Domain expertise was utilised to determine what other 5-6 factors may influence claim cost. Cross-tabulations were used to quantify the influence.

Upon completion of the analysis it was found that the project was not successful since the factors chosen for the analysis explained less than 40% of the claim cost and churn. Once again this is suggestive of a lack of early Analytics-business dialogue regarding the necessary steps to identify the significant factors impacting on the business outcome of interest.

2.2.3 Lack of proper data quality assessment and preliminary analysis gives misleading conclusions.

In the case of an insurance company, a model aiming to predict the probability of a workplace claim suggested the claimant's salary to be the best predictor of that. This suggested that those who are paid more, were more likely to claim, which was a counter-intuitive result, not consistent with the domain expertise. Further data

investigation showed that the "claimant' salary" field was filled out only after the claim occurred, otherwise the value was missing, and was read in as "0" by the data extracting tool. In other words, the fact of claim drove the fact of "claimant' salary" field having value greater than 0, rather than salary value drove the probability of a claim.

2.2.4 Inappropriate choice of software tool leads to productivity decrease and poor staff morale.

A communications company was overwhelmed by number of cases of potential fraud to investigate. An Analytics model was designed externally and implemented in order to evaluate likelihood of fraud. The rationale was to screen potential fraudulent cases, and allow the fraud investigators to focus on those that are most likely to be fraudulent. However, the model merely provided the investigators with a numerical score. The users of the model were not given information of how and from what data the scores were generated by the model. Eventually, the investigators ended up ignoring the outputs provided by the model as they felt that top-scores were often not indicative of greater likelihood of fraudulent cases. They disregarded the output and retreated back to relying more on their own "hunches" than on fraud model output. As such the implementation of the model hindered, rather than improved team productivity and morale.

2.2.5 Non-implementable insights of analysis

A major communication company wanted to predict customer churn by analysis of recent 2 months' data on customer behaviour. The solution built predicted whether or not a customer left the company on the third month. The model achieved good technical results in terms of accuracy and robustness, however could not be implemented as after it was established that a customer was likely to leave next month, there was no time period when organisation could intervene in order to retain the customer.

2.2.6 Lack of stakeholders' buy-in on the implementing of analysis results.

Sometimes insights implementation does not happen due to the difficulty of the interpretation of the insights (for example, model scores without explanation of the practical meaning behind the figures), high implementation cost or lack of fit with the organisational strategy. Such issues can be prevented by constant communication, providing documentation, which interprets the analytics output in a clear, easy to understand form, and analyst's understanding of organisational strategy and implementation cost.

2.2.7 Lack of monitoring and updating processes.

Monitoring and updating is important as due to changes in economic environment, political situation and other factors, analytics results become after a while out of date. There has to be a documented time frame for model review agreed with the business. This stage seems obvious, but often is not in place. For example, a mortgage retention model in a major bank had not been updated in 3 years which led to poor results in marketing campaigns based on the model in the third year of model deployment and as a result, affected trust between business and analytics units in the bank. Van Rooyen formulated this problem as "concept drift" in a broader way than it is formulated in machine learning and addressed it in (Van Rooyen, 2005).

3 How to Maximise Analytics Project Success.

In this section we present some solutions to the problems discussed in Section 2. We start with focusing on what needs to be put in place in an organisation to enable it to obtain maximum benefits from Analytics projects.

3.1 Enabling organisational awareness

3.1.1 Providing awareness at all levels of management about the capabilities of Analytics

A crucial point is the provision of awareness at all levels of management within the organisation of what Analytics can do for them. As Davenport and Harris (2007) note, "organizations that want to be competitive must have some attribute at which they are better than anyone else in their industry—a distinctive capability." To achieve this, a critical factor is the awareness at all levels of the organisation that Analytics can add value to the business. Ideally, Analytics should be a trusted advisor to the business.

The necessary requirements for that to occur are:

- an awareness of all stakeholders at the senior and middle management level of what Analytics have to offer;
- the development of a blend of Analytics and business skills and knowledge within the organisation;
- development of effective project management approaches specially suited for Analytics projects.

The development of a blend of Analytics and business skills and knowledge combined with knowledge and skills in project management approaches specially suited for Analytics projects should be an essential component in University analytics degrees.

3.1.2 Providing organisational awareness at all levels about the keys to successful utilisation of Data

To achieve Analytics success, organisation should ensure Availability, Quality and Accessibility of data, as well as developed data understanding.

3.2 Enabling knowledge and skills in Analytics

3.2.1 Updating Knowledge and Skills in Analytic Techniques

The existing communication gap between analysts and business can be effectively reduced by encouraging development of analysts' business acumen and related skills. Every year new, more effective Analytics techniques and tools are being developed. Industry analysts should constantly build awareness of:

- industry trends;
- approaches that have proven to be successful, and;
- new techniques and knowledge regarding skilful implementation of these techniques in analytics projects.

3.2.2 Updating Knowledge and Skills in Analytic Software Tools.

Analytics software space is very dynamic, with new exciting players entering the market. In fact sometimes the degree of benefit from new software is surprising. For example, large organisations such as ATO get good value from using Open Source software in specialised projects as well as getting good return out of high-end multifunctional Analytics software. The lesson is: remain informed of new offerings at the Analytics software market to ensure that you get the right software for the job. Organisations would benefit by ensuring that analysts are given time to do these steps periodically.

3.3 Enabling the Analytics process

However, even if all of the necessary elements described in sections 3.1 and 3.2 are in place, the organisation needs a clear framework of the process that the Analytics project goes through. This is a key component to ensure that future Customer Analytics projects work.

Figure 3 shows the steps in the process model of an Analytics project. While all steps in the model are crucial for the success of the Analytics project, most issues with Analytics projects currently arise due to the skipping of stages linked to the preliminary analysis. These stages involve thorough scoping and the discovery/resolution of any hidden issues and factors which may impact upon the trajectory or the quality of the main stage of analysis. The Preliminary Analysis step should precede the scoping stage of the main analysis phase, which will be driven and guided by the discoveries made during preliminary analysis.

3.3.1 Is the preliminary analysis stage really necessary?

The preliminary analysis stage is essential in order to be able to scope the main analysis stage. Data can hide surprises and this stage allows uncovering and screening any hidden factors so that the main analysis stage is scoped correctly. The mentality is to prevent rather than treat. For example, often stakeholders may try to influence the project by suggesting the technical approach that they tried in the past, but which might be inappropriate in the current project. In such cases the preliminary analysis' stage's results will provide the analyst with the information necessary to make informed, evidence-based decisions regarding the best Analytics techniques to apply and software to use. Clearly, if this is not addressed, main stage of analysis is likely to be compromised. For example, a lack of clarity in early project planning may easily lead to an inappropriate choice of analysis technique, and often it can result in using "tried and proven" but outdated Analytics techniques such as correlation analysis, cross-tabulations or linear regression. In such cases such a decision can impact the time frame and cost of the project due to decrease in productivity as well as hinder the quality of insights and the decisions based upon them. This can be prevented if Preliminary Analysis results are documented and explained to the business stakeholders.

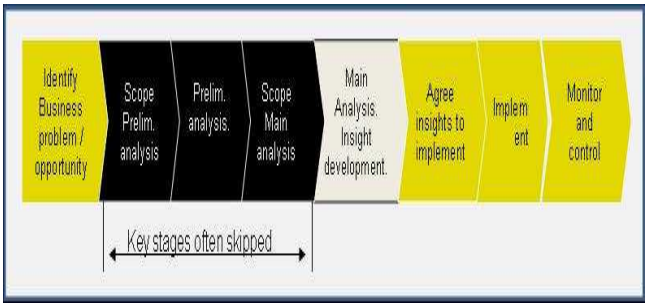


Figure 3: Stage model of the Analytics project process.

3.3.2 Deliverables of Preliminary Analysis

To ensure maximum effectiveness, the deliverables of Preliminary Analysis should be as follows:

- Extracted preliminary data
- Documented basic data summaries and descriptions
- Confirmed data definitions
- Availability of the data fields required to achieve the project goals
- Documented identified data issues and recommendations how to deal with them
- Recommendations on the additional data (internal or external such as Mosaic scores) that may be required to achieve the project goals
- Documented insights to help shape the main stage of analysis such as recommendations on
    - the most important data fields to use
    - presence of homogeneous groups that should be analysed separately as well as any outlier groups
    - Analytics techniques and methods to use
    - software
- Documented insights/hypotheses/unexpected results that may need to be considered at the Main Analysis stage.

3.3.3 Relating results to business

The step "Discuss results with business. Agree insights to implement." (shown in light grey background in Figure 3) is another key stage that sometimes is not fully emphasised. Unless the stakeholders' buy-in is achieved and the business understands the insights, agrees with them, is happy to implement them and has a clear view on how to implement, there is a risk of the project insights not being correctly implemented or not being implemented at all. In case that occurs, it obviously would prevent the project of adding value and affect the level of credibility and trust between Analytics and business.

The remaining key steps were described adequately by methodologies such as Crisp-DM. "Monitor and Control" step and its importance has been discussed and described in detail in (Van Rooyen, 2005).

4 Implementing the stage model of the Analytics process

In this section we illustrate and reflect on the practical application of the stage model of the Analytics process, presented in Figure 3 in order to resolve some of the issues presented in Section 2.

4.1 Using Preliminary Analysis Stage to address the issues with Airline Customer Analytics

In the case of the two-branch regional airline customer discussed in section 2.2.1, the preliminary analysis showed that all KPI's were strongly correlated with each other. This discovery meant lack of validity of recommendations made by the previous analysis. It also established some discrepancies which pointed at lack of data necessary for correct insight development. The behaviour of the two branches to see if they can be combined or they behave differently, on the basis of this it was decided that the regions needed to be analysed separately.

Based on the results of preliminary analysis, a project plan was developed such that the analysis included relevant data such as, fuel competitor activities, cost and service charges. The use of a team of business domain experts and Analytics experts to interpret the Analytics results and selected the KPI's that were strongly linked to sustainable increased profits. The project was a success as it delivered insights into what drives profit in each operation, and how structural change will complement the incentive program.

4.2 Using Preliminary Analysis Stage to address the issues with Health Insurance Analytics

When addressing the problem described in section 2.2.2 in the case of Analytics for Health Insurance, the preliminary analysis showed that different age groups have different churn and claim cost drivers. For example, customers aged over 60 were more likely to churn if they pay their premiums by physically attending branch. Additionally, the preliminary analysis allowed to do quick exhaustive search on 300 available data fields and identified all fields that were predictive of churn (about 25) and all fields that were predictive of claim cost (about 20). This increased dramatically the accuracy of results and built credibility with the business. In the main analysis stage, valuable and practical insights relating customer behaviour to claim cost were delivered. For example, the customers who had larger number of physiotherapy services were more likely to have a hospital claim. Overall, the project helped to identify strategies to retain high-value customers in risk of churning and helped the business forecast the expected increases in profit.

4.3 Expanding the range of Analytics techniques to address the issues with the customer retention in a Telecommunications Company.

The conservative part of the business community, in particular, in the industry as opposed to consulting, tends to be cautious of innovative sophisticated Analytics techniques. It is felt that it can hinder the buy-in of the stakeholders and adversely impact the project success. In this section we demonstrate the importance of the selection of the appropriate analytics technique during the Preliminary Analysis. Such selection may require "thinking out of the box" on the part of the analyst. In this example, the change was in the shift to another data source – unstructured and semi-structured text data and the implementation of recently developed text mining techniques and tools.

A major telecommunications company had many customers churning after 3 months. Insight was needed into which customers were at risk of leaving, why they were leaving, and which customers needed to be retained.

Text mining was implemented to inbound call centre records of the conversations between customers and staff. The analysis uncovered early warning signs of customer churn, allowing the development of strategies to "save" customers before they had committed to leaving the company. Furthermore, the project allowed the building profiles of the different segments of at-risk customers, showing their needs, motivations and their value to the company.

The insights from textual data provided the company with improved accuracy of churn prediction by 12%. The company was thus able to prioritise its retention resources on profitable, risky customers first, to deliver maximum return on investment.

5 Conclusions

Analytics is the tool that has the proven potential to give valuable insights into ways of helping organisations to address improving revenue growth, customer retention and other such fundamental goals. It has the value bringing into light key hidden factors holding a business back from achieving its full potential, and as such allow skilful ways of approaching overcoming the restraints posed by these factors. Foregoing such an advantage is the cost of not utilising the strength of Analytics as part of the company's competitive artillery.

Among the necessary conditions for the success of Analytics projects in an organisation are proper data quality and availability, the right choice of analytics software and organisation's awareness of analytics' potential.

However, even if such foundation is in place, the effectiveness of Analytics is dependent upon whether a correct methodological approach is followed. This article explicitly outlines an approach designed to allow optimal utilisation of Analytics in the business context. As the case studies provided in this article demonstrate, it is often the skipping of stages, especially the preliminary analysis stage that are currently responsible for preventing success of an Analytics project. It has been shown that in practice these stages, which involve primarily effective planning as well as constant business-analyst dialogue, are absolutely crucial for the project effectiveness. Thus, paying careful attention to early design of the project trajectory can allow businesses from across the industry to fully realise the benefits that Analytics has to offer.

6 References

Chapman, P., J. Clinton, et al. (2000). CRISP-DM 1.0: Cross Industry Standard Process for Data Mining, CRISP-DM Consortium.

Chatfield, C. S. and T. D. Johnson (2000). Step by Step Microsoft Project 2000. Redmond, Washington, Microsoft Press.

Davenport, T. H. and Harris J. G. (2007) Competing on Analytics. The New Science of Winning. Harvard Business School press. Boston, Masssachusetts

Dragoon, A. (2006). How to do customer segmentation right. *CIO Magazine*, 8 March 2006, Accessed August 10, 2007

Fayyad, U. (2004). "Editorial." *SIKDD Explorations* **5** (2).

Ganti, V., J. Gehrke, et al. (1999). "Mining Very Large Databases." IEEE Computer 32(38): 6875.

Han, J. and M. Kamber (2001). Data Mining: Concepts and Techniques. San Francisco, Morgan Kaufmann Publishers.

Hastie, T., R. Tibshirani, et al. (2001). The Elements of Statistical Learning. New York, Heidelberg, Berlin, Springer-Verlag.

Klinkerberg, R. and T. Joachims (2000). Detecting Concept Drift with Support Vector Machines. In Proceedings of the Seventeenth International Conference on Machine Learning, San Francisco, Morgan Kaufmann.

Pearce, I. J. A. and J. R. B. Robinson (1991). Strategic Management: Formulation, Implementation, and Control. Boston, Irwin.

Pyle, D. (1999). Data Preparation for Data Mining. San Francisco, Morgan Kaufmann Publishers.

Pyle, D. (2004). This Way Failure Lies. DB2 Magazine. 2004.

SAS Institute (2000). Data Mining Projects Methodology. Cary, NC, SAS Institute Inc.: 133.

Van Rooyen, M. (2004). An evaluation of the utility of two data mining project methodologies. In Simoff, S. J. and Williams, G. (eds), *Proceedings of the 3rd Australasian Data Mining Conference*, 6-7th December, 2004, Cairns, Australia, pp. 85-94.

Van Rooyen, M. (2005). *Strategic Analytics Methodology*. Master of Science Thesis, Faculty of Information Technology, University of Technology, Sydney.

Zikmund, W. G. (2003). Business Research Methods. Various, Thomson South-Western.

Predictive Model of Insolvency Risk for Australian Corporations

Rohan A. Baxter, Mark Gawler, Russell Ang

Analytics, Office of the Chief Knowledge Officer,

Australian Taxation Office,

P.O. Box 900, Civic Square, ACT 2608

{firstname.lastname@ato.gov.au}

Abstract

This paper describes the development of a predictive model for corporate insolvency risk in Australia. The model building methodology is empirical with out-of-sample future year test sets. The regression method used is logistic regression after pre-processing by quantisation of interval (or numeric) attributes. We show that logistic regression matches the performance of ensemble methods, such as random forests and ada boost, provided that pre-processing and variable selection is performed.

A distinctive feature of the insolvency risk model described in this paper is its breadth; since we are using income tax return data we are able to risk score one million companies across all industries, all corporation types (public, private) and all sizes, as measured either by assets or number of employees. This is an application paper that uses standard credit scoring methodology on a new data source. The contribution is to demonstrate that insolvency risk can be estimated using income tax return data. The corporate insolvency prediction model is still in development and so we welcome suggestions for improvements on the current methodology.

Keywords: corporate insolvency prediction, logistic regression, random forests, ada boost.

1 Introduction

We define corporate insolvency risk as the probability that a company will become insolvent in the next 12 months. Corporate insolvency risk is used, often in tandem with credit risk scores, to identify debtors who are at risk of becoming insolvent. Debt collection strategies can then be selected with the insolvency risk in mind. For example, an important debt collection strategy is early intervention to avoid an insolvent company increasing its debt, thus avoiding an increase in the eventual legal write-off of debt at insolvency.

The project described in this paper had a number of goals. The first was to test whether corporate insolvency prediction was possible using the available income tax return data. The second was to test the feasibility of a model designed to risk score across the full spectrum of companies (as opposed to constraining the target field to industry sector, for example). The third goal was to identify a preferred regression method after assessing logistic regression, ada boost, and random forests.

We should clarify goals that we consider are beyond the scope of this paper, although they are of interest for future work. First, we are not comparing the relative effectiveness of tax return data and financial statement data. Note that publically available financial statement data is only available for a tiny fraction of Australian companies, whereas this paper is focussed on all Australian companies that are registered in the tax system. Second, we are not comparing stratified models to a single all-company model. We intend to perform and describe such a comparison in future work. We also intend to test a multi-level model where both company-level and industry-level effects are jointly included.

Section 2 puts the current work in the context of a long history of insolvency prediction models and of recent work in Australia. Section 3 describes the data we obtained for building the model. In Section 4, we provide our particular predictive evaluation model-building methodology. All model performance evaluation is done on out-of-sample, future year test datasets. This means that not only are our test datasets distinct from the training datasets, they are also constrained to be test data from future years. Section 5 gives some descriptive data understanding results, then describes and evaluates the predictive models. Preliminary results on Financial Year 2006/2007 are then given. Section 6 discusses issues arising from the present work and possible future directions. We give our conclusions in Section 7.

2 Related Work

We have developed an empirical model of insolvency, as opposed to a structural model. An empirical model is data-driven and is built and assessed using predictive performance as the criterion, whereas structural models use an explicit function based on a theory of companies and insolvency. In the mid-1960s, Altman (Altman, 1993) developed the Z-Score model which uses 6 ratios and a linear discriminant model. There have been many variants since then by Altman and others. We use Altman's ratios, as well as a further 8 financial ratio variables defined by Ohlson (Ohlson, 1980), who used a logistic regression model.

Shin et al (Shin, 2006) compare ensemble models (bagging, boosting) with logistic regression, decision trees, neural networks and nearest neighbour. They also compare different feature selection methods. Their dataset

is restricted to 76 Turkish banks and their evaluation test data does not use future year sample datasets. They conclude that neural networks with appropriate feature selection is competitive with ensemble models and logistic regression on their Turkish bank dataset.

2.1 Recent Australian models

Jones and Hensher (Jones and Hensher, 2004; Hensher, Jones and Greene, 2007) have published a series of papers using new methods for models of predicting corporate insolvency. Their main focus is predictive performance at a group level, rather than at the individual company level. They note that predictive performance of models has not improved greatly since the 1960s (Hensher, Jones and Greene, 2007, p88). They observe that a type II error rate (where a solvent company is predicted to go insolvent) of 20% is typical for in-sample modelling results and even higher for out-of-sample future year results. In our project, a 20% type II error rate is acceptable as long as the results are still actionable to improve the operational efficiency of our business. One reason to support this contention is that many of the solvent companies in the 20% type II error category may not be technically or legally insolvent, but instead, may be financially distressed or even trading while insolvent. We have confirmed this hypothesis using surrogate variables for financial distress, such as level of indebtedness.

Similarly to Hensher and Jones, Hossari's PhD thesis (Hossari, 2006), focuses on improving the methodology in model building for predicting corporate collapse. Hossari uses multi-level models with financial ratio data extracted from audited financial statements. The data selected is a balanced sample, matched by industry classification, with fewer than 100 companies in the dataset. Model assessment is done on the single sample. Hossari found that vailable software for the multi-level models didn't scale well to larger datasets.

Moody's has an existing corporate default model for private companies, with 27K private companies in the dataset (Moody's, 2000; Moody's, 2000a). Our methodology mirrors Moody's *RiskCalc* approach. However, instead of audited financial statements we use income tax return fields as data sources for financial ratios. This allows us to score approximately one million private companies. Moody's early Australian model in 2000 achieved an Area under the ROC curve (AUC) (Fawcett, 2004) of 0.7, compared with an AUC score of 0.93, as generated by the most recent version of our model.

3 Data

Our population consists of active Australian companies, which we define as those companies that have had at least one income tax return since 2003. This covers about one million companies from all industry sectors, all size ranges and all corporation types, such as public and private.

3.1 Insolvency Target Variable

We obtain insolvency data from the Australian Securities and Investments Commission (ASIC 2007). This data is publicly available. There are at least seven different stages in the insolvency process, ranging from voluntary administration to liquidation. In our modelling process we use the widest possible definition of insolvency and so define a company to be insolvent if it enters any stage of insolvency, even if only temporarily, during a financial year.

Our principal interest is in predicting financial distress in general rather than insolvency specifically. However, the use of insolvency as the target variable has the advantage of definiteness and objectivity. Nonetheless, it is a broad target; there will many companies trading while insolvent that do not actually go into administration. This is consistent with David Hand's hypothesis that financial and customer modelling often involves ambiguous target concepts (Hand, 2006).

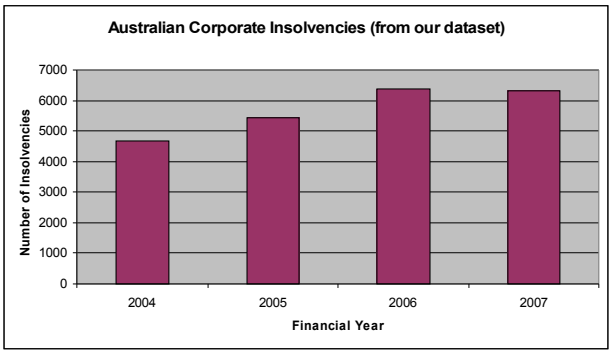


Figure 1: Insolvencies by Financial Year

3.2 Financial Ratio Variables

We adopt the financial ratio variables used by Altman (Altman, 1993) and Ohlson (Ohlson, 1990). Their financial data are obtained from audited financial statements provided by companies to the relevant corporate regulator (i.e. Securities Commission in the U.S.). Since we are scoring both public and private corporations, we need to exclude ratios using variables that apply only to public companies, such as *market value of equity*; data sources like those used by Altman and Ohlson are inadequate for our purposes. Instead, we have taken company income tax return data and adapted them for use as financial ratios.

Given that tax financial data are different from accounting financial statements, the question arose as to whether they would be suitable for accurate insolvency risk prediction. We shall see in the Results section that the results are roughly comparable with those using audited financial data. This is a useful and, as far as we know, novel finding.

The non-ratio financial variables used are:

1. Total Assets.
2. Net Income.

The two sets of financial ratio variabes that we have used are:

1. Altman variables (Altman, 1968):

i. Earnings before Tax and Interest / Total Assets.

ii. Net Sales / Total Assets.

iii. Market Value of Equity / Total Liabilities [No income tax return label equivalent of this is available, so we are unable to use it.]

iv. Working Capital / Total Assets.

v. Retained Earnings / Total Assets.

2. Ohlson Variables (Ohlson, 1980)

vi. TltoTA: Total Liabilities / Total Assets.

vii. CLtoCA: Current Liabilities / Current Assets.

viii. NItoTA: Net income / Total Assets.

ix. FFOtoTL: Pre-tax income plus depreciation and amortization costs / Total Liabilities.

x. INTWO: Flag that is 1 if cumulative net income over previous two years is positive.

xi. OENEG: Flag that is 1 if Owner's Equity is negative [Not available in income tax return data.]

xii. CHIN: Change in Income from previous year to current year.

xiii. TA: Size as ln (Total Assets/ GDP price growth] [This definition not available in income tax return]

xiv. Berry Ratio (Gross Profit / Operating Expenses)

3.3 Financial Distress Indicator Variables

Our modelling has also included input variables derived from the lodgment and payment behaviour of companies. Does a company lodge returns and pay taxes on time? If it is late, then how late is it? It is not surprising that issues such as these have proven sound indicators of financial corporate distress. Intuitively, a company at high risk of insolvency with cash flow problems or ongoing lack of profitability will be a poor debtor. However, there are counter-examples which must be managed; profitable companies with disorganised book-keeping may also be late lodgers and make late tax payments.

The precise definitions of these variables is not critical to the main thrust of this paper and so are not provided here.

3.4 Company Demographic Variables

Two company demographic variables that have been included are:

1. Age of company according to ASIC registration.
2. Industry classification using ABS industry codes.

The financial distress indicator variables, when added to the financial ratio variables, greatly improve our model's predictive performance. Company Demographic variables have a relatively minor effect relative to the other two input variable categories.

4 Predictive Model Methodology

We developed training and test datasets using the fundamental design principle that test data should be in the future relative to the training data. As mentioned in Section 2, this approach is not always used for model assessment, thereby bringing test results into question. For our business needs, real out-of-sample performance is what determines long-run utility of the model for the client and hence, client acceptance of the model (Moody's, 2000). Therefore, out-of-sample performance is the key assessment criterion for the insolvency risk model.

Our out-of-sample, future-year approach has been to train and test the model using data from consecutive financial years and then scoring a dataset derived from a later year. Table 1 shows the time frames for the extraction of training, testing and scoring datasets.

Dataset Type	Input Variable Year	Insolvency Target Year
Training	FY 2005 or before	FY 2006
Test	FY 2006 or before	FY 2007
Score	latest available data	not applicable

Table 1: Training/Test/Score Dataset Design

We use pair sampling (i.e. for every insolvent company, we find a solvent company), thus training and testing our model with balanced datasets. Pair sampling introduces a bias that causes an overestimate of variable significance in the model (Zmijewski 1984). This might be problematic if we were to interpret model parameters for explanatory purposes, but is less so in our current context of maximising predictive performance. As yet we have not tried matched pair sampling, where insolvent and solvent companies are compared based on size, industry classification and private/public status.

5 Results

5.1 Data Exploration

In order to check data quality and that the variable relationships are consistent with commercial practice, univariate plots of input variables versus the insolvency rate were produced. For example, the Berry Ratio (Gross Profit/Operating Expenses) is shown in Figure 2, where a value of zero (the wide x-axis value labelled '03:0-0.68') or a high value (the right-hand most x-axis value labelled '08:1.27-high') indicate the least risk of insolvency.

These observations are consistent with business knowledge. A Berry Ratio value of zero applies to companies with no operating profit and often with no operating expenses. Such companies, which include passive investment companies relying solely on investment income, carry little risk of becoming insolvent. A high Berry Ratio value indicates companies with large profit margins, where operating profits are much greater than expenses.

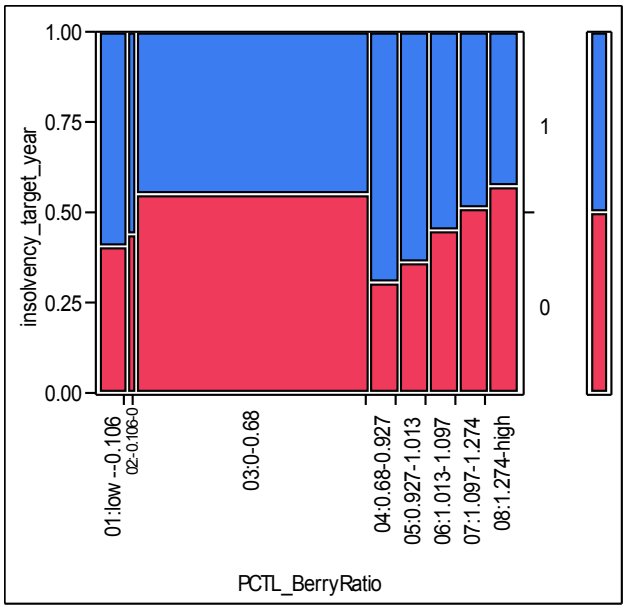
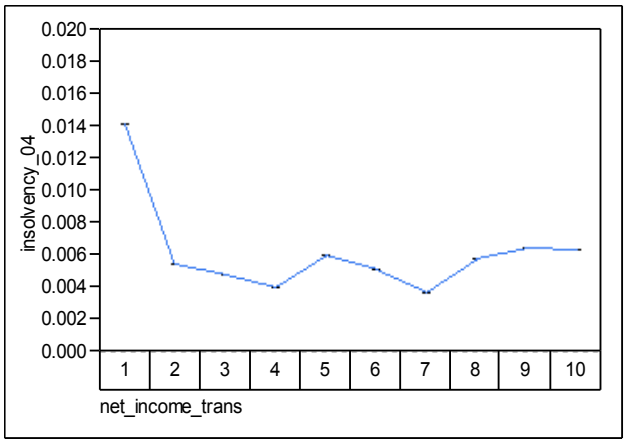


Figure 2: Discretised Berry Ratio vs Proportion of Insolvencies (for balanced sample). The lower part(red) of each category indicates of the proportion of solvent companies.

We performed another data quality check on the stability of the univariate relationships across Financial Years (FY). An example of this is shown for Net Income in Figure 3, which presents the rate of insolvency against net income deciles for two financial years (2004 and 2005). The overall rate of corporate insolvency approximates to 0.006 (roughly six in 1000 companies) for FY 2004 and 0.010 for FY 2005. The left-most net income category (labelled 1) is for negative income less than $-32K. The insolvency rate is highest for this category. While there is variance across the financial years, the insolvency rate pattern shares roughly the same trend. Note that the pattern of maximum and minimum values (at deciles 1 and 4) are fairly consistent across years.



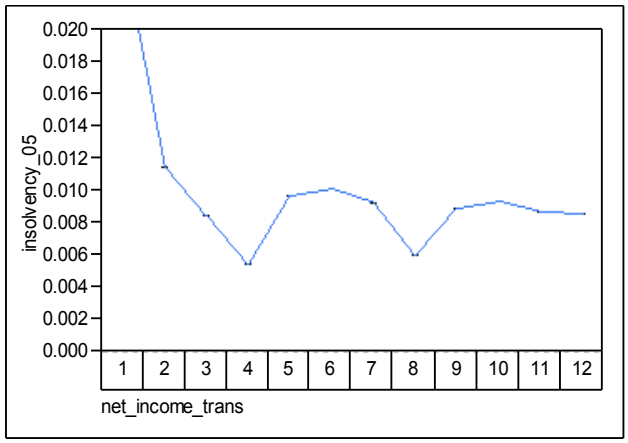
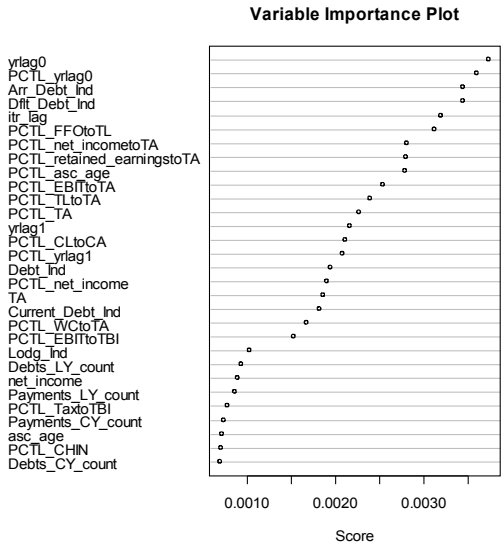


Figure 3: Net Income (quantised into deciles) vs. Insolvency Rate (for unbalanced sample) for Financial Years 2004 and 2005 . The y-axis gives the insolvency rate.

5.2 Variable Selection and Importance

Figure 4 plots variable importance based on the ada boost, while Figure 5 plots variable importance for the random forest model (showing two measures of importance). The variable importance measures used in these figures are defined in their respective R packages. They are based on the average % change in predictive accuracy when the variable is included and then excluded from the model.

There are significant differences in the variable importance rankings. The Ada Boost model flags its first four variables as being of much higher importance than the rest, while in Figure 5 (Random Forest model) these same variables appear in position 14 at the highest. This shows that variable importance ranks can be very model specific. It suggests that no single variable, operating alone, is highly predictive of insolvency and so rankings of importance are not definitive (Hand 2006). We also found that variable importance rankings are dataset sample specific. Resampling the training dataset and retraining the model leads to major changes in the variable ordering and minor changes in predictive performance.



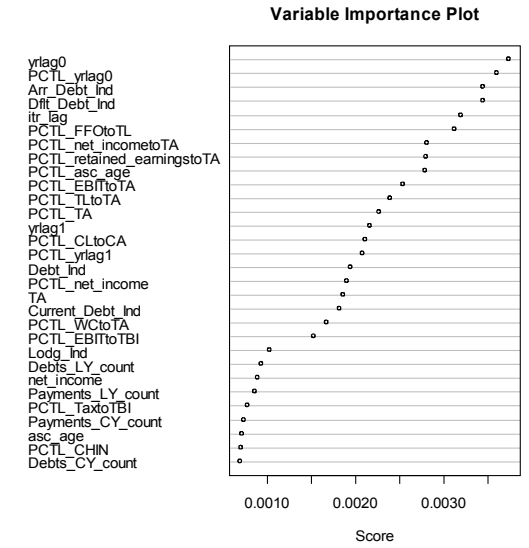


Figure 4: Variable Importance according to the Ada Boost model

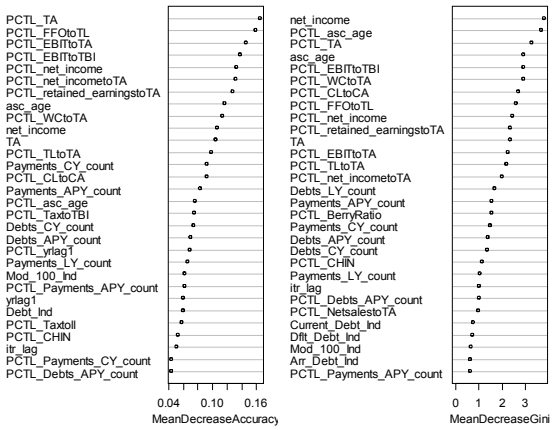


Figure 5: Variable Importance according to the Random Forest model

5.3 Model Building

Our production models are built using SAS Enteprise Miner 5.1. In our data preprocessing, we quantise interval variables into up to 10 quantiles. The quantisation of interval (continuous) variables helps prevent over-fitting by the regression model. It also helps with extreme values by allocating them to a single bin such as 'lowest quantile' or 'highest quantile'. Handling extreme values in this way improves the regression model's robustness, making it less sensitive to a particular data sample. We used a logistic regression model with variable selection. The optimisation target is validation misclassification cost and the cost ratio between insolvency and solvency is 50 to 1 (i.e. It is 50 times more beneficial to correctly identify an insolvent company than it is to correctly identify a solvent company). This cost was used because correct identification of insolvency is more important to our decision making than identifying true solvent companies.

SAS Enterprise Miner does not have the recent ensemble methods readily available. We were interested in benchmarking our SAS results with the results achieved using R and an R package, Rattle (Rattle, 2006), both of which are freely available open source software. A direct comparison could not be made because the R package is currently incapable of handling datasets that are of the scale of our company dataset (close to one million companies). Instead we sampled down to datasets of about 20K companies for both training and test datasets.

The Rattle classifier methods that we use include Random Forests, Ada Boost, SVM, Decision Tree (rpart) and Logistic Regression (glm). Note that we use classifiers to predict the probability of insolvency, normally a regression-like task. Classifiers that predict only a categorical class outcome rather than a probability are not applicable here. It should be noted that R's logistic regression package does not have variable selection (available in other R packages). It also does not have the ability to optimise cost using a validation dataset instead of the training dataset.

Classifier	(AUC)	AUC with transformed interval variables	(AUC) with transformed interval variables, variable selection
rpart	0.80	0.84	0.81
ada boost	0.88	0.88	0.89
rf	0.88	0.87	0.87
ksvm	0.84	0.88	0.88
glm	0.84	0.86	0.86

Table 4: Area under the ROC curve (AUC). It should be noted that there is a significant variance in the AUC estimates when the sampling of the test dataset is decreased from 1m to 20K. We intend to incorporate this source of variance into the model once we have computed it (our best guess is ± 0.03).

We have chosen to present our results using the Area Under Curve (AUC) measure derived from ROC curves. We present AUC results for each classifier on test data for a number of different samples:

1. sample without pre-processing
2. sample with interval variables quantised (following the SAS EM approach)
3. sample with quantisation and using variables only selected by SAS EM logistic regression.

The question that arises is: are the results returned by the various classifiers affected by pre-processing or by the variable selection step?

Figure 6 and Table 4 give the classifer results for the discretised interval variable dataset, using the variables as selected in the SAS EM model. The two ensemble methods (ada boost and random forests) are consistent

across the different data pre-processing steps. Logistic regression and SVM improve with the discretised interval variable dataset. As can be seen in both Figure 6 and, Table 4 the performance of the decision tree (rpart) is consistently lower than that of other models. This is as expected, given that, relative to the ensemble methods, decision trees generally have a high variance and low bias (Hastie, et al, 2001).

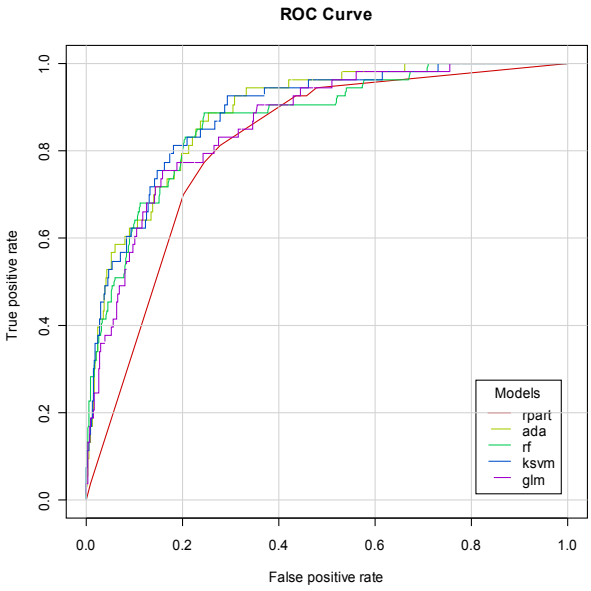


Figure 6: ROC Curves for classifiers estimating the probability of insolvency: rpart, ada, rf, ksvm and glm for the sampled test dataset with quantised interval variables and variable selection as done in SAS EM. Note these curves are based on a single test dataset sample and so we expect they will have relatively large confidence intervals on the curves (see Table 4 caption).

5.4 Results on Test Financial Year 2007

Figure 7 shows the predictive performance of the model when applied to Financial Year 2007. Note that this year has just ended so all of the 2007 insolvency data is not yet available. The trend across the quantiles (low risk on left, high risk on right) shows a general trend upwards as we would expect if the model were predictive.

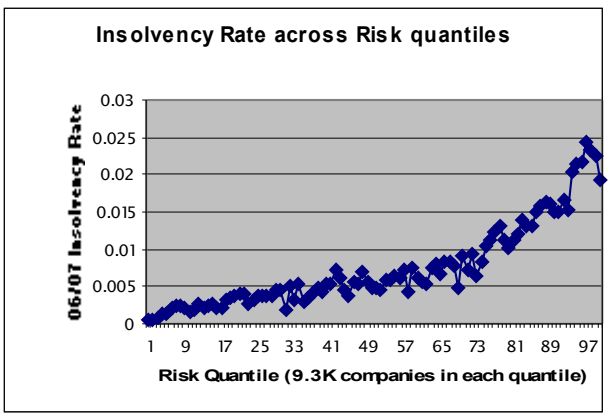


Figure 7: Result on FY 2007 test

The model places 15% of all insolvencies in the top 5% risk quantile, 27% of insolvencies in the top 10% risk quantile and 50% of insolvencies in the top 25% risk quantile. These results are comparable with those achieved by similarly large-scale commercial models making future year insolvency predictions. Personal bankruptcy prediction have even better results than company bankruptcy prediction, with 50% of bankruptcies being placed in the top 10% risk quantile (Experian, 2007).

6 Discussion

6.1 Stratification Models

We elected not to stratify the set of companies into sub-segments, despite the likelihood that it would improve model predictive performance, due to pragmatic, operational resource reasons. The first phase of the modelling process has been a proof-of-concept. Should the accuracy of the broad, single company type model prove sufficient, there will be no need to develop sub-segment models. We have briefly explored segmentation by:

1. public vs. private company
2. industry sector : Finance and Property sectors have very different financial ratio behaviour to Retail, Manufacturing and Construction.
3. size (as measured by total assets)

It is evident that predictive performance is improved by developing models for particular segments.

6.2 Related Entities

The first version of our model treats each company as independent of other companies. In reality, there are many types of corporate groups, involving interrelated companies. An extension of our model would identify related entities and include some form of risk score aggregation.

For small companies (<$100K assets), the credit risk (ability to pay debts on time) of the proprietor plays a significant role in the company's insolvency risk (ability to pay debts at all). In some cases, the proprietor's risk is as critical as the financial status of the company. For these smaller firms, the bankruptcy risk of business owners should be assessed and, where necessary, combined with the insolvency risk of the company entity (Moody's, 2000a). We plan to incorporate this relationship in future versions of our model.

6.3 Hazard Models

For this prototype, we have adopted a single insolvency period (one year) as a target. Some authors have postulated that hazard models, which utilise time-series data, are more accurate than static models (Shumway, 1999). However, in practice, hazard models have not been found to improve predictive accuracy significantly. Even so, with a view to optimising performance, we intend to extend our model to include some time-series data in future work.

7 Conclusion

We have built a corporate insolvency risk model for one million active Australian corporations using income tax return data and data from the Australian Securities and Investments Commission. The predictive performance of this model matches that achieved by commercial models whose scope is restricted to particular industries or public companies. Our data sources have been found to be suitable for corporate insolvency prediction and a single predictive model can be built for all corporations. The ensemble methods slightly outperform logistic regression at this stage (we do need to check test data variability issues). At this stage, we prefer logistic regression for its convenience of deployment as SQL in a data warehouse environment.

8 Acknowledgements

We thank Brian Irving, David Kuhl, and Stuart Hamilton for several helpful discussions. We thank Anthony Siouclis for his expert economist advice on the definition and use of tax label ratios. We also thank the referees for comments that improved the clarity of the paper.

9 References

Altman, E.I. (1968): Financial Ratios, Discriminant analysis and the prediction of corporate bankruptcy, Journal of Finance, **23**:4, pp589-609.

Altman, E.I; Haldeman, R.G; and Narayanan, P. (1977) Zeta Analysis: A new Model to Identify Bankruptcy Risk of Corporations. Journal of Banking and Finance, **1**, 9-24

Altman, E.I. (1993): Corporate Financial Distress and Bankruptcy: A Complete Guide to Predicting and Avoiding Distress. New York: Wiley

ASIC (2007): Australian Securities and Investments Commission, http://www.asic.gov.au/.

Baxter, R.A. (2006) Finding Robust Models Using a Stratified Design, AI2006: Advances in Artificial Intelligence, Lecture Notes in Computer Science**, 4304**, pp 1064-1068, Springer.

Experian (2007) Harben, S. and Curtis,C., Modelling Personal Bankruptcy in the UK, White Paper, Experian-Scorex, http://www.experian-scorex.com/.

Fawcett, T. (2004). ROC Graphs: Notes and Practical Considerations for Researchers. Technical report, Palo Alto, USA: HP Laboratories.

Hand, D. (2006) Classifier Technology and the Illusion of Progress. Statistical Science **21**(1). pp 1-15.

Hastie, T., Tibshirani, R., and Friedman, J. (2001) The Elements of Statistical Learning, Springer.

Hensher, D.A., Jones, S. and Greene, W.H. (2007): An Error Component Logit Analysis of Corporate Bankruptcy and Insolvency Risk in Australia.

Hossari, G. (2006): A Ratio-Based Multi-Level Modelling Approach for Signalling Corporate Collapse: A Study of Australian Corporations. PhD Thesis, Swinburne University of Technology.

Jones, S. and Hensher, D.A. (2004) Predicting Firm Financial Distress: A Mixed Logit Model. The Accounting Review, **79**(4), pp. 1011-1038.

Kaski, S. (2001) Bankruptcy Analysis with Self-Organizing Maps in Learning Metrics. IEEE Transactions on Neural Networks, **12**:4, 2001.

Keasy, K; and Watson, R. (1991): Financial Distress Prediction Models: A Review of their Usefulness. British Journal of Management, **2**, 89-102.

Lin, L; and Piesse, J. (2004): The Identification of Corporate Distress in UK Industrials: A Conditional Probability Analysis Approach. Research Paper 024 The Management Research Papers. Kings College London. University of London.

Moody's (2000): RiskCalc For Private Companies: Moody's Default Model. Rating Methodology. May 2000, Moody's Investor Service.

Moody's (2000a): RiskCalc For Private Companies II: More Results and The Australian Model. Dec. 2000, Moody's Investor Service.

Ohlson, J.S. (1980): Financial Ratios and the Probabilistic Prediction of Bankruptcy, Journal of Accounting Research, **19**, pp109-31.

Rattle (2006). Rattle Software, An R Package, http://rattle.togaware.com/, R software, http://r-project.org/.

Shumway, T. (1999): Forecasting Bankruptcy More Accurately: A Simple Hazard Model.

Shin, S.W., Lee, K.C. and Kilic, S.B. (2006) Ensemble Prediction of Commercial Bank Failure Through Diversification of Bank Features, AI2006: Advances in Artificial Intelligence, Lecture Notes in Computer Science**, 4304**, pp887-896, Springer.

Sung, T.K., Chang, N. and Lee, G. (1999) Dynamics of Modeling in Data Mining: Interpretative Approach to Bankruptcy Prediction. Journal of Management Information Systems, 16:1, pp. 63-86.

Wilson, R.L; and Sharda, R. (1994): Bankruptcy Prediction using Neural Networks. Decision Support Systems, **11**, 545-557

Zmijewski, M. (1984) Methodological Issues Related to the Estimation of Financial Distress Prediction Models. Journal of Accounting Research, **22**, 59-62.

CreditRisk: Credit Risk Website, http://www.creditrisk.com/. Accessed 29 Jun 2007.

Establishing a Lineage for Medical Knowledge Discovery

Anna Shillabeer[1,2] **and John F. Roddick**[1]

[1] School of Informatics and Engineering,
Flinders University,
PO Box 2100, Adelaide,
South Australia 5001
Email: {anna.shillabeer, roddick}@infoeng.flinders.edu.au

[2] Heinz School Australia
Carnegie Mellon University
Torrens Building
220 Victoria Square
Adelaide SA 5000

Abstract

Medical science has a long history characterised by incidents of extraordinary insights that have resulted in a paradigm shift in the methodologies and approaches used and have moved the discipline forward. While knowledge discovery has much to offer medicine, it cannot be done in ignorance of either this history or the norms of modern medical investigation. This paper explores the lineage of medical knowledge acquisition and discusses the adverse perceptions that data mining techniques will have to surmount to gain acceptance.

Keywords: Medical and Health Data Mining.

1 Introduction

The nature of data mining research is that it requires a second discipline to be validated as useful. To do this, data mining must adapt to this second discipline and conform to the norms expected of that discipline. In contrast to many disciplines where data mining has been applied, medicine has a strong, established and accepted research methodology and application of data mining technology that falls outside this, however well-meaning, will struggle to be taken seriously.

This paper thus explores the history of knowledge acquisition in medicine and extracts from this history some important issues that data miners should take into account when mining medical data.

The paper is organised as follows. The next section explores the history of knowledge acquisition methodologies in medicine, while Section 3 discusses the role of intuition and serendipity in many medical advances. Section 4 then briefly discusses the role of data mining to the medical context and includes some examples of where data mining techniques have implicitly been used. Section 5 then outlines a number of arguments that have been raised against the adoption of data mining in medicine and briefly discusses each in turn.

2 Knowledge Acquisition in Medicine

Throughout recorded history there has been debate over what constitutes knowledge and therefore what constitutes proof of knowledge. Early practitioners of medical science, such as Hippocrates, based their knowledge development in philosophy and their ability *to see with the eye of the mind what was hidden from their eyes* (Hanson 2006). By the first century A.D. physicians such as Galen were beginning to question the validity and contradictions of Hippocrates' work which had stood mostly unopposed since the 5th Century B.C. It is not clear if there were any agreement or understanding of the methods applied by physicians to develop their knowledge base at this time as there was no empirical proof or scientific process documented. Galen was one of the first to suggest that there should be a process for the provision of substantiated evidence to convince people of the value of long held medical beliefs. He raised the notion of a practical clinical method of knowledge acquisition which combined the Hippocratic concept of hypothesis development through considered thought and a priori knowledge, with clinical observation to evaluate and hence provide proof or otherwise of the hypothesis. This general process has survived to the present day and is reflected not only in the provision and acceptance of new knowledge but also in clinical diagnosis.

The historical debate on knowledge acquisition methodologies has primarily focused on three philosophical groups; Methodists, Empiricists and Rationalists. Whilst these three groups are most frequently discussed in a Graeco-Roman context, they were either being applied or paralleled in various other cultural contexts including India and Islam. All three of these contexts are discussed here briefly to demonstrate the extent and foundations of medical knowledge acquisition debate in the ancient world.

2.1 The Graeco-Roman Context

- Methodists

  The first prominent physician practising the Methodist philosophy was Hippocrates of Cos (460-380 B.C.) – the so-called *Father of Medicine* (Hanson 2006). It is believed by many that he initiated the production of over 60 medical treatises known as the Hippocratic Corpus. The corpus was written over a period of 200 years and hence had more than one author which is reflected in the sometimes contradictory material which it contains. The body of work was, how-

ever, consistent in its reliance on defining a natural basis for the treatment of illnesses without the incorporation or attribution of magic or other spiritual or supernatural means as had occurred previously. Methodists founded their knowledge on an understanding of the nature of bodily fluids and developing methods for the restoration of fluid levels. They were not concerned with the cause of the imbalance or the effect on the body of the imbalance, only in recognising whether it was an excess or lack of fluid and the method for treating that observation.

- Rationalists

  Rationalists believed that to understand the workings of the human body it was necessary to understand the mechanism of illness in terms of where and how it affected the body's functioning (Brieger 1977). They were not interested in the treatment or diagnosis of illness but focused on understanding and recording the functioning of the living system. Two works are prominent in this group (Cosans 1997); the *Timaeus* by Plato, which systematically described the anatomical organisation of the human body and; *Historia animalium* by Aristotle, which discussed further both human and animal anatomy and the links between such entities as the heart and blood circulation. This method of knowledge acquisition was criticised as it effectively removed medicine from the grasp of the average man and moved it into a more knowledge based field where philosophical debate or an observational experiential approach were not deemed sufficient (Brieger 1977). Essentially, rationalists did not believe in a theory unless it was accompanied by reason. They espoused the requirement for knowledge to be founded on understanding both the cause and effect of physical change in the body (Horton 2000).

- Empiricists

  The Empiricists believed that it was not sufficient to understand how the body works and reacts to illness. They pursued a philosophy which stated that it was necessary to demonstrate the efficacy of treatments and provide proof that a treatment is directly responsible for the recovery of a patient rather than providing academic argument regarding why it *should* result in recovery. Galen is considered to be one of the earliest, better known and more frequently quoted empiricists (Brieger 1977). He was particularly interested in testing the theories proposed in the Hippocratic Corpus, especially given its frequent contradictions. His work was also produced when medicine as a science was evolving from its previous status as a branch of philosophy. In his work Galen argues that *medicine, understood correctly, can have the same epistemological certainty, linguistic clarity, and intellectual status that philosophy enjoyed* (Pearcy 1985). Empiricists were the first to concentrate on the acquisition of knowledge through demonstrated clinical proof developed through scientific methodologies which provided conclusive statements of cause and effect.

2.2 The Islamic Context

- The Methodists (Ashab al-Hiyal).

  In direct parallel to the Methodist philosophy discussed earlier, this sect believed in a generalist view of illness and treatment and categorised conditions in terms of the extent to which bodily fluids and wastes are either retained and/or expelled. Treatments were generally natural remedies based upon adjusting the balance between such aspects of life as food and drink, rest and activity etc. Methodists were not interested in the type of patient or cause and effect of illness and were hence considered to be more prone to error (Muhaqqiq n.d.).

- The Empiricists (Ashab al-Tajarib).

  Islamic empiricists believed that medical knowledge was derived from experience obtained through the use of the senses and that the knowledge is comprised of four types (Muhaqqiq n.d.):

  - Incident (*ittifaq*) – this can either describe a natural event, such as a sweat or headache, or an accidental event, such as a cut or a broken limb.
  - Intention (*iradah*) – denotes an event experienced by choice, such as taking a cool bath to reduce a fever.
  - Comparison (*tashbih*) – a technique employed by a practitioner whereby it is noted that a technique results in a useful effect which can be applied to other similar presentations. For example, applying cold water to reduce localised burning of the skin following the observation that a cool bath can reduce generalised fever or body heat.
  - The adoption of a treatment that was used in another similar case (*naql min shay' iki shabihihi*) – a technique whereby the physician applies a treatment for a similar presentation for a presentation which has not been encountered before. For example, the prescribing of a medication for a previously unencountered infected tooth where that medication had only previously been used for an infection elsewhere in the body.

  The empiricists treated a patient through knowledge of the patient's demographics, and therefore all patients of a certain age and sex with some similar complaint were treated the same whereas patients of the opposite sex might have been treated differently though the condition was the same. Their knowledge was based on patient characteristics rather than a specific condition or set of symptoms. Whilst this seems to differ from the Graeco-Roman definition of empiricism, both groups believed that knowledge acquisition occurred through observing or testing the effect of a treatment and producing rules based on what is considered reliable empirical proof rather than conjecture and debate.

- The Dogmatists (Ashab al-Qiyas).

  The Dogmatists believed that scientific belief and knowledge should be derived from experience and observation, tempered by considered evaluation (Mohaghegh 1988). They believed that changes in the bodily functions must be precipitated by some event and that it is necessary to not only understand what these changes are but also what the specific causes of those changes are to correctly diagnose and treat any condition. They define changes as being of two types (Muhaqqiq n.d.):

  - *Necessary change* - drink reducing thirst. This is a change which is required for normal bodily functioning.

– *Unnecessary change* - dog bite causing bleeding. This change is not a requirement to aid or enhance bodily well-being.

Dogmatists based their treatments upon the nature of the condition rather than the type of patient as seen with the empiricists. The treatments were therefore selected through knowledge of the causes of illness and the effects of those treatments upon the illness or symptoms. This required an understanding of the physical body and the changes that result from illness in a similar manner to the Graeco-Roman Rationalists.

It has been suggested that in general, Islamic physicians relied primarily upon analogy which reflects their focus on logic in other scholarly areas (Mohaghegh 1988). This has resulted in widespread support for the Dogmatist methods of knowledge acquisition through research and understanding of cause and effect in the human system. However there is still debate between scholars with some believing that Dogmatism alone is the only method of ensuring progress in medical diagnosis and treatment as it is the only method which tries to seek new understanding rather than relying upon past experience or a closed assumption that there is a single cause for all illness (Mohaghegh 1988). Others prefer to adhere to the Graeco-Roman perspective (developed by Plato) that a combination of experience and analogy is required if a holistic, 'correct' practice of medicine is to be achieved (Muhaqqiq n.d.).

2.3 The Indian Context

India is not well known for its scientific contributions or texts, however it has a long history in the development of a quorum of medical knowledge. In the 11th century a Spanish scholar, Said Al-Andalusi, stated that he believed that the Indian people *are the most learned in the science of medicine and thoroughly informed about the properties of drugs, the nature of composite elements and the peculiarities of the existing things* (al Andalusi 1991). The reasons for this apparent invisibility of Indian scientific progress may be due to religious debate in India which has frequently negated the influence of scientific explanation instead preferring to rely upon mystical or spiritual beliefs. There are however, documented scientific approaches to the development of a body of knowledge regarding medicine from centuries before the texts of Hippocrates and which, although often earlier, discuss similar theories to those presented in the Graeco-Roman texts.

- The Rationalist schools

  One of the earliest groups to produce texts concerning to acquisition of knowledge regarding the human state was the Upanishads which were believed to have been written between 1500 and 600 B.C.E. and were concerned with knowledge regarding the spirit, soul and god (South Asian History Project 2002, Kaul & Thadani 2000). Although these texts were embedded in mysticism and spirituality, they used natural analogy to explain the notion of the soul and god and allowed the expression of scientific and mathematical thought and argument. This formed the basis for the emergence of the rationalist period. Early rationalists included the Lokyata, Vaisheshika and Nyaya schools. These groups espoused a scientific basis for human existence and a non-mystical relationship between the human body and mind. They also developed primitive scientific methodologies to provide *valid knowledge* (South Asian History Project 2002, Kaul & Thadani 2000)

  – The Lokyata were widely maligned by Buddhist and Hindu evangelicals as being heretics and unbelievers due to their refusal to *make artificial distinctions between body and soul* (Kaul & Thadani 2000). They saw all things in terms of their physical properties and reactions and gave little attention to metaphysical or philosophical argument, preferring to believe only what could be seen and understood. They developed a detailed understanding of chemistry, chemical interactions and relationships between entities. They are also believed to be the first group to document the properties of plants and their uses, this provided an elementary foundation for all pharmaceutical knowledge which followed.
  – The Vaisheshika school's main achievement in the progression of human knowledge was in their development of a process for the classification of entities in the natural world, and in their hypothesis that all matter is composed of vary small particles with differing characteristics. (South Asian History Project 2002). Their theory stated that particles, when combined, gave rise to the wide variety of compounds found upon the earth and allowed them to be classified by the particles from which they were formed. This school also introduced the notion of cause and effect through monitoring and understanding temporal changes in entities. The importance of this work lay in the application of a methodology for identification and classification of relationships between previously unconnected entities. This early recognition of the need for a documented scientific process provided a mechanism for the schools which followed to present substantiated proof of evidence for theories in the sciences including physics, chemistry and medicine.
  – The Nyaya school further developed the work of the Vaisheshika school by continuing to document and elaborate a process for acquiring valid scientific knowledge and determining what is true. They documented a methodology consisting of four steps (South Asian History Project 2002):
    * *Uddesa* was a process of defining a hypothesis.
    * *Laksan* was the determination of required facts *through perception, inference or deduction.*
    * *Pariksa* detailed the scientific examination of facts.
    * *Nirnaya* was the final step which involved verification of the facts.

  This process would result in a conclusive finding which would either support or refute the original hypothesis.

  The Nyaya school also developed definitions for three non scientific pursuits or arguments which were contrary to the determination of scientific truth but which were often applied to provide apparent evidence for theories or knowledge (South Asian History Project 2002, Kaul

& Thadani 2000). These included *jalpa* to describe an argument which contained exaggerated or rhetorical statements or truths aimed at proving a point rather than seeking evidence for or against a point; *vitanda* which aimed to lower the credibility of another person and their theories and generally composed of specious arguments; and *chal*, the use of language to confuse or divert the argument.

Further to this again a set of five 'logical fallacies' was developed:

- *Savyabhichara* - denotes the situation where a single conclusion is drawn where there could be several possible conclusions,
- *Viruddha* - where contradictory reasoning was applied to produce proof of the hypothesis,
- *Kalatita* - where the result was not presented in a timely manner and could therefore be invalidated,
- *Sadhyasama* - where proof of a hypothesis was based upon the application of another unproven theory, and
- *Prakaranasama* - where the process simply leads to a restating of the question.

These concepts were unique in their time and many remain applicable in modern scientific research.

- The Jains are worthy of note not because of the size of their impact on the process of acquisition of scientific knowledge but due to their identification of a truth matrix which demonstrated that there are more possible outcomes from scientific research than simply true or false as shown in Table 1.

	Proved	Indeterminate
True	⋆	
False	⋆	
True or false	⋆	
Indeterminate		⋆
True or indeterminate		⋆
False or indeterminate		⋆
True or false or indeterminate		⋆

Table 1: The 7 states of truth according to the Jains

Prior to the work of the Jains, scientists described their outcomes only in terms of true or false and did not consider that there may be degrees of truth or that a hypothesis might not have been proved or disproved but may remain open to debate or require further testing to gain a conclusive result.

This section has demonstrated that the quest for new medical knowledge and a deeper understanding of the human system is not a recent initiative but one which has its foundations up to four centuries B.C. While there were several distinct cultural groups all were primarily concerned with defining the most reliable methodology for evaluating what knowledge could be trusted and applied clinically. The Graeco-Roman and Islamic practitioners were concerned with the means by which evidence was obtained and the Indians were more concerned with methods for proving the validity of knowledge after it had been discovered. Both foci remain the topic of debate and as late as 1997 a report was published by the *International Humanist and Ethical Union* regarding trusted versus untrusted clinical practices and the requirement for proof of the benefits of medical treatments. The opening of a Mantra Healing Centre at the Maulana Azad Medical College in New Delhi was described as *ridiculing the spirit of inquiry and science* through its application of *sorcery and superstition in their rudest form* (Gopal 1997). The report did not however argue that there was no worth in mantra healing but that there was no proof of worth as per the requirements of the still flourishing rationalist opinion. The debate on what is trusted and clinically applicable knowledge rightly informs the focus of much research.

3 Non-scientific knowledge acquisition

History has shown that the acquisition of much currently accepted medical knowledge was based on serendipity or chance accompanied by a strong personal belief in an unproven hypothesis. Indeed, much knowledge was acquired through a process which directly contradicts accepted scientific practice. Whilst there was usually a scientific basis to the subsequent development of proof, this was often produced through a non traditional and often untrusted application of scientific processes. Unfortunately, this often resulted in lengthy delays in acceptance of the work. The following list provides a range of such breakthroughs over the past 250 years which can be attributed to chance, tenaciousness and/or the application of non-conventional methods to obtain evidence.

- James Lind (1716-1794) (Buck et al. 1988). Based upon an unsubstantiated personal belief that diet played a role in the development of scurvy on naval vessels, Lind performed limited randomised trials to provide proof and then published his *Treatise on the Scurvy*, which is still relevant to this day.
- Edward Jenner (1749-1823) (Sprang 2002). During his apprenticeship, Jenner overheard a milkmaid suggest that those who have had cowpox did not contract smallpox. He then tested the theory by infecting a young boy sequentially with each pathogen and as a result created the concept of a vaccine and initiated the global eradication of smallpox.
- John Snow (1813-1854) (Burke 1985). Snow believed, without any direct evidence, that the transmission of viral agents was possible through contaminated water. In 1854 he applied the theory and provided an answer to the cholera epidemic.
- Alexander Fleming (1881-1955) (Mulcahy 1996). Fleming stumbled upon a discarded culture plate containing a mould which was demonstrated to destroy staphylococcus. The mould was isolated and became the active ingredient in penicillin based antibiotics.
- Carlos J. Finlay (1833-1915) (Adams 1992). Finlay's observations regarding cholera, although similar to Snow's, were not taken seriously because of a perceived criticism of the local authorities. His observations regarding the mosquito as a vector in the transmission of Yellow Fever were also nearly dismissed and it was 20 years before his theory was taken seriously.
- Henri Laborit (1914-1995) (Pollard 2006). During his ward visits, Laborit noticed that patients given an antihistamine named promethazine to

treat shock not only slept but reported pain relief and displayed a calm and relaxed disposition leading to the development of medications to treat mental disorders including schizophrenia.

- Robert Edwards and Patrick Steptoe (1925-, 1913-1988) (Fauser & Edwards 2005). These doctors were the first men to deliver a baby through in-vitro fertilisation after 20 failed attempts and great ethical debate following a lack of proof in animal subjects.

- Barry J. Marshall (1951-) (Marshall 1998). Marshall worked against accepted medical knowledge to provide proof of the bacterial agent, Heliobacter Pylori, as the cause of stomach and duodenal ulcers. So strong was the opposition to initial clinical testing of the theory he resorted to using himself as the test subject.

Whilst each of these examples provided wide reaching benefits to human health and contributed significantly to the body of medical knowledge in some cases, they would not have been possible if only standardised scientific methodologies had been applied using only trusted traditional processes. This demonstrates that there is often a need to depart from conventional methodologies to facilitate the acquisition of knowledge, although there is always a requirement to subsequently provide substantiated proof and an argument based upon accepted scientific principles.

The applicability of this notion of departing from conventional methodologies is particularly relevant to data mining research with its focus on the application of new techniques and technologies which already have demonstrated an ability to provide an important impetus to the acquisition of knowledge in other domains. However, the same proof of hypothesis hurdles must be overcome and an equally strong argument and testing methodology must be provided for the resulting knowledge to be accepted. Throughout history the same quality of evidence has been required and the omission of this evidence has often resulted in decades of latency between hypothesis statement and the generation of conclusive evidence in support (or otherwise) of that hypothesis.

Despite the methodology for producing the evidence required for knowledge acquisition, the above examples all fulfilled a number of basic requirements prior to acceptance. These requirements are summarised below:

1. Replication of results. For data mining, this means that datasets must be, at least in theory, publicly available. Moreover, as Freitas (2000) points out, association rule mining generates the same set of rules for every subsequent run over the same data, whereas some classifiers can be unstable, generating markedly different classifiers for small changes to the input dataset.

2. Non contradictory results. There are many data mining algorithms that will produce results that include an apparent contradiction but few that attempt to detect these contradictions and explain them.

3. Scientifically justified theories and hypotheses.

4. Ethical methodologies and measures.

5. Results demonstrated to be representative of the population. This means that accepted methods of statistical confidence must be adopted.

6. Results derived from sufficient numbers of cases. Data mining works best over large quantities of data. Running data mining over small datasets is unsound and, arguably, not within the scope of data mining technologies.

7. Publicly documented processes and results.

Some of the impediments to the adoption of data mining in medicine are discussed in Section 5 and demonstrate the need to understand and apply these requirements.

4 Medical knowledge acquisition through data mining

One of the more commonly quoted definitions of data mining is that it is a *non-trivial process of identifying valid, novel, potentially useful and ultimately understandable patterns in data* (Houston et al. 1999). Data mining is essentially part of a larger knowledge discovery process whereby data is transformed into information and knowledge is then extracted from that information with the focus being on the identification of understandable patterns in data. This definition is used as a foundation for the development of many medical data mining systems. However, data mining has the potential in the medical domain to expand this definition even though use to date has been sparse (Forsstrom & Rigby 1998).

Data mining in medicine is most often used to complement and expand the work of the clinician and researcher by qualifying or expanding knowledge rather than providing new knowledge, as has been the trend in other domains. Very little health data mining is purely exploratory and hence it is generally not applied to provide novel knowledge, that is, to identify new patterns hidden within the data. One of the difficulties in providing new knowledge in the health domain is the need to sufficiently cross reference and validate the results. It is not sufficient to provide a standard rule in the form of A gives B in the presence of C without substantiating the information held therein. This information could already be known, may be contrary to known medical facts due to missing attributes leading to incomplete information, may not be statistically valid by trusted measures or may simply not relate to the specialisation of the user and is therefore irrelevant. Hence data mining systems to date have generally been developed for a specific user or data type and for a specific purpose.

The health domain is complex and standardised data mining techniques are often not applicable (Cios & Moore 2002), however, there are four procedures that are frequently documented.

1. Production of association rules;

2. Clustering;

3. Trend or temporal pattern analysis; and

4. Classification.

These processes are applied to provide three generalised functions:

1. Prediction;

2. Expert decision support;

3. Diagnosis.

It should be noted that over centuries medical professionals have (often unknowingly) employed the

same scientific analytical methods to data as are applied during data mining in order to develop hypotheses or to validate beliefs. Whilst these techniques have been applied in a simplistic form they clearly demonstrate the applicability of the founding principles of data mining to medical inquiry and knowledge acquisition.

- Data sampling - James Lind (Katch 1997).

  Lind performed small randomised trials to provide proof of the cause of scurvy. In his position as Naval doctor he could test his theories on the crews of the vessels he sailed on, however without documented proof it was not possible to test the entire navy on mass. Developing sufficient proof in this manner was a lengthy process and it was 50 years before the British Admiralty accepted and applied his theories, a delay which cost the lives of many sailors. Lind's process shows the use of examining subsets of the population, being able to clearly identify the variant in the knowledge gained and then substantiating that knowledge by testing on similar populations to ensure the finding is representative is a suitable technique for hypothesis testing and knowledge substantiation.

- Association mining - Edward Jenner (Sprang 2002).

  Following development of a hypothesis from the knowledge that milkmaids were less likely than members of the general population to develop smallpox due to their increased contact with cow pox, Jenner conducted further tests over a period of 25 years to validate the relationship and publish his findings. This work demonstrates the use of the concept of support through Jenner's realisation that there was a frequently occurring and previously unknown pattern in a data set or population. That pattern was subsequently tested to provide confidence levels by showing that contracting cowpox almost always results in an inability to contract smallpox.

- Clustering - John Snow (Burke 1985).

  In his investigation of the cholera outbreak of 1854, Snow applied a meticulous process of interviewing to collect data. He used the information collected to develop a statistical map which clustered interview responses based upon the water pump which supplied water to the individual. This revealed that every victim had used a single supply of water and no non-sufferers had used that supply. Further investigation showed that this pump was contaminated by a nearby cracked sewage pipe. This shows not only the power of the use of medical data for statistical purposes, but the benefits that can result from applying clustering techniques to that data.

- Association rules and classification - Henri Laboit (Pollard 2006).

  Laboit extended the use of promethazine to treat mental disorders including schizophrenia by realising patterns in side effects from administering the drug during surgery on non mentally ill patients. This was achieved through identifying association rule style patterns to describe associations between focal and non focal attributes, for example, combinations of relationships between diagnosis, treatment, symptoms, side effects and medications. Analytical techniques were employed to classify conditions exhibiting similar patterns of presentation and clinical testing was utilised to demonstrate the effect of applying an identified drug to control those classes of symptoms.

These techniques until recently were employed manually and hence were on a much smaller scale than we see today through the application of automated data mining systems. However they demonstrate the impressive potential for automated data analysis techniques to be applied with greater benefits and applicability than previously thought possible.

5 Perceived Impediments to the Adoption of Data Mining in Medicine

The history of medical knowledge acquisition can be seen to inform some of the criticisms of medical data mining which have led, in some cases, to the technology being overlooked as a tool. In particular, the (initially at least) exploratory nature of data mining seems to negate the need for a hypothesis. Such criticisms must be addressed and this section discusses six of the strongest arguments cited against the application of data mining in medicine.

Data mining outcomes are seen as generalisations and not verified for medical validity or accuracy. (Elwood & Burton 2004, Milloy 1995)

Medicine is a highly complex domain for which data mining processes were not designed. Frequently they originated in response to changes in commerce or management practices where there was no methodological need to substantiate results on the basis of protocols or domain knowledge. Medicine has requirements which are outside the original scope of the technology, and to be applicable to a science which is concerned with critical decision making there is a need to modify the technology to reflect this different environment.

Whilst this is a serious issue, it is often borne from misrepresentation of the results of data mining rather than from the process itself. There is a need for careful consideration of the language used when reporting results (Maindonald 1998). It is possible for the results to be specific but for the language of reporting to generalise the message. For example, Elwood & Burton (2004) describe a case where a mining outcome showed that smoking does not have a direct link with skin cancer. However the resulting media story reported that smoking is not linked with cancer generally. While a scientific data mining process was applied the language of information presentation was misleading and the resultant reporting was inaccurate and medically invalid. Medicine is especially sensitive to this form of information distortion and the consequences have the potential to be life threatening, politically sensitive, costly and persistent which is rarely the case in other domains.

There is little to sustain this argument in light of recent work in the field. By the application of suitable statistical methods, evaluation of all results and applying industry accepted standards there is no reason to believe that data mining cannot provide effective validation and accuracy checking processes (Gebski & Keech 2003, Shillabeer & Roddick 2006). Three steps have been suggested to safeguard against this criticism (Smith & Ebrahim 2002).

1. Results should not be published on the basis of correlation alone.
2. An explanation should be provided with the results to provide clarification e.g. A definition of the unique quality of the allergen that triggers the alleged immune response.
3. Results should be replicated, confirmed and documented prior to publication.

These steps are not part of standard data mining methodologies but are required to be undertaken if the mining of medical data is to overcome criticism, be viewed as 'good science' and gain trust in the medical community.

Associations are not representative of other similar attributes and do not consider other potential contributors. (Milloy 1995, Smith & Ebrahim 2002)

In a medical context, relationships found between one allergen and symptoms must be substantiated through analysis of similar allergens or the same allergen in other temporal, spacial or demographic instances. If this cannot be shown it suggests that there is not a conclusive argument for cause and effect or that some other catalyst or cause has been missed (Smith & Ebrahim 2002). Again, data mining was not designed to do this however this should not be a preventative. Methods are available to achieve this where it is important to determine the semantic closeness of results. Criticism often focuses on data dredgers who promote results as facts rather than being indicative of a possible scenario requiring further investigation. Where an association is found it is important to compare this with other associations or to apply a clustering algorithm to group semantically and determine where there is similarity or otherwise to other attributes or rules.

***P*-values are set arbitrarily and therefore the results cannot be trusted.** (Milloy 1995, Smith & Ebrahim 2002)

P-values may be applied in two ways: to evaluate and discriminate the acceptability of mining results and, as a guideline or tool for reducing irrelevant outcomes. Data mining can also be applied in divergent modes; to show what the common patterns in data are, or to show where common patterns are refuted in the data. Obviously the p-values required for these two mining runs would be different thus obviating the need for a range of p values to be applied. An issue arises where the p-value is modified iteratively until the outcomes meet some predefined need. It is important to always set heuristic thresholds in context of the specific analysis being done and in fact a calculation for p should be only one test among many (Gebski & Keech 2003, Shillabeer & Roddick 2006). In the medical domain, attribute-value relationships which occur frequently, and hence have a low p-value, are likely to be known already and would generally be of little if any interest. This is a major difference between traditional data mining applications, where generally the events which occur most frequently are of the greatest interest and hence have a similar p threshold, and applications in the medical domain where frequency is not a conclusive determinant in defining the usefulness, validity or applicability of results and hence may require varying p values.

Associations between attributes are dependent upon the data set being analysed and are not representative. (Smith & Ebrahim 2002)

There is often a poor approach to the collection and description of data sources and samples which is not consistent with the process of data mining and other scientific methodologies (Milloy 1995, Maindonald 1998). For results to be accepted the data source should be from an identifiable population with defined characteristics such as location, demographics, and proportions (Smith & Ebrahim 2002). In a clinical research setting this is overcome using protocols and guidelines to ensure that results are representative and can be replicated. One such protocol is CONSORT which is used globally by medical researchers and is endorsed by a number of prominent journals (Moher 1998).

In epidemiological studies, this problem is exacerbated by the use of external, non medical domain specific datasets that, while generally accurate and reliable, where not collected for the purposes of data mining (Roddick et al. 2003).

Data mining provides validation through tools such as artificial intelligence and neural nets being applied in the knowledge mining step to sample the data, provide outcomes then automatically test them on the whole data source to show that the outcome holds true for all available data not just one small subset (Smith & Ebrahim 2002). Data mining is a highly intensive machine process which utilises huge processing power, memory and time. Data sampling is often used as an initial step to reduce these constraints but correct utilisation may help to overcome this criticism also.

Data mining is simply a desperate search for something interesting without knowing what to look for. (Milloy 1995, Smith & Ebrahim 2002)

Exploratory mining, which is not constrained by user expectations, can uncover unexpected or unknown knowledge with wide reaching benefit and can be utilised to review and extend current medical knowledge. With the wealth of data being produced daily in the medical field the argument that it should not be used in an exploratory fashion to at least note important changes in data patterns demonstrates a misunderstanding of the potential value held therein. It is argued by some (Maindonald 1998, Smith & Ebrahim 2002) that it can be beneficial to look simply for something interesting rather than make an assumption about what is present in the data as if we only ever look for what is known we will potentially never find anything new and progress cannot be made. Provided this is a result of a scientific process then further mining or clinical trials can be undertaken for evidence to substantiate the initial findings. This criticism is only valid where the search is for anything interesting even if only minimally and where there is little or no validation.

Data mining displaces research and testing and presents results as facts requiring no further justification. (Milloy 1995)

Contrary to the criticism, data mining in medicine is generally viewed as an efficient tool for enhancing the work done in the field rather than as a replacement for it (Maindonald 1998). Its value is seen as a process of *automated serendipity* that stimulates and supports testing

rather than replaces it. When considering the use of mining outcomes there are two questions often asked; is this result representative of what has been recorded over time?, and can the analysis outcome be verified through real world application? (Smith & Ebrahim 2002). Whilst the first can be answered with some conviction by data mining the second requires clinical input and hence the process of providing trusted knowledge from data requires a collaborative effort by automated and clinical processes. When we consider that time from hypothesis to application of new knowledge is often measured in decades we should feel compelled to find new knowledge as quickly as possible and data mining offers the ideal tool for this.

6 Conclusions

History and current practice are important issues and need to be taken into account when constructing or justifying data mining techniques in medicine. This paper has outlined some of these issues.

There is a belief by some that the rate of medical breakthroughs of the calibre of those listed above has slowed dramatically since the 1970's (Horton 2000) This could be attributed to the inability of the human mind to manage the volume of data available and that most if not all patterns in data which may reveal knowledge and which occur frequently enough to be noticed by the human analyst are now known. This adds significant weight to the argument for the application of more effective and efficient automated technologies to uncover the less visible knowledge or less frequent but equally important patterns in the data. We must however learn from history and ensure that the validation requirements for knowledge acquisition, as discussed above, are adhered to by any automated process as for other methods of knowledge acquisition.

References

Adams, J. R. (1992), *Insect Potpourri: Adventures in Entomology*, Sandhill Crane Press.

al Andalusi, S. (1991), *Science and the Medieval World: "Book of the Categories of Nations"*, University of Texas, Austin.

Brieger, G. H. (1977), 'H l coulter. divided legacy: A history of the schism in medical thought.', *Isis* **69**(1), 103–105.

Buck, C., Llopis, A., Nájera, E. & Terris, M. (1988), *The Challenge of Epidemiology: Issues and Selected Readings*, World Health Organization.

Burke, J. (1985), *The Day the Universe Changed*, BBC, London.

Cios, K. & Moore, G. (2002), 'Uniqueness of medical data mining', *Artificial Intelligence in Medicine* **26**(1-2), 1–24.

Cosans, C. E. (1997), 'Galen's critique of rationalist and empiricist anatomy', *Journal of the History of Biology* **30**, 35–54.

Elwood, J. & Burton, R. (2004), 'Passive smoking and breast cancer: is the evidence for cause now convincing?', *Medical Journal of Australia* **181**(5), 236–237.

Fauser, B. C. & Edwards, R. G. (2005), 'The early days of IVF', *Human Reproduction Update* **11**(5), 437–438.

Forsstrom, J. J. & Rigby, M. (1998), Addressing the quality of the it tool - assessing the quality of medical software and information services, Technical report, University of Turku and Keele University.

Freitas, A. A. (2000), 'Understanding the crucial differences between classification and discovery of association rules: a position paper', *ACM SIGKDD Explorations Newsletter* **2**(1), 65–69.

Gebski, V. & Keech, A. (2003), 'Statistical methods in clinical trials', *Medical Journal of Australia* **178**(4), 182–184.

Gopal, K. (1997), Rationalist victory, Technical report, Online at http://www.iheu.org/node/668.

Hanson, A. E. (2006), Hippocrates: The "greek miracle" in medicine, Technical report, Online at www.medicineantiqua.org.uk/sa_hippint.html.

Horton, R. (2000), 'How sick is modern medicine?', *The New York Review of Books* **47**(17).

Houston, A., Chen, H., Hubbard, S. M., Schatz, B. R., Ng, T. D., Sewell, R. R. & Tolle, K. M. (1999), 'Medical data mining on the internet: Research on a cancer information system', *Artificial Intelligence Review, special issue on the Application of Data Mining* **13**(5-6), 437–466.

Katch, F. I. (1997), History makers, Technical report, Online at www.sportsci.org/news/history/lind/lind_sp.html.

Kaul, M. & Thadani, S. (2000), Development of philosophical thought and scientific method in ancient India, Technical report, Online at http://members.tripod.com/ INDIA_RESOURCE/scienceh.htm.

Maindonald, J. (1998), New approaches to using scientific data- statistics, data mining and related technologies in research and research training, Occasional paper, Australian National University.

Marshall, B. J. (1998), Peptic ulcers, stomach cancer and the bacteria which are responsible, Technical report.

Milloy, S. (1995), *Science without sense. The risky business of public health*, Cato Institute, Washington DC.

Mohaghegh, M. (1988), 'Miftah al-tibb wa minhaj altullab (a summary translation)', *Medical Journal of the Islamic Republic of Iran* **2**(1), 61–63.

Moher, D. (1998), 'CONSORT: An evolving tool to help improve the quality of reports of randomized controlled trials', *Journal of the American Medical Association* **279**(18), 1489–1491.

Muhaqqiq, M. (n.d.), 'Medical sects in islam', *al-Tawhid Islamic Journal* **8**(2).

Mulcahy, R. (1996), *Diseases: Finding the Cure*, The Oliver Press, Inc.

Pearcy, L. (1985), 'Galen: a biographical sketch', *Archaeology* **38**(6 (Nov/Dec)), 33–39.

Pollard, R. (2006), Fortuitous discovery led to a revolution in treatment, Technical report, Online at http://www.smh.com.au/news/science /fortuitous-discovery-led-to-a-revolution-in-treatment/ 2006/10/11/1160246197925.html.

Roddick, J. F., Fule, P. & Graco, W. J. (2003), 'Exploratory medical knowledge discovery : Experiences and issues', *SigKDD Explorations* **5**(1), 94–99.

Shillabeer, A. & Roddick, J. F. (2006), 'Towards role-based hypothesis evaluation for health data mining', *Electronic Journal of Health Informatics* **1**(1), e6.

Smith, G. D. & Ebrahim, S. (2002), 'Data dredging, bias or confounding', *British Medical Journal* **325**(21-28 December 2002), 1437–1438.

South Asian History Project (2002), Philosophical development from Upanishadic metaphysics to scientific realism, Technical report, Online at http://india_resource.tripod.com/upanishad.html.

Sprang, K. (2002), Dr. Edward Jenner and the smallpox vaccination, Technical report, Online at http://scsc.essortment.com/edwardjennersm_rmfk.htm.

Measuring Data-Driven Ontology Changes using Text Mining

Majigsuren Enkhsaikhan **Wilson Wong** **Wei Liu**[†] **Mark Reynolds**

School of Computer Science and Software Engineering
University of Western Australia
35 Stirling Highway, Crawley, WA 6009
E-mail: majigaa,wilson,wei,mark}@csse.uwa.edu.au

[†] Corresponding author.

Abstract

Most current ontology management systems concentrate on detecting usage-driven changes and representing changes formally in order to maintain the consistency. In this paper, we present a semi-automatic approach for measuring and visualising data-driven changes through ontology learning. Terms are first generated using text mining techniques using an ontology learning module, and then classified automatically into clusters. The clusters are then manually named, which is the only manual process in this system. Each cluster is considered as a sub-concept of the root concept, and thus one dimension of the feature space describing the root concept. The changes of terms in each cluster contributes to the change of the root concept. Using our system, Web documents are collected at different time periods and fed into the system to generate different versions of the same ontology for each time period. The paper presents several ways of visualising and analysing the changes. Initial experiments on online media data have demonstrated the promising capabilities of our system.

1 Introduction

An ontology for a dynamic domain is constantly evolving. It changes over time as the underlying knowledge fluctuates. Ontology changes also occur when human and software agents modify an ontology when applying them in either a centralised or distributed environment. (Flouris 2006) identified nine different tasks related to ontology change to summarise the state of the art. These include ontology mapping (Ehrig & Sure 2004), morphism (Kalfoglou & Schorlemmer 2005), alignment (Euzenat et al. 2004, de Bruijn et al. 2004), articulation, translation, evolution (Stojanovic 2004), versioning (Stojanovic 2004), integration (Pinto & Martins 2001) and merging (de Bruijn et al. 2004, Hitzler et al. 2005). Ontology mapping, morphism, alignment, articulation are to resolve the heterogeneities between ontologies to enable interoperability, while translation, evolution, versioning are to maintain the consistency and integrity of a single ontology in the face of changes, merging and integration are to unify all concepts and relations of source ontologies to suit the domain needs.

In this paper, we are interested in ontology evolution, which is defined as the timely adaptation to the changes in the business requirements, in the usage of the ontology-based applications, and in the modifications of consistent management of the changes (Stojanovic 2004). Changes may generate other changes because of the relations of vocabularies or axioms in a single ontology or the relations of several ontologies. Inconsistencies in the same or dependent ontologies can be also generated. Therefore, ontology evolution can become a very complex operation to maintain the consistency without losing information. In this paper, we simplify the notion of ontology evolution to only look at the change of vocabularies or terminologies over time in certain domains such that we can identify the changes and determine the amount of changes automatically.

Detecting and measuring ontology change over time is often done by manual inspection of different versions of the ontology before and after the change, or keeping change logs to record the modifications. In the case of ontologies generated automatically or semi-automatically through text mining (Liu et al. 2005), it is possible to automatically detect the changes and the degree of changes by comparing the ontologies generated for different time periods.

This paper describes a text mining approach for ranking domain terms and generating term clusters to measure the amount of changes in certain domains. Section 2 provides the background and relevant work in ontology evolution. Section 3 presents the techniques used in building different modules of our system. Section 4 discusses the experiments and results. The paper concludes with an outlook to future work in Section 5.

2 Related Work

2.1 Process of ontology evolution

According to (Stojanovic 2004), the process of ontology evolution has 6 cyclic phases:*Change Capturing*, *Change Representation*, *Semantics of change*, *Change implementation*, *Change propagation*, *Change validation*.

The approach proposed in this paper deals with the change capturing phase only. Changes can be captured from explicit requirements or implicit requirements. Explicit requirements generated by ontology engineers or end users are defined as *top-down changes*. Changes from implicit requirements like change discovery methods are called *bottom-up changes*. Implicit changes are categorized in three types: *structure-driven* change from ontology structure, *usage-driven* change from usage patterns created over time and *data-driven* change from modification of underlying knowledge such as text files. Specifically, while Stojanovic et. al. (Stojanovic 2004) discuss how to keep and mine the usage logs to discover usage driven changes, here we are interested in the data-driven changes that are hidden in large domain corpora, which is constantly modified and updated by adding new documents or deleting obsolete documents.

2.2 Formalisms for change representation

A change log records an exact sequence of changes that occurred when an ontology developer updated an old version of an ontology to a new version. If the recording of

a change log is unavailable in a dynamic and distributed environment like the Semantic Web, then old and new versions of an ontology bring the possibility to define changes (Klein 2004, Plessers et al. 2007). To declare changes from old and new versions of the same ontology, we can use the following techniques: *structural difference*, *conceptual change* and *transformation set*. Structural difference is a map of correspondences between old and new versions of the ontology. It gives the declarative view of ontology transition by comparing the versions of the ontology. Conceptual changes between old and new versions identify changes in the concepts. Transformation set provides a set of change operations that specify how an old ontology version can be transformed into a new version. It lists the changes as a result of comparison of two versions of the ontology.

Preserving history of an ontology is used for creating ontology versions. There are two ways to preserve history of an ontology. *Timestamp* approach labels elements of an ontology with specific time. *Snapshot* approach applies snapshots to capture the different states of an ontology over time. Snapshots can be for the whole ontology or for a single concept definition (Plessers 2006).

In this paper, we opt to the snapshot approach. At each time period, an ontology consisting of concepts and relations is kept. The same ontology at different time period populates an ontology repository.

2.3 Existing Ontology Evolution Systems

Existing ontology evolution systems are centered around two dominant ontology management infrastructures, the KArlsruhe ONtology and Semantic Web Framework (KAON) [1] and Protégé [2]. They both provide a suite of tools or plug-ins for creating, editing, modifying and reasoning about ontologies for Semantic Web applications.

Stojanovic (Stojanovic 2004) detects usage-driven changes by analysing usage patterns and query logs. Haase et. al. (Haase et al. 2004, Haase & Sure 2005)'s system to maintain consistency and discover changes during the usage of an ontology based application (in this case, an online digital library). Flouris et. al. (Flouris et al. 2006) proposed automatic ontology evolution based on belief change theory. Plessers et. al. (Plessers et al. 2007) developed *Change Definition Language* to facilitate formal representation and reasoning about changes. Noy et. al. (Noy et al. 2006) developed an ontology evolution framework in a collaborative environment. It supports different modes of ontology changes in a single framework by using several formalisms of change representation.

Overall, detecting usage-driven changes and maintaining consistency through reasoning about changes and formal representation are the main concerns of the current systems. Our system complements the existing ones on dealing with data-driven changes.

3 Change Detection using an Ontology Learning System

Figure 1 shows an overview of our system, consisting of an ontology learning module, a visualisation module and various sub-modules. Text collected from the Web is first pre-processed to remove spelling errors and resolve acronyms. Clean text is then fed into term extraction and concept discovery sub-modules to generate domain relevant terms and group relevant ones into clusters. The clusters are then named manually by domain experts. The clusters and the weights of the domain terms are then used in the visualisation module to produce graphs and tables to aid human comprehension of the results.

[1] http://kaon.semanticweb.org/
[2] http://protege.stanford.edu/

The main building blocks of domain ontologies are domain-specific concepts. Since concepts are merely mental symbols that we employ to represent the different aspects of a domain, they can never really be computationally captured from written or spoken resources. Instead, in ontology learning, terms are regarded as lexical realisations for expressing or representing concepts that characterise various aspects of specialised domains. Therefore, hereafter, we use terms and concepts interchangeably.

3.1 Term Extraction and Concept Discovery

The main task in term extraction is to determine whether a word or phrase is a term that characterises the target domain. This key question can be further decomposed to reveal two critical notions in this area, namely, *unithood* and *termhood*. Unithood concerns whether sequences of words should be combined to form stable lexical units, while termhood is the degree to which these stable lexical units are relevant to some domains. After processing the domain text using a linguistic parser and unithood analysis, Wong et al. (2007*b*,*a*) presented an empirically-derived scoring and ranking scheme for the determination of termhood based on a set of heuristically-motivated term characteristics. The scoring scheme ranks term candidates using numerical weights, indicating the significance of the concepts in the domain. Therefore, here we give a brief explanation of how we obtained the weights.

Two base measures are introduced for capturing the statistical evidence based on the cross-domain and intra-domain distribution of term candidates and their context words, respectively:

- *Domain Prevalence (DP)*, to measure the extent of term usage in the domain.
- *Domain Tendency (DT)*, to measure the extent of inclination of term usage in the domain.

A high *DP* means that the term is frequently encountered (i.e. prevalent) in the target domain. It is a sign of high domain relevance if and only if the frequent usage of that term is exclusive to the target domain, that is, has high domain tendency DT. These two measures are further adjusted when taking into account three types of linguistic evidence. Namely, **Candidate evidence** as *discriminative weight (DW)*, **Modifier evidence** as *modifier factor (MF)* and **Contextual evidence** as *average contextual discriminative weight (ACDW)*.

This new mechanism requires a corpus containing text generated using the special language of the target domain (i.e. domain corpus) and a set of corpora produced using special languages from domains other than the target domain (i.e. contrastive corpora). Details of how to obtain the above measures are presented below:

Given that we have a list of term candidates (both simple and complex) $TC = \{a_i, ..., a_n\}$, to determine termhood is to assign weights to term candidates in order to identify the m most suitable candidates as terms in a domain $t \in T$. Each complex term, a will comprise of a head a^h and modifiers $m \in M(a)$. Each term candidate is assigned a weight depending on its type (i.e. simple or complex). Inspired by the contrastive weights (CW)by Basili et. al. (Basili, Moschitti, Pazienza & Zanzotto 2001), the *domain prevalence (DP)* for a simple term a is defined as:

$$DP(a) = \log_{10}(f_{ad} + 10) \log_{10}\left(\frac{F_{TC}}{f_{ad} + f_{a\bar{d}}} + 10\right)$$

where $F_{TC} = \sum_j f_{jd} + \sum_j f_{j\bar{d}}$ is the sum of the frequencies of occurrences of all $a \in TC$ in both domain and contrastive corpora, while f_{ad} and $f_{a\bar{d}}$ are the frequencies of occurrences of a in the domain corpus and contrastive corpora, respectively. If the term candidate is complex, we

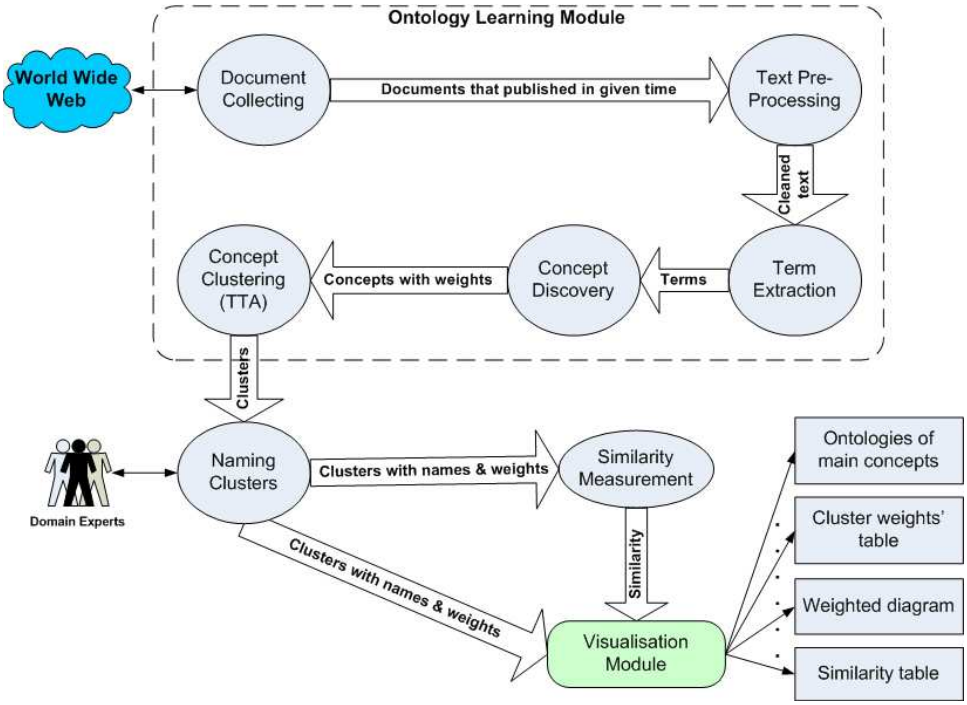


Figure 1: Architecture for Detecting Changes using an Ontology Learning System

define its *DP* as:

$$DP(a) = \log_{10}(f_{ad} + 10)DP(a^h)MF(a)$$

Based on our preliminary experiments on comparing *DP* with the original *CW*, to address the biased ranking by *CW*, we add a constant 10 to f_{ad} prior to log and introduce another new measure called *modifier factor (MF)*. The *MF* of a complex term a is measured using the cumulative domain frequency and cumulative contrastive frequency of modifiers which also happen to be term candidates, $m \in M_a \cap TC$. Formally, the *MF* of a complex term a is defined as:

$$MF(a) = \log_2\left(\frac{\sum_{m \in M_a \cap TC} f_{md} + 1}{\sum_{m \in M_a \cap TC} f_{m\bar{d}} + 1} + 1\right)$$

MF is actually a derived measure modelled after our second new base measure *domain tendency (DT)*. The only difference is that *MF* works with modifiers while *DT* works with the entire term candidate, both simple and complex. Formally, we can determine the inclination of using term candidate a for domain and non-domain purposes through:

$$DT(a) = \log_2\left(\frac{f_{ad} + 1}{f_{a\bar{d}} + 1} + 1\right)$$

If term candidate a is equally common in both domain and non-domain (i.e. contrastive domain), $DT = 1$. If the usage of a is more inclined toward the target domain, $f_{ad} > f_{a\bar{d}}$, then $DT > 1$, and $DT < 1$ otherwise. Next, this new base measure *DT* together with *DP* will contribute to a new weight known as *discriminative weight (DW)*, which is simply the product of *DP* and the corresponding *DT* of the term candidate:

$$DW(a) = DP(a)DT(a)$$

Assuming that term candidate a has the set of context words C_a, the *average contextual discriminative weight (ACDW)* is defined as:

$$ACDW(a) = \frac{\sum_{c \in C_a} DW(c)NGD(a,c)}{|C_a|}$$

where $NGD(a,c)$ is the normalized google distance (Cilibrasi & Vitanyi 2006) between term candidate a and c. The *ACDW* weight allows us to take into consideration the company a term candidate keeps. Nonetheless, not all context words describe or are related to the terms they appear with. Therefore, we adjust *ACDW* according to its ratio with the corresponding *DW* to produce the *adjusted contextual contribution (ACC)* as:

$$ACC(a) = ACDW(a)\frac{e^{\left(1-\frac{ACDW(a)+1}{DW(a)+1}\right)}e^{\left(1-\frac{DW(a)+1}{ACDW(a)+1}\right)}}{\log_2 \frac{ACDW(a)+1}{DW(a)+1} + 1}$$

In the end, we define the final weight known as *termhood (TH)* for each term candidate as:

$$TH(a) = DW(a) + ACC(a) \quad (1)$$

3.2 Concept clustering

Once the term candidates have been scored and ranked, filtering and selection are usually performed using some thresholds or with minimal expert involvements. The result is a list of terms which has been deemed fit to denote some domain-specific concepts or is relevant to certain domains of interest. As shown in Figure 1, these terms are the ones that will contribute to concept discovery. Wong et al. (2006, 2007*c*) presented a novel clustering algorithm known as the *Tree-Traversing Ant (TTA)* for discovering concepts from terms. *TTA* was designed as an attempt to fuse the strengths of standard ant-based methods with certain advantages of conventional clustering methods. One of the biggest issues associated with term clustering is the lack of visible (e.g. physical and behavioral) features required for the computation of similarity. Together with the use of featureless similarity measures, namely *Normalised Google Distance (NGD)* (Cilibrasi & Vitanyi 2006) and *n° of Wikipedia (n°W)* (Wong et al. 2007*c*), *TTA* was able to address the issues related to similarity measurement and many other unique requirements of term clustering in ontology learning. Seven of the most notable strengths of *TTA* with respect to clustering are:

- Able to further distinguish hidden structures within clusters;
- Flexible in regards to the discovery of clusters;
- Capable of identifying and isolating outliers;
- Tolerance to differing cluster sizes;
- Able to produce consistent results;
- Able to identify implicit taxonomic relationships between clusters; and
- Inherent capability of coping with synonyms, word senses and the fluctuation in terms usage.

3.3 Measuring the similarities between concepts

Techniques for measuring the similarities between concepts can employ extensive and well-ground semantic resources such as WordNet, OpenCyc or domain specific ontologies to compute the distance between two concepts. For example, (Maynard & Ananiadou 1999, 2000) consult the *Unified Medical Language System (UMLS)* to compute two weights, namely, positional and commonality in order to measure the similarity between two concepts. Positional weight is obtained based on the combined number of nodes belonging to each word, while commonality is measured by the number of shared common ancestors multiplied by the number of words. Accordingly, the similarity between two term candidates is defined as:

$$sim(a,b) = \frac{com(a,b)}{pos(a,b)} \quad (2)$$

where $com(a,b)$ and $pos(a,b)$ is the commonality and positional weight respectively, between term candidate a and b.

However, such an approach is not viable when there is no ready to use domain ontology or taxonomy available. In addition, Basili et. al. (Basili, Pazienza & Zanzotto 2001) criticised that such approach for being so reliant on existing ontologies and thus not portable to other domains. Therefore, they combine the use of contextual information and the head-modifier principle to capture term candidates and their context words on a feature space for computing similarity. Given the term candidate a, the feature vector for a is:

$$\tau(a) = (f_1, ..., f_n)$$

where f_i is the value of the attribute F_i and n is the number of features. The authors chose cosine measure for computing similarity over the syntactic feature space:

$$sim(\tau_i, \tau_j) = \frac{\tau_i \tau_j}{|\tau_i||\tau_j|}$$

This approach require large corpora and high density of the domain terms to be effective. To ensure the general applicability of our approach, the cosine similarity measure is more suitable for us in measuring the similarities between ontologies generated at different periods of time. This will free us from assuming prior ontologies. It is also feasible because our system is capable of collecting large volumes of online text data to ensure an optimal size of the corpora. As you will see in Section 4, a general root concept is represented using a vector of sub-domains (or clusters), the children concepts in each sub-domain contributes to the overall weight of the sub-domain. Therefore, changes in the significance of each sub-domain measured by the corresponding overall weight can be used in measuring the shift in the semantics of the root concept. Cosine similarity measure is used to compute the amount of changes over a specified time period.

4 Experiments and Results

The *document collecting* sub-module as shown in Figure 1 reads the RSS feed from major Australian online News outlets every day from September 2006 to April 2007. We choose to use online news articles to demonstrate the capability of our system because media data are more volatile than other types of domain corpora. Changes are therefore more obvious.

As online media articles in Australia cover news in Australia as well as international news that made their way into Australia, we treat the terms and relations extracted using the ontology learning module as features of a general root concept `Australia`. As far as the online media domain is concerned, `Australia` is a constantly evolving concept. In our case, the changes are represented by the changes in its features, that is, terms and relations.

As shown in Figure 2-4, three versions of the ontology for the root concept `Australia` in the media domain are generated for September 2006, December 2006 and April 2007. The generation of terms and the creation of links are done automatically using the termhood measurement TH and Tree Traversing Ants algorithm discussed in Section 3. Unfortunately, as it is still an open research question we are currently pursuing, there is not yet means of automatically naming the clusters in our current system. Manual input from domain experts is therefore necessary in determining the name of the clusters based on the common node shared by the terms. Identifying the cluster is important in our approach of measuring the amount of changes, because each cluster is considered as a sub-domain and therefore a dimension in the feature space of the general root concept. In our case, the concept `Australia` in the media domain is considered represented by a vector with 7 features, namely, `Politics`, `Economy`, `Policies`, `Environmental Issues`, `Domestic Affairs` and `International Affairs`. However, as one will notice that some anomaly might exist. For example, `poor housing affordability` ideally should be classified in the domestic affairs cluster. This issue was discussed in (Wong et al. 2006, 2007*c*), the infrequent occurrences of such anomalies in a semi-automatic system is tolerable.

The *visualisation module* in Figure 1 allows different types of plug-ins to display the changes in sensible ways to ease the comprehension and interpretation of the changes. To quantify the changes and to weigh the importance of each cluster in the general root concept, we defined the accumulative weight (ω) of each cluster as the sum of the TH value of each term in the cluster,

$$\omega_i = \sum_{k=1}^{n} (TH_k)$$

where n is the number of terms in cluster i.

Figure 5 represents these accumulative weights of each cluster for all three periods. The data can be better visualised using scaled circles to indicate the amount of media attention in certain feature dimensions, as shown in Figure 6.

It is not surprising to see that media attention to the main topic areas in Australia are constantly changing. Figure 6 is very intuitive in showing the volatile clusters and the relative stable clusters. From here, it is obvious that some areas, such as coverage on `Environmental issues`, `Sports and Entertainment` and `Economy`, are changing quite dramatically over different periods, while `Politics` and `Domestic Affairs`, are relatively stable. The steady increase in the amount politics coverage from September 2006, to December 2006 and April 2007 indeed makes sense, as one would expect more talking-about on politics as the election draws close.

Overall changes in such an ontology are generated by the changes of internal terms. Looking into the terms comprising of each cluster, the topic changes depend on many aspects such as sudden events, sports season, weather and

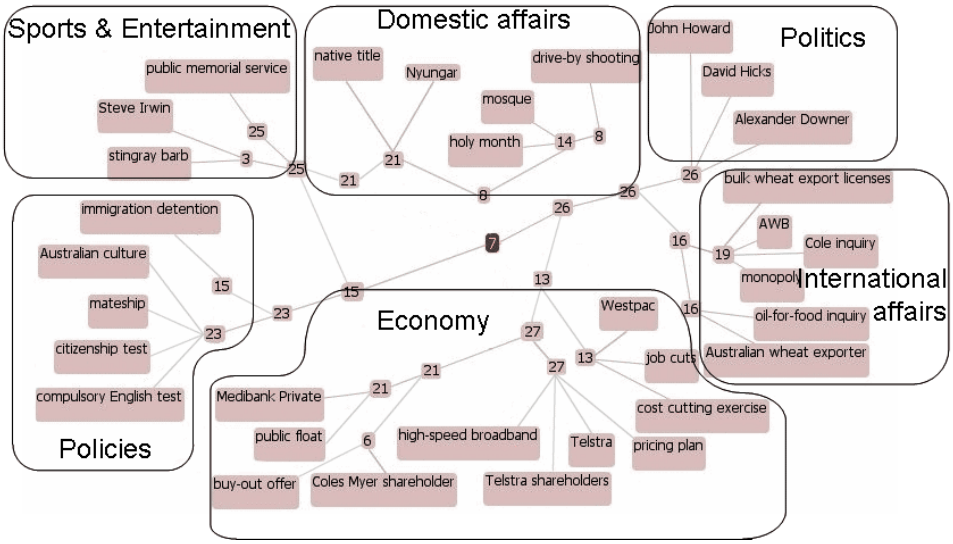


Figure 2: Clusters of September 2006

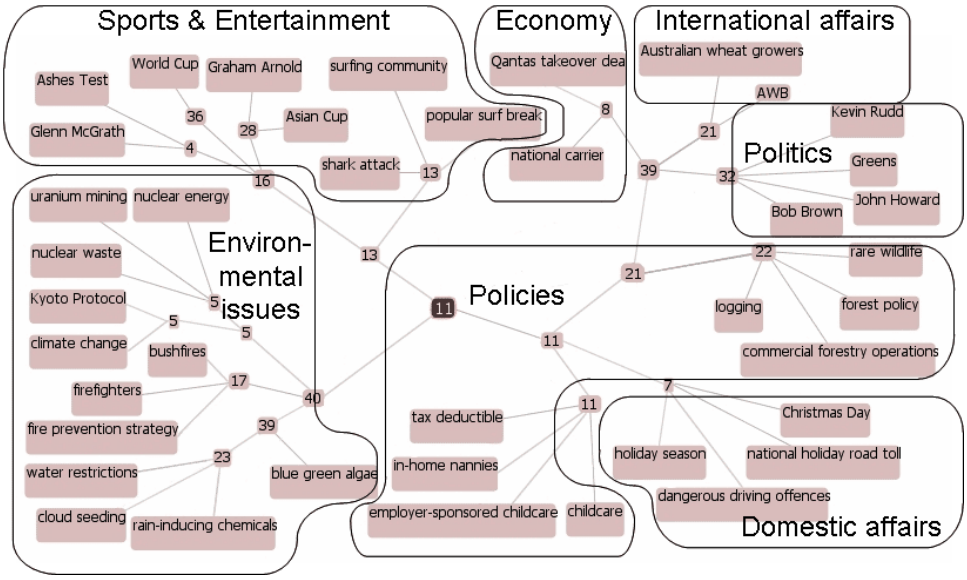


Figure 3: Clusters of December 2006

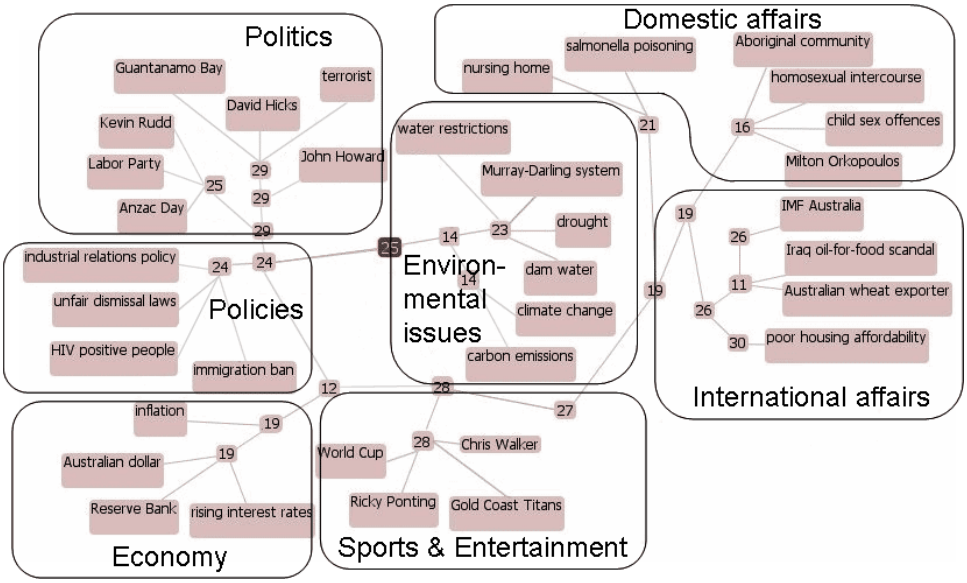


Figure 4: Clusters of April 2007

Cluster / Period	Politics	Sports & Entertainment	Economy	Environmental issues	International affairs	Domestic affairs	Policies
April 2007	120.37	60.81	52.86	80.3	49.68	91.54	59.83
December 2006	90.75	128.93	28.9	105.17	33.09	47.3	107
September 2006	69.96	54.54	161.08	0	92.09	72.17	4

Figure 5: Cluster weights

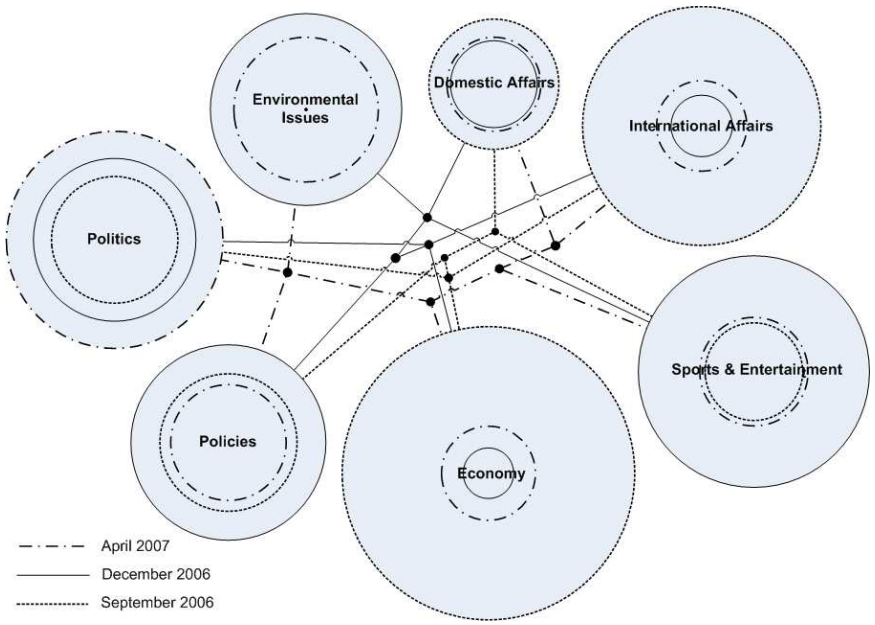


Figure 6: Clusters in all three periods

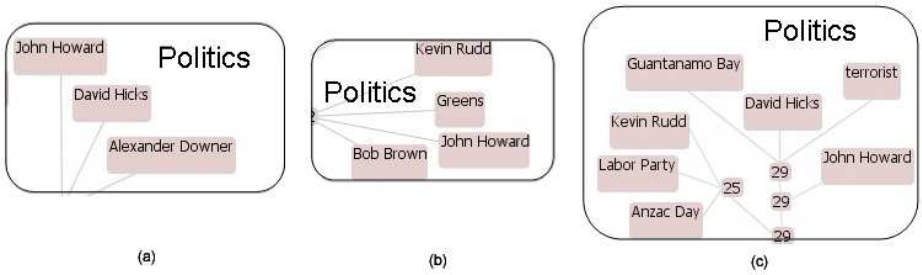


Figure 7: Politics area: (a) in September 2006; (b) in December 2006; (c) in April 2007

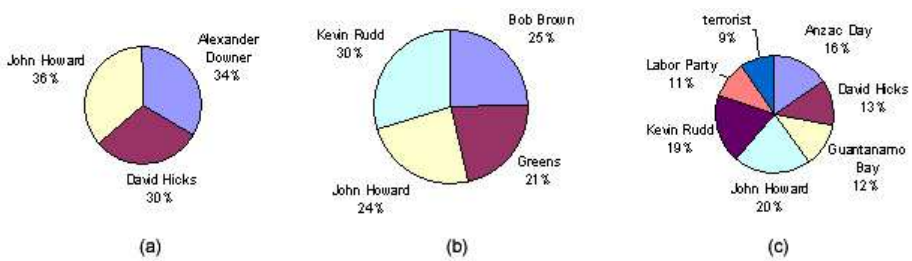


Figure 8: Term contributions in Politics: (a) in September 2006; (b) in December 2006; (c) in April 2007

Time periods	April 2007	December 2006	September 2006
April 2007	x	0.627	0.762
December 2006	0.627	x	0.622
September 2006	0.762	0.622	x

Figure 9: Similarity measurement

etc. In December 2006, which was a hot summer in Australia, `bushfires` appeared and environmental issues became one of the hot topics, even though they were not strongly mentioned during September 2006. Taken the `Politics` cluster for example, Figure 7 zoom into this cluster in the three different periods. As we can see, all three of them includes the Prime Minister of Australia, Mr. John Howard, as one of the main terms. Other terms differed and did not appear in all of them. Before becoming the new Opposition leader, Mr. Kevin Rudd was not in the cluster in September 2006, but appeared in December 2006 and came stronger in significance (TH value) after he took the position. Therefore, measuring the contribution from each terms to the cluster can also be important. As shown in Figure 8(b), `Kevin Rudd` contributes 30% to the cluster as compared to `John Howard`'s 24%. The overall changes in the ontology over different time periods can be measured using the cosine similarity of the vectors representing each period. Given two versions of the ontology at two different time period, O_t and $O_t^{'}$, the vectors are $\tau = (\omega_1, ..., \omega_m)$ and $\tau^{'} = (\omega_1^{'}, ..., \omega_m^{'})$, where m is the number of features. According to the formula in Subsection 3.3, we have

$$sim(O_t, O_t^{'}) = \frac{\Sigma_{i=1}^{n}(\omega_i \omega_i^{'})}{\sqrt{\Sigma_{i=1}^{n}(\omega_i)^2}\sqrt{\Sigma_{i=1}^{n}(\omega_i^{'})^2}}$$

where n is the number of clusters in the ontology, ω_i is the cumulative weight for a selected cluster. A $sim(O_t, O_t^{'})$ value closer to 1 indicates higher similarity, whereas a value closer to 0 signifies higher distance. We measured the cosine similarity of every two different versions for the general root concept `Australia`. The value varied between 0.62 and 0.76 in our selected time periods as shown in Figure 9.

From here, we can say that the concept `Australia` fluctuates 24%∼38% between September 2006 and April 2007.

5 Conclusion and Future Work

Data driven ontology changes can be detected using text mining based ontology learning systems. In this paper, we presented an approach for semi-automatically generating ontology concept clusters at different time periods, measuring and visualising the changes in sensible ways to help understand the overall concept changes as well as the individual terms contributed to the change. Experiments have confirmed the effectiveness and intuitiveness of the system. The results are also quite sensible in explaining real world events.

As the data source is from volatile online media domain, the generated ontology shows little consistency at the term level. However, this can be helped by grouping the relevant terms into clusters, which are relatively stable over the time span considered. In the future, comparison experiments are planned on more domain specific text to see if the same approach can be used to discover general or upper ontology.

References

Basili, R., Moschitti, A., Pazienza, M. & Zanzotto, F. (2001), A contrastive approach to term extraction, *in* 'Proceedings of the 4th Terminology and Artificial Intelligence Conference (TIA)', France.

Basili, R., Pazienza, M. & Zanzotto, F. (2001), Modelling syntactic context in automatic term extraction, *in* 'Proceedings of the International Conference on Recent Advances in Natural Language Processing', Bulgaria.

Cilibrasi, R. & Vitanyi, P. (2006), Automatic extraction of meaning from the web, *in* 'Proceedings of the IEEE International Symposium on Information Theory', Seattle, USA.

de Bruijn, J., Martin-Recuerda, F., Manov, D. & Ehrig, M. (2004), State-of-the-art survey on ontology merging and aligning, Sekt deliverable d4.2.1, Digital Enterprise Research Institute, University of Innsbruck.

Ehrig, M. & Sure, Y. (2004), Ontology mapping - an integrated approach, *in* C. Bussler, J. Davis, D. Fensel & R. Studer, eds, 'Proceedings of the 1st European Semantic Web Symposium (ESWS 2004) on The Semantic Web: Research and Applications', Vol. 3053 of *Lecture Notes in Computer Science*, Springer, pp. 76–91.

Euzenat, J., Bach, T. & J. Barrasa, P. Bouquet, e. a. (2004), State of the art on ontology alignment, Knowledge web deliverable, Knowledge Web Consortium.

Flouris, G. (2006), On Belief Change and Ontology Evolution, PhD thesis, University of Crete, Greece.

Flouris, G., Plexousakis, D. & Antoniou, G. (2006), Evolving ontology evolution, *in* 'Proceedings of the 32nd International Conference on Current Trend in Theory and Practice of Computer Science (SOFSEM-2006)'.

Haase, P. & Sure, Y. (2005), Incremental ontology evolution - evaluation, Sekt deliverable d3.1.2, Institute AIFB, University of Karlsruhe.

Haase, P., Sure, Y. & Vrandecic, D. (2004), Ontology management and evolution - survey, methods and prototypes, Sekt deliverable d3.1.1, Institute AIFB, University of Karlsruhe.

Hitzler, P., Krotzsch, M., Ehrig, M. & Sure, Y. (2005), What is ontology merging? - a category-theoretical perspective using pushouts.

Kalfoglou, Y. & Schorlemmer, M. (2005), Ontology mapping: The state of the art, *in* Y. Kalfoglou, M. Schorlemmer, A. Sheth, S. Staab & M. Uschold, eds, 'Semantic Interoperability and Integration', number 04391 *in* 'Dagstuhl Seminar Proceedings'.

Klein, M. (2004), Change Management for Distributed Ontologies, PhD thesis, Vrije Universiteit Amsterdam.

Liu, W., Weichselbraun, A., Scharl, A. & Chang, E. (2005), 'Semi-automatic ontology extension using spreading activation', *Journal of Universal Knowledge Management* **0**(1), 50–58.

Maynard, D. & Ananiadou, S. (1999), Term extraction using a similarity-based approach, *in* 'Recent Advances in Computational Terminology', John Benjamins.

Maynard, D. & Ananiadou, S. (2000), Identifying terms by their family and friends, *in* 'Proceedings of the 18th International Conference on Computational Linguistics', Germany.

Noy, N., Chugh, A., Liu, W. & Musen, M. (2006), A framework for ontology evolution in collaborative environments, *in* 'Proceedings of the 5th International Semantic Web Conference (ISWC-2006)', Vol. 4273 of *Lecture Notes in Computer Science*.

Pinto, H. & Martins, J. (2001), Ontology integration: How to perform the process, *in* 'Proceedings of IJCAI2001 Workshop on Ontologies and Information Sharing', AAAI Press, pp. 71–80.

Plessers, P. (2006), An Approach to Web-Based Ontology Evolution, PhD thesis, Vrije Universiteit Brussel.

Plessers, P., Troyer, O. D. & Casteleyn, S. (2007), 'Understanding ontology evolution: A change detection approach', *Journal of Web Semantics: Science, Services and Agents on the World Wide Web* **5**, 39–49.

Stojanovic, L. (2004), Methods and Tools for Ontology Evolution, PhD thesis, University of Karlsruhe.

Wong, W., Liu, W. & Bennamoun, M. (2006), Terms clustering using tree-traversing ants and featureless similarities, *in* 'Proceedings of the International Symposium on Practical Cognitive Agents and Robots', Perth, Australia.

Wong, W., Liu, W. & Bennamoun, M. (2007*a*), Determining termhood for learning domain ontologies in a probabilistic framework, *in* 'Proceedings of the 6th Australasian Conference on Data Mining (AusDM)', Gold Coast.

Wong, W., Liu, W. & Bennamoun, M. (2007*b*), Determining the unithood of word sequences using mutual information and independence measure, *in* 'Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics (PACLING)', Melbourne, Australia.

Wong, W., Liu, W. & Bennamoun, M. (2007*c*), 'Tree-traversing ant algorithm for term clustering based on featureless similarities', Journal on Data Mining and Knowledge Discovery, doi: 10.1007/s10618-007-0073-y.

Determining Termhood for Learning Domain Ontologies using Domain Prevalence and Tendency

Wilson Wong, Wei Liu and Mohammed Bennamoun

School of Computer Science and Software Engineering
University of Western Australia
Crawley WA 6009
{wilson,wei,bennamou}@csse.uwa.edu.au

Abstract

In the course of reviewing existing automatic term recognition techniques for applications in ontology learning, we came across four issues which can be improved upon. We proposed a new mechanism that incorporates both statistical and linguistic evidences for the computation of a final weight defined as *Termhood (TH)* for ranking term candidates. The analysis of the frequency distributions of the term candidates during our initial experiments revealed three advantages for higher quality term recognition.

1 Introduction

Automatic term recognition, also known as *term extraction* or *terminology mining*, is an integral part of many applications that deal with natural language such as document retrieval (Teevan & Karger 2003), automatic thesaurus construction (Grefenstette 1994), and ontology learning (Wong, Liu & Bennamoun 2007*b*). It involves the extraction and filtering of term candidates for the purpose of identifying domain-relevant terms. The main aim in automatic term recognition is to determine whether a word or a sequence of words is a term that characterises the target domain. The key question can be further decomposed to reveal two critical notions in this area, namely *unithood* and *termhood*. Formally, (Kageura & Umino 1996) defines unithood and termhood as the *"degree of strength or stability of syntagmatic combinations and collocations"* and *"degree that a linguistic unit is related to domain-specific concepts"*, respectively. Unithood is only relevant to *complex terms* (i.e. multi-word terms), while termhood deals with both *simple terms* (i.e. single-word terms) and complex terms.

The determination of unithood and of termhood inevitably requires the use of frequency of occurrence or co-occurrence of words and documents. This is clearly demonstrated through surveys of term extraction approaches by (Kit 2002, Cabre-Castellvi, Estopa & Vivaldi-Palatresi 2001, Kageura & Umino 1996). The difference between unithood and termhood measures lies in how the frequency is employed as evidence. Unithood measurements rely heavily on statistical tests or information-theoretic measures for determining if a sequence of words has strong collocational strength (Wong, Liu & Bennamoun 2007*a*), while termhood determination mostly employ measures of relevance such as those in information retrieval.

This paper is the result of our attempt to look at how advances in automatic term recognition can assist in the term extraction phase of ontology learning. Surveys (e.g. (Gomez-Perez & Manzano-Macho 2003)) have shown that many existing approaches in ontology learning merely employ isolated and non-comprehensive techniques that were not designed to address the various requirements and peculiarities in terminology. During our review of the start-of-the-art in automatic term recognition, we have identified four issues and open problems that remain unaddressed:

- *Inadequate attention to the difference between prevalence and tendency*: One of the main issues in automatic term recognition is that many frequency-oriented techniques adapted from *term frequency inverse document frequency (TF-IDF)* and others alike from information retrieval fail to comprehend that terms are properties of domains and not documents (Basili, Moschitti, Pazienza & Zanzotto 2001). Existing weights do not reflect the tendency of term usage across different domains. They merely measure the prevalence of the term in a particular target domain.

- *Oversimplication of the role of heads and modifiers*: Many approaches have attempted to utilise the head as the representative of the entire complex term in various occasions. Such move is an oversimplication of the relation between a complex term and its head. For example, the assumption that *"term sense is usually determined by its head"* by (Basili, Pazienza & Zanzotto 2001) is not entirely true since heads are inherently ambiguous and modifiers are required to narrow down their possible interpretations.

- *How to determine the relatedness between terms and their context?*: For approaches that attempt to use contextual information, they have realised the prior need to examine the relatedness of context words with the associated term candidate. The existing solutions to this requirement have yet to deliver overall satisfactory results. For example, (Maynard & Ananiadou 1999) rely on the use of rare and static resources for computation of similarity while others such as (Basili, Pazienza & Zanzotto 2001) require large corpora to enable the extraction of context as features to enable the use of feature-based similarity measures.

- *The overemphasis on the role of contextual evidence*: Many researchers have repeatedly stressed on the significance of the old cliché *"you shall know a word by the company it keeps"*. Unless one has the mechanism of identifying the

companies which are truly in the position to describe a word, the overemphasis on contextual evidence may results in negative effects.

To address the highlighted issues above, we propose a new scoring and ranking mechanism that incorporates a series of weights that lead to a final new score known as *Termhood (TH)*. The main aim of this new mechanism is to utilise as much evidences as possible in order to improve the approximation of termhood. The mechanism consists of two new base measures which capture the statistical evidence, and four new derived measures that employ the statistical evidence to quantify the linguistic evidences. In Section 2, we will have an elaborate review on the existing techniques for measuring termhood. In Section 3, we will present our new mechanism, the measures involved and the justification behind every aspect of these measures. In Section 4, we will summarize some findings from our initial experiments. Finally, we conclude this paper with an outlook to future works in Section 5.

2 Related Works

Commonly, the mechanism for assessing termhood will require a ranking scheme, similar to that of relevance ranking for information retrieval, where each term is assigned a score. Such ranking scheme will assist in the selection of *"true"* terms from less likely ones. Existing measures based on formal probabilistic models for determining the relevance of words with respect to certain topics or documents are mainly studied within the realm of document retrieval and automatic indexing. In probabilistic indexing, one of the first few detailed quantitative models was proposed by (Bookstein & Swanson 1974). In this model, the differences in the distributional behavior of words is employed as a guide to determine if a word should be considered as an index term. This model is founded upon works on how function words can be closely modeled by a Poisson distribution whereas content words deviates from it (Church & Gale 1995, Manning & Schutze 1999). An even larger collection of literature on probabilistic models can be found in a related area of document retrieval. The simplest of all the retrieval models is the binary independence model (Fuhr 1986, Lewis 1998). As with all other retrieval models, the binary independence model is designed to estimate the probability that a document j is considered as relevant given a specific query k. Let $T = \{t_1, ...t_n\}$ be the set of terms in the collection of documents (i.e. corpus). We can then represent the set of terms T_j occurring in document j as a binary vector $v_j = \{x_1, ..., x_n\}$ where $x_i = 1$ if $t_i \in T_j$ and $x_i = 0$ otherwise. This way, the odds of document j, represented by a binary vector v_j being relevant to query k can be computed as (Fuhr 1992):

$$O(R|k, v_j) = \frac{P(R|k, v_j)}{P(\bar{R}|k, v_j)} = \frac{P(R|k)}{P(\bar{R}|k)} \frac{P(v_j|R, k)}{P(v_j|\bar{R}, k)}$$

and based on the assumption of independence between the presence and absence of terms,

$$\frac{P(v_j|R, k)}{P(v_j|\bar{R}, k)} = \prod_{i=1}^{n} \frac{P(x_i|R, k)}{P(x_i|\bar{R}, k)}$$

Other more advanced models that take into considerations other factors such as term frequency, document frequency and document length have also been proposed (Jones, Walker & Robertson 1998).

Besides formal models for term weighting and ranking, a more straightforward and commonly-adopted method is *term frequency inverse document frequency (TF-IDF)* and its variants (Salton & Buckley 1988). (Basili, Moschitti, Pazienza & Zanzotto 2001) proposed a TF-IDF inspired measure for assigning terms with more accurate weight that reflects their specificity with respect to the target domain. This contrastive analysis is based on the heuristic that general language-dependent phenomena should spread similarly across different domain corpus and special-language phenomena should portray odd behaviors. This *contrastive weight* for simple term candidate a in target domain d is defined as:

$$CW(a) = \log f_{ad} \left(\log \frac{\sum_j \sum_i f_{ij}}{\sum_j f_{aj}} \right) \quad (1)$$

where f_{ad} is the frequency of the simple term candidate a in the target domain d, $\sum_j \sum_i f_{ij}$ is the sum of the frequencies of all term candidates in all domains, and $\sum_j f_{aj}$ is the sum of the frequencies of term candidate a in all domains. For complex term candidates, the frequency of their heads are utilised to compute their weights. This is necessary as the low frequencies among complex terms make estimations difficult. Consequently, the weight for complex term candidate a in domain d is defined as:

$$CW(a) = f_{ad} CW(a^h) \quad (2)$$

where f_{ad} is the frequency of the complex term candidate a in the target domain d, and $CW(a^h)$ is the contrastive weight for the head of the complex term candidate, a^h. The use of heads by (Basili, Moschitti, Pazienza & Zanzotto 2001) for computing the contrastive weights $CW(a)$ for complex term candidates reflects the head-modifier principle (Hippisley, Cheng & Ahmad 2005). The principle suggests that the information being conveyed by complex terms manifest itself in the arrangement of the constituents. The head acts as the key that refers to a general category to which all other modifications of the head belong. The modifiers are responsible for distinguishing the head from other forms in the same category.

Besides contrastive analysis, the use of contextual evidence to assist in the correct identification of terms has become popular. There are currently two dominant approaches to extract context words: the use of fixed-size windows (Maynard & Ananiadou 1999), and the use of grammatical relations (Basili, Pazienza & Zanzotto 2001, LeMoigno, Charlet, Bourigault, Degoulet & Jaulent 2002). One of the works along the line of incorporating contextual evidence is *Cvalue* and *NCvalue* by (Frantzi & Ananiadou 1997). *Cvalue* can be regarded as a unithood measure that contributes to the calculation of *NCvalue*. Discussions on *Cvalue* is beyond the scope of this paper. It suffices to know that given a simple or complex term candidate a to be examined for unithood, the *Cvalue* is defined as:

$$Cvalue(a) = \begin{cases} \log_2 |a| f_a & \text{if } |a| = g \\ \log_2 |a| \left(f_a - \frac{\sum_{l \in L_a} f_l}{|L_a|} \right) & \text{otherwise} \end{cases} \quad (3)$$

where $|a|$ is the number of words in a, L_a is the set of potential longer term candidates that contain a, and g is the longest n-gram considered, f_a is frequency of occurrences of a in the corpus. As for *NCvalue*, this measure involves the assignment of weights to context words (in the form of nouns, adjectives and verbs) located within a fixed-size window from the term candidate. Given that TC is the set of all term candidates

and c is a noun, verb or adjective which appears with term candidates, the $weight(c)$ is defined as:

$$weight(c) = 0.5 \left(\frac{|TC_c|}{|TC|} + \frac{\sum_{a \in TC_c} f_a}{f_c} \right) \quad (4)$$

where $TC_c \subset TC$ is the set of term candidates that appear with c, $\sum_{a \in TC_c} f_a$ is the total frequency of c appearing with term candidates, and f_c is the frequency of c. After calculating the weights for all possible context words, the sum of weights for context words appearing with each term candidate can be obtained. Formally, for each simple or complex term candidate a that has a set of accompanying context words C_a, the cumulative context weight is defined as:

$$cweight(a) = \sum_{c \in C_a} weight(c) + 1 \quad (5)$$

Eventually, the *NCvalue* for a term candidate is defined as:

$$NCvalue(a) = \frac{1}{\log F} Cvalue(a) cweight(a) \quad (6)$$

where F is the size of the corpus in terms of number of words.

There has also been an increasing interest in incorporating semantic information for measuring termhood. The use of semantic measures is mainly to gauge the relatedness of context words with the associated term candidates in the process of measuring termhood. Maynard & Ananiadou (Maynard & Ananiadou 1999) employ the *Unified Medical Language System (UMLS)* to compute two weights, namely, positional $pos(a, b)$ and commonality $com(a, b)$ where a and b are term candidates. The *UMLS* is organised as a hierarchical structure of concepts. Each concept has a set of related terms. Positional weight is obtained based on the combined number of concepts belonging to each term, while commonality is measured by the number of shared common ancestors multiplied by the number of times the term occurs. The similarity of two term candidates is simply $sim(a, b) = com(a, b)/pos(a, b)$. The authors then modified the *NCvalue* discussed in Equation 6 by incorporating the similarity measure as part of a *context factor (CF)* (Maynard & Ananiadou 2000) defined as:

$$CF(a) = weight(c) \sum_{c \in C_a} f_c + sim(a, b) \sum_{b \in CT_a} f_b$$

where C_a is the set of context words of a, f_c is the frequency of c as a context word of a, $weight(c)$ is the weight for context word c as defined in Equation 4, CT_a is the set of context words of a which also happens to be term candidates (i.e. context terms), f_b is the frequency of b as a context term of a, and $sim(a, b)$ is the similarity between term candidate a and its context term b using *UMLS*. The new *NCvalue* is defined as:

$$NCvalue(a) = 0.8Cvalue(a) + 0.2CF(a)$$

(Basili, Pazienza & Zanzotto 2001) commented that the use of extensive and well-grounded semantic resources by (Maynard & Ananiadou 1999) faces the issue of portability to other domains. Instead, (Basili, Pazienza & Zanzotto 2001) combine the use of contextual information and the head-modifier principle to capture term candidates and their context words on a feature space for computing similarity. Given the term candidate a, the feature vector for a is $\tau_a = (v_1, ..., v_n)$ where v_i is the value of the attribute F_i and n is the number of features. F_i comprises of the tuple (T_i, h_i) where T_i is the type of grammatical relations (e.g. *subj*, *obj*) and h_i is the head of the relation. In other words, only the head of complex terms will be used as referent to the entire structure. According to the authors, *"the term sense is usually determined by its head."*. This statement opposes the fundamental fact, not only in terminology but in general linguistic, that simple terms are polysemous and modification of such terms are necessary to narrow down their possible interpretations (Hippisley et al. 2005). The authors chose the cosine measure for computing similarity, $sim(\tau_a, \tau_b)$ between two terms over the syntactic feature space. To assist in the ranking of the term candidates using their heads, a hand-crafted controlled terminology *CTO* is employed as evidence of *"correct"* terms during the computation of similarity. Accordingly, given that $\tau_{CTO} = \sum_{e \in CTO} \tau_e$ the measure for assigning weight to term candidate a is defined as:

$$ext(a) = sim(\tau_a, \tau_{CTO}) f_a \quad (7)$$

where f_a is the frequency of a in the corpus.

3 The Proposed Approach for Combining Termhood Evidences

We propose a new mechanism for scoring and ranking that employs distributional behaviors of term candidates within the target domain and also across different domains as statistical evidence to quantify the linguistic evidences in the form of candidate, modifier and context. The evidences are gathered from two types of corpus, namely, *domain corpus* d which contains text in the target domain, and *contrastive corpus* $\bar{d}$ which contains text across different genre other than the target domain. Since the quality of the evidences of terms with respect to the domain is dependent on the issue of representativeness of the corresponding corpus, we will assume that both d and $\bar{d}$ are balanced, unbiased and randomised samples of the population text representing the corresponding domain. The actual discussion on corpus representativeness is nevertheless important but the issue is beyond the scope of this paper. Next, we introduce two base measures for capturing the statistical evidence based on the cross-domain and intra-domain distribution:

- **Intra-domain distribution** of term candidates and context words are employed to compute the basic *domain prevalence (DP)*. DP measures the extent of term usage in the target domain.
- **Cross-domain distributional behavior** of term candidates and context words are employed to compute the *domain tendency (DT)*. DT measures the extent of inclination of term usage toward the target domain.

A high DP means that the term is frequently encountered (i.e. prevalent) in the target domain. It is a sign of high domain relevance if and only if the frequent usage of that term is exclusive to the target domain (i.e. high DT). The three types of linguistic evidences, which are essential to the estimation of termhood are quantified using new measures derived from the prevalence and tendency measures described above. The linguistic evidences and the corresponding derived measures are:

- **Candidate evidence**, in the form of *discriminative weight (DW)*, is measured as the product of the domain tendency and the domain prevalence of term candidates. This evidence constitutes the first step in an attempt to isolate domain-relevant from general candidates.
- **Modifier evidence**, in the form of *modifier factor (MF)* is obtained by computing DT using the cross-domain cumulative frequency of modifiers of complex terms.
- **Contextual evidence**, in the form of *average contextual discriminative weight (ACDW)*, is computed using the cumulative DW of context words, scaled according to their semantic relatedness with the corresponding term candidates. $ACDW$ is later adjusted with respect to the DW of the term candidate to obtain the *adjusted contextual contribution (ACC)* to reflect the reliability of the contextual evidence.

Given that we have a list of term candidates (both simple and complex) $TC = \{a_i, ..., a_n\}$, the aim of this mechanism is to assign scores to term candidates to assist in the ranking and identification of the most suitable candidates as terms $t \in T$. Each complex term, a will comprise of a head a^h and modifiers $m \in M(a)$. Each term candidate is assigned a weight depending on its type (i.e. simple or complex). We refer to this new CW-inspired weight as *domain prevalence (DP)* because of its ability to capture the extent of occurrences of terms in the target domain. If a is a simple term, its DP is defined as:

$$DP(a) = \log_{10}(f_{ad} + 10) \log_{10}\left(\frac{F_{TC}}{f_{ad} + f_{a\bar{d}}} + 10\right)$$

where $F_{TC} = \sum_j f_{jd} + \sum_j f_{j\bar{d}}$ is the sum of the frequencies of occurrences of all $a \in TC$ in both domain and contrastive corpora, while f_{ad} and $f_{a\bar{d}}$ are the frequencies of occurrences of a in the domain corpus and contrastive corpora, respectively. If the term candidate is complex, we define its DP as:

$$DP(a) = \log_{10}(f_{ad} + 10) DP(a^h) MF(a)$$

Please note the use of the DP of the head a^h for the computation of DP for complex terms. Motivated by CW, rarity of appearances of complex terms does not allow a proper computation of the weight. Nonetheless, we have noticed from the original CW that the direct multiplication of f_{ad} of extremely common and general complex terms will distort the weights and give a false impression of their importance in the domain. Consequently, unlike the original CW, we add a constant 10 and later log the domain frequency of complex terms. This modification has shown to eliminate the biased ranking of CW as demonstrated in Section 4. Besides the addition of the constant 10 to f_{ad} prior to log, we introduce another new measure called *modifier factor (MF)* to:

- provide relevant complex terms with higher weights than their head;
- penalise those potentially deceiving domain-unrelated complex terms that have domain-related heads. For example, the head *"virus"* will yield high weight in the *"technology"* domain. If we did not take into consideration the fact that the head was modified by *"H5N1"* to form the complex term *"H5N1 virus"*, the complex term makes its way into the list of terms for the *"technology"* domain; and
- compensate for the low weight of domain-related complex terms that have domain-unrelated heads. For example, upon looking at the head *"account"*, one would be tempted to immediately rule the corresponding complex term out as irrelevant for the *"technology"* domain. With MF, we can take into consideration the modifiers *"Google"* and *"Gmail"* to safely assign higher weights to the complex term *"Google Gmail account"*.

The MF of a complex term a is measured using the cumulative domain frequency and cumulative contrastive frequency of modifiers which also happen to be term candidates, $m \in M_a \cap TC$. Formally, the MF of a complex term a is defined as:

$$MF(a) = \log_2\left(\frac{\sum_{m \in M_a \cap TC} f_{md} + 1}{\sum_{m \in M_a \cap TC} f_{m\bar{d}} + 1} + 1\right)$$

MF is actually a derived measure modelled after our second new base measure *domain tendency (DT)*. The only difference between the two is that MF works with modifiers while DT works with the entire term candidate, both simple and complex. MF and DT are two powerful discriminating measures that help to differentiate between candidates which are truly relevant to the target domain from generally-prevalent candidates. Formally, we can determine the extent of the inclination of the usage of term candidate a for domain and non-domain purposes through:

$$DT(a) = \log_2\left(\frac{f_{ad} + 1}{f_{a\bar{d}} + 1} + 1\right)$$

where f_{ad} is the frequency of occurrences of a in the domain corpus, while $f_{a\bar{d}}$ is the frequency of occurrences of a in the contrastive corpora. If term candidate a is equally common in both domain and non-domain (i.e. contrastive domain), $DT = 1$. If the usage of a is more inclined toward the target domain, $f_{ad} > f_{a\bar{d}}$, then $DT > 1$, and $DT < 1$ otherwise. Next, this new base measure DT together with DP will contribute to a new weight known as *discriminative weight (DW)*. A term that appears frequently in the target domain (i.e. high DP) will still have a low overall weight DW if its usage is more inclined toward the contrastive domains (i.e. low DT). DW is simply the product of DP and the corresponding DT of the term candidate:

$$DW(a) = DP(a) DT(a)$$

Assuming that term candidate a has the set of context words C_a, the *average contextual discriminative weight (ACDW)* is defined as:

$$ACDW(a) = \frac{\sum_{c \in C_a} DW(c) sim(a, c)}{|C_a|}$$

where $sim(a, c) = 1 - NGD(a, c)\theta$, $NGD(a, c)$ is the *Normalized Google Distance* (Cilibrasi & Vitanyi 2007) between term candidate a and c, and θ is a constant for scaling the distance value of NGD. The $ACDW$ weight allows us to take into consideration the company a term candidate keeps. Nonetheless, not all context words describe or are related to the terms they appear with. Unlike other applications that can completely rely on contextual information, we cannot allow $ACDW$ to have direct contribution to the overall termhood. In this regard, we employ two measures to adjust the contribution of the contextual weight to the overall termhood. First, we

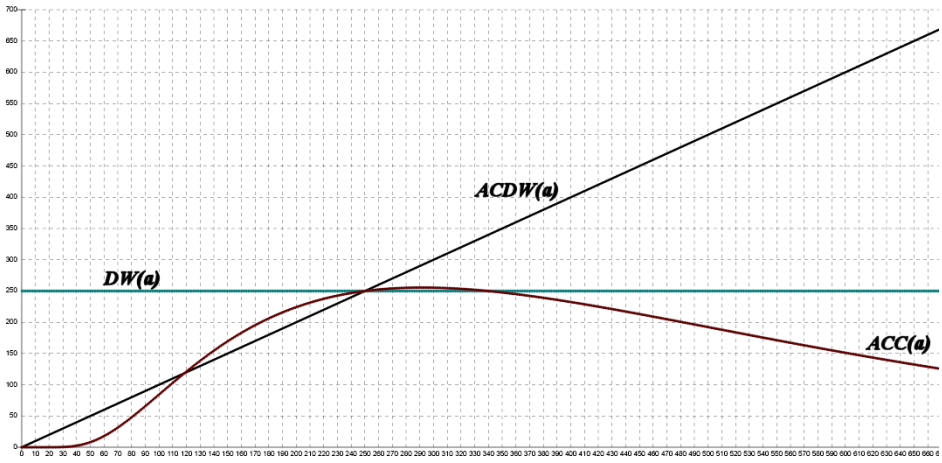


Figure 1: ACC experiences an increase for $ACDW < DW$. The distribution of ACC is reflected at the meeting point of DW and $ACDW$ and experiences subsequent decrease more or less inversely proportional to $ACDW$.

utilise NGD to quantify the relatedness between a term and its context words during the computation of $ACDW$. So far, NGD has only been successfully adopted for use with clustering (Wong et al. 2007*b*). Contextual words which are more related to the term candidate will have higher contribution to the overall $ACDW$. NGD is at present the most ideal solution for the problems introduced by the use of static and restricted semantic resources faced by many researchers. Secondly, we adjust $ACDW$ according to its ratio with the corresponding DW to produce the *adjusted contextual contribution (ACC)* as shown in Figure 1. Formally, we define ACC as:

$$ACC(a) = ACDW(a)\frac{e^{\left(1-\frac{ACDW(a)+1}{DW(a)+1}\right)}e^{\left(1-\frac{DW(a)+1}{ACDW(a)+1}\right)}}{\log_2 \frac{ACDW(a)+1}{DW(a)+1}+1}$$

In the end, we define the final weight known as *Termhood (TH)* for each term candidate as:

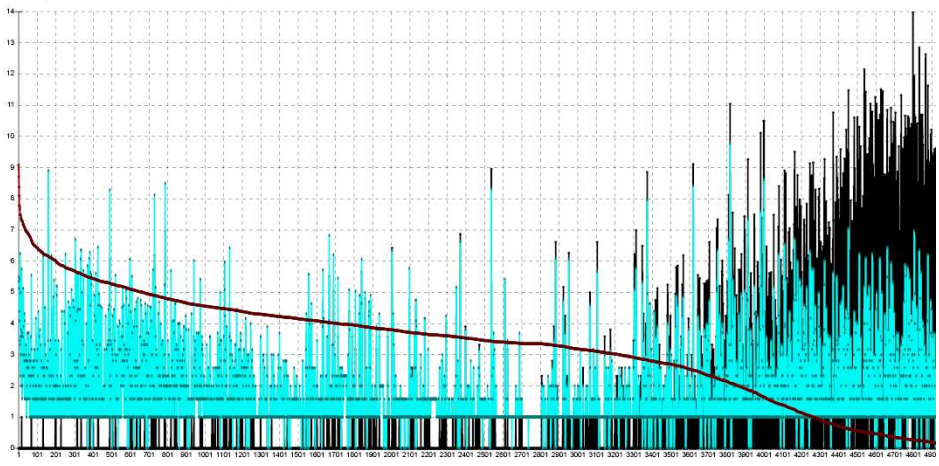
$$TH(a) = DW(a) + ACC(a) \tag{8}$$

4 Experiments

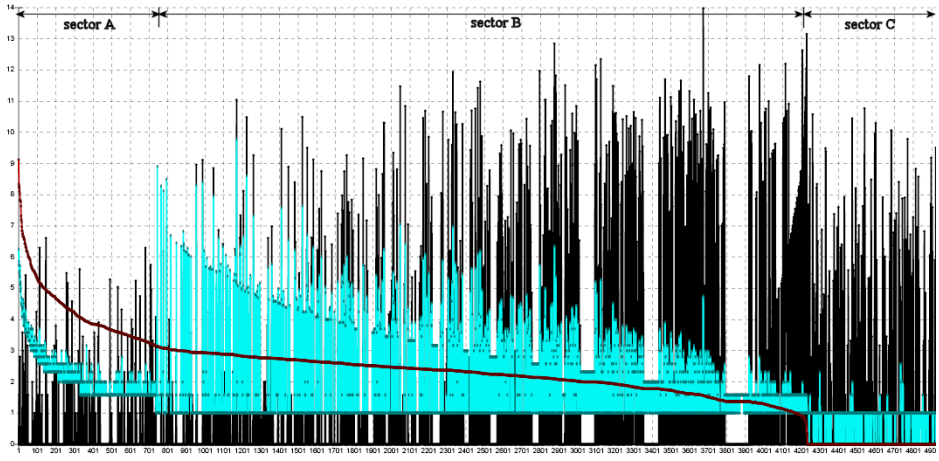
We employ two text sources: a domain corpus containing 24 documents (with 51,289 word count) in *"liver cancer"* from `BioMedCentral.com`, and a contrastive corpora consisting of 11,115 news articles (with 4,378,210 word count) in various domains such as *"technology"*, *"business"*, *"politics"* and *"sports"*. The news articles are gathered from `Reuters.com`, `CNet.com` and `ABC.com` between the period of February 2006 and April 2007. The implementation of *Contrastive Weight (CW)*, *NCvalue (NCV)* and *Termhood (TH)* are in accordance to Equation 1 and 2, 6, and 8 respectively. The source of the term candidates and context words is a list of 6,000 instantiated subcategorisation frames (Wong 2005) extracted from the *"liver cancer"* domain corpus. By selecting only noun phrases from the first and second arguments of the frames, we obtained 5,156 term candidates.

Like other evaluations in ontology learning, determining the quality of the extracted terms is a lengthy and challenging process due to the subjective nature of the task and the efforts required. The quantitative evaluations required for a numerical analysis and comparative study of CW, NCV and TH is beyond the scope of this paper. Nonetheless, we have decided to assess CW, NCV and TH using an equally effective method, namely, the analysis of the frequency distributions. The role of term frequency as the main source of termhood evidence makes this evaluation method (i.e. analysis of frequency distribution) highly applicable. We will discuss the reasons and ramifications of the various interesting characteristics displayed in the graphs, with reference to the related measures. For this experiment, the performance of the termhood measures are judged solely based on their ability to provide higher ranks to candidates that have domain frequencies higher than contrastive frequencies. Ideally, candidates with higher domain frequencies should congregate along the left end of the x-axis in the graphs while those with higher contrastive frequencies are pushed to the far right of the x-axis. The graphs are based on the logarithmic scale to accommodate the frequencies which can become very large.

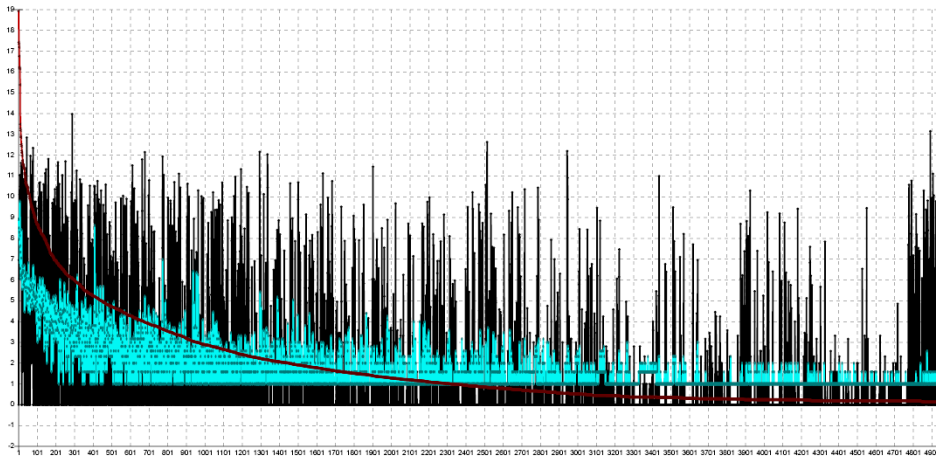
Firstly, the frequency distribution of the candidates sorted according to CW displays an interesting trend that reflects the different treatment given to simple and complex terms. Despite exhibiting some characteristics of being able to discriminate and identify domain-relevant candidates but certain peculiarities in Figure 2(b) call for deeper analysis. If we recall what has been discussed regarding Equations 1 and 2, we noticed that sector A in Figure 2(b) is the direct result from the use of heads to compute the contrastive weights of complex terms. The listing of the ranked terms which corresponds to Figure 2(b) clearly shows that all candidates in sector A are complex terms. Despite the much lower domain frequency of complex terms f_{ad} in sector A, these terms are ranked among the highest by CW due to the high domain frequency of their corresponding heads $f_{a^h d}$. In addition, instead of $\log_{10} f_{ad}$, the direct multiplication of f_{ad} to the contrastive weight of the head of complex terms $CW(a^h)$ further increases the contrastive weight of the complex terms $CW(a)$. In sector B, the distributions behave as intended due to the design of the contrastive weight that places emphasis on the domain frequency. This sector contains a fair mix of simple and complex terms. In this sector, one will notice that the candidates are ordered from left to right in such a way that their f_{ad} decreases as $f_{a\bar{d}}$ increases. A point worth noting about this section is the somewhat periodic clusters of candidates



(a) Candidates ranked according to the scores by TH



(b) Candidates ranked according to the scores by CW



(c) Candidates ranked according to the scores by NCV

Figure 2: This graph shows the frequency distributions of 5,156 candidates ranked in descending order according to the scores assigned by the respective measures. The black, oscillating line represents the frequency of the candidates in the contrastive corpora $f_{\bar{d}}$ while the greyish, oscillating line is the domain frequency f_d. The single dark line that spans from the left to the right of the graph in descending order represents the scores assigned by the respective measures.

characterised by the sudden drop and surge in frequency. This phenomena can be explained by the use of heads for the computation of weights for complex terms. Due to the increasingly less significant contribution of f_{ad} to $CW(a)$ in sector B, complex terms having similar heads are inevitably grouped together. Once in sector C, the contrastive weights of term candidates $CW(a)$ drop to zero. Most terms in this sector are single-word and have $f_{ad} = 1$. Without the support of $CW(a^h)$, the single-occurrence simple terms in this sector are weighted 0.

Secondly, Figure 2(c) illustrates why NCV, the version discussed in Equation 6, is not suitable for recognizing terms for domain-specific uses. The weight helps to rank those candidates in the order of decreasing domain frequency f_d, regardless of the candidates' prevalence and tendency in other domains. Firstly, looking at the list of ranked terms, one would notice that those candidates which are highly ranked are accompanied by more context words. As formulated in Equation 5, more context words will contribute to an increasingly higher contextual weight. As we have pointed out repeatedly, contextual evidence does contribute to the determination of termhood but indiscriminate use will results in undesirable effects. Secondly, the highly-ranked candidates are also those that have extremely high $Cvalue$ as defined in Equation 3. In other words, a candidate that demonstrates better independence from the longer term candidates which it is part of will have higher rank. This results in the emphasis and high ranking for mostly simple terms or shorter complex terms that are extremely popular and at the same, commonly used to construct other longer complex terms.

Lastly, Figure 2(a) shows the discriminating power of our new scoring and ranking mechanism using termhood TH. Based on the distribution of f_{ad} and $f_{a\bar{d}}$, one will notice that candidates with high f_{ad} and near-zero $f_{a\bar{d}}$ are highly ranked. This trend progresses until all candidates with high $f_{a\bar{d}}$ and comparatively lower f_{ad} are pushed to the far right end. It is worth pointing out that the candidates in Figure 2(a) are not ordered in a smooth descending flow according to their domain frequencies f_{ad}. This can be explained by the diversified evidences that our new measures rely on. For example, using our measures, some candidates with high f_{ad} may appear in lower ranks than those with lower f_{ad}. This is in sharp contrast to the other two existing weights namely CW and NCV where one can observe from sector B in Figure 2(b) and the whole of Figure 2(c) that, despite minor fluctuations, the ranking of the candidates are largely influenced by frequencies.

In addition to the absence of evaluation methods and datasets, the subjective nature of termhood assessment makes the tasks of objectively evaluating and comparing our new measure with existing approaches a problem domain by itself. In the words of (Damle & Uren 2005), *"Unfortunately, there is no objective evaluation method reported in the literature for term extraction..."*. However, our initial experiments using the frequency distributions alone have revealed three positive traits present in our new weight TH that will assist in improving the quality of automatic term recognition:

- Unlike CW and NCV, the diversification of evidence to cover both statistical and linguistic has allowed TH to be all-inclusive and not purely dependent on frequency. Despite the subjectivity of the notion termhood, such diversification provides opportunity for more accurate approximation to reflect the *"actual"* termhood.
- Unlike NCV, the increase in the number of context words does not indiscriminately propel our new weight TH. The selective use of contextual evidence allows its contribution in TH to be adjusted depending on the relationship between the context words and the term candidates.
- In NCV, shorter term candidates have better chances of gaining higher rank, while complex term candidates always top the list in CW. The size of candidates should not have any influence on their weights. Unlike the two measures which impose baseless, intrinsic prejudice on the candidates during the scoring process, both simple and complex term candidates are given equal opportunities using TH, based purely on observable evidences.

5 Conclusion and Future Work

In this paper, we have presented a new mechanism consisting of a series of base and derived measures for recognising terms. The base measures, namely, *domain prevalence (DP)* and *domain tendency (DT)* capture the statistical evidence that appear in the form of intra-domain and cross-domain term distributional behavior. Using these base measures, four additional measures, namely *discriminative weight (DW)*, *modifier factor (MF)*, *average contextual discriminative weight (ACDW)*, and *adjusted contextual contribution (ACC)* were derived to quantify linguistic evidences in the form of candidates, modifiers and context words. Together, these base and derived measures contribute to the computation of a final weight known as *Termhood (TH)* that is used for the ranking of candidates and selection of terms. Our initial experiments revealed three advantages exhibited by our new weight TH. Such revelation has prompted us to plan for future works to evaluate our new mechanism using larger datasets and established measures such as precision and recall to enable numerical analysis and comparative study. In addition, to demonstrate the applicability of our new approach in real-word, domain-specific scenarios, we intend to have domain experts assisting in future evaluations.

Acknowledgement

This research was supported by the Australian Endeavour International Postgraduate Research Scholarship, and the Research Grant 2006 by the University of Western Australia.

References

Basili, R., Moschitti, A., Pazienza, M. & Zanzotto, F. (2001), A contrastive approach to term extraction, *in* 'Proceedings of the 4th Terminology and Artificial Intelligence Conference (TIA)', France.

Basili, R., Pazienza, M. & Zanzotto, F. (2001), Modelling syntactic context in automatic term extraction, *in* 'Proceedings of the International Conference on Recent Advances in Natural Language Processing', Bulgaria.

Bookstein, A. & Swanson, D. (1974), 'Probabilistic models for automatic indexing', *Journal of the American Society for Information Science* **25**(5), 312–8.

Cabre-Castellvi, T., Estopa, R. & Vivaldi-Palatresi, J. (2001), Automatic term detection: A review of current systems, *in* D. Bourigault, C. Jacquemin & M. LHomme, eds, 'Recent Advances in Computational Terminology', John Benjamins.

Church, K. & Gale, W. (1995), Inverse document frequency (idf): A measure of deviations from poisson, *in* 'Proceedings of the ACL 3rd Workshop on Very Large Corpora'.

Cilibrasi, R. & Vitanyi, P. (2007), 'The google similarity distance', *IEEE Transactions on Knowledge and Data Engineering* **19**(3), 370–383.

Damle, D. & Uren, V. (2005), Extracting significant words from corpora for ontology extraction, *in* 'Proceedings of the 3rd International Conference on Knowledge Capture', Alberta, Canada.

Frantzi, K. & Ananiadou, S. (1997), Automatic term recognition using contextual cues, *in* 'Proceedings of the IJCAI Workshop on Multilinguality in Software Industry: the AI Contribution', Japan.

Fuhr, N. (1986), Two models of retrieval with probabilistic indexing, *in* 'Proceedings of the 9th ACM SIGIR International Conference on Research and Development in Information Retrieval'.

Fuhr, N. (1992), 'Probabilistic models in information retrieval', *The Computer Journal* **35**(3), 243–255.

Gomez-Perez, A. & Manzano-Macho, D. (2003), A survey of ontology learning methods and techniques, Deliverable 1.5, OntoWeb Consortium.

Grefenstette, G. (1994), Explorations in automatic thesaurus discovery, Kluwer Academic Publishers, MA, USA.

Hippisley, A., Cheng, D. & Ahmad, K. (2005), 'The head-modifier principle and multilingual term extraction', *Natural Language Engineering* **11**(2), 129–157.

Jones, K., Walker, S. & Robertson, S. (1998), 'A probabilistic model of information retrieval: Development and status', *Information Processing and Management* **36**(6), 809–840.

Kageura, K. & Umino, B. (1996), 'Methods of automatic term recognition: A review', *Terminology* **3**(2), 259–289.

Kit, C. (2002), Corpus tools for retrieving and deriving termhood evidence, *in* 'Proceedings of the 5th East Asia Forum of Terminology', Haikou, China.

LeMoigno, S., Charlet, J., Bourigault, D., Degoulet, P. & Jaulent, M. (2002), Terminology extraction from text to build an ontology in surgical intensive care, *in* 'Proceedings of the ECAI Workshop on Machine Learning and Natural Language Processing for Ontology Engine'.

Lewis, D. (1998), Naive (bayes) at forty: The independence assumption in information retrieval, *in* 'Proceedings of the 10th European Conference on Machine Learning'.

Manning, C. & Schutze, H. (1999), Foundations of statistical natural language processing, MIT Press, MA, USA.

Maynard, D. & Ananiadou, S. (1999), Term extraction using a similarity-based approach, *in* 'Recent Advances in Computational Terminology', John Benjamins.

Maynard, D. & Ananiadou, S. (2000), Identifying terms by their family and friends, *in* 'Proceedings of the 18th International Conference on Computational Linguistics', Germany.

Salton, G. & Buckley, C. (1988), 'Term-weighting approaches in automatic text retrieval', *Information Processing & Management* **24**(5), 513–523.

Teevan, J. & Karger, D. (2003), Empirical development of an exponential probabilistic model for text retrieval: Using textual analysis to build a better model, *in* 'Proceedings of the 26th ACM SIGIR International Conference on Research and Development in Information Retrieval'.

Wong, W. (2005), Practical approach to knowledge-based question answering with natural language understanding and advanced reasoning, Master's thesis, National Technical University College of Malaysia, arXiv:cs.CL/0707.3559.

Wong, W., Liu, W. & Bennamoun, M. (2007*a*), Determining the unithood of word sequences using mutual information and independence measure, *in* 'Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics (PACLING)', Melbourne, Australia.

Wong, W., Liu, W. & Bennamoun, M. (2007*b*), 'Tree-traversing ant algorithm for term clustering based on featureless similarities', Journal on Data Mining and Knowledge Discovery, doi: 10.1007/s10618-007-0073-y.

Determining Termhood for Learning Domain Ontologies in a Probabilistic Framework

Wilson Wong, Wei Liu and Mohammed Bennamoun

School of Computer Science and Software Engineering
University of Western Australia
Crawley WA 6009
{wilson,wei,bennamou}@csse.uwa.edu.au

Abstract

Many existing techniques for term extraction are heuristically-motivated and criticised as ad-hoc. The definitions and assumptions critical to set the boundary for the effectiveness of the techniques are often implicit and unclear. Here we present a probabilistic framework for measuring termhood to address the lack of mathematical foundation in existing techniques.

1 Introduction

Term extraction, also known as *automatic term recognition* and *terminology mining*, is essential to many text mining applications such as ontology learning. The aim of term extraction is to identify content-bearing lexical units (i.e. terms) from text, which can either be individual or group of words. Term extraction consists of two fundamental steps: 1) identifying term candidates from text, and 2) filtering through the candidates to separate terms from non-terms. The first step involves the determination of *unithood*, which concerns with whether sequences of words can be combined to form stable lexical units (Wong, Liu & Bennamoun 2007*b*). On the other hand, *termhood* characterises the second step, which is to determine to what extent a stable lexical unit is related to a certain domain-specific concept. This paper focuses on developing a probabilistic framework for measuring termhood.

The tasks of termhood determination is different from the two well-known problems of named-entity recognition and information retrieval. The biggest dissimilarity between named-entity recognition and termhood determination is that the former is a deterministic problem of classification whereas the latter involves the subjective measurement of relevance and ranking. Hence, unlike the availability of various platforms for the evaluation of named-entity recognition such as the *BioCreAtIvE Task 1* (Hirschman, Yeh, Blaschke & Valencia 2005) and the *Message Understanding Conference (MUC)* (Chinchor, Lewis & Hirschman 1993), determining the performance of term extraction remains an extremely subjective problem domain. While appearing more similar to information retrieval in that both involves relevance ranking, the determination of termhood does have its unique requirements in processing text. Most importantly, the determination of termhood does not have user queries as evidences for deciding on relevance. As such, the only source of evidence for determining termhood is a set of heuristically-motivated term characteristics.

The surveys by (Cabre-Castellvi, Estopa & Vivaldi-Palatresi 2001) and (Kit 2002) show that existing term extraction methods rely on the above-mentioned hypothetical term characteristics to devise empirical measures for termhood. Consequently, these methods are often criticised for their lack of theorisation and mathematical validity. Such criticisms become obvious when one poses simple but crucial questions on the ways certain measures are derived, for example, *"Why taking different bases for logarithm?"*, or *"Why combining two weights using addition and not multiplication?"*.

To address the lack of proper theorisation in term extraction, we present a probabilistic framework for scoring and ranking term candidates to measure termhood. This measure is founded on Bayes Theorem and the Zipf-Mandelbrot model (Tullo & Hurford 2003) for computing the evidences. Our new measure is adaptable in that new or obsolete evidences can be added or removed based on different requirements of the system. Section 2 summarises some prominent methods in term extraction. Section 3 develops the probabilistic framework and so-derived measure of termhood. Section 4 presents the results of a comparative study and the paper concludes in Section 5 with an outlook to future works.

2 Related Work

Surveys (Cabre-Castellvi et al. 2001, Kageura & Umino 1996) on term extraction approaches revealed that most of the existing methods were based on ad-hoc statistical measures combined with linguistics information. These measures are usually put together using term or document frequency, and are modified as per need as the observation of immediate results progresses. As such, the significance of the different weights that compose the measures usually assume an empirical viewpoint. Obviously, such methods are at most inspired by, but not derived from formal models. Many critics claim that such methods are unfounded and the results that were reported using these methods are merely coincidental. In the words of (Kageura & Umino 1996), *"As for the validity of statistical methods or models, we have seen that many use intuitively reasonable by mathematically unfounded measures."*.

Existing measures based on formal probabilistic models for determining the relevance of words with respect to certain topics or documents are mainly studied within the realm of document retrieval and automatic indexing. In probabilistic indexing, one of the first few detailed quantitative models was proposed by (Bookstein & Swanson 1974). In this model, the differences in the distributional behavior of words

is employed as a guide to determine if a word should be considered as an index term. This model is derived from the fact that single Poisson distribution is only a good fit for functional words while content words tend to deviate from it (vanRijsbergen 1979, Church & Gale 1995, Manning & Schutze 1999). Such variation from the Poisson distribution or colloquially known *"non-poissonness"* can then be employed as a predictor of whether a lexical unit is a content word or not, and hence as an indicator of possible termhood.

An even larger collection of literature on probabilistic models can be found in a related area of document retrieval. The simplest of all the retrieval models is the binary independence model (Fuhr 1986, Lewis 1998). As with all other retrieval models, the binary independence model is designed to estimate the probability that a document j is considered as relevant given a specific query k. Let $T = \{t_1, ...t_n\}$ be the set of terms in the collection of documents (i.e. corpus). We can then represent the set of terms T_j occurring in document j as a binary vector $v_j = \{x_1, ..., x_n\}$ where $x_i = 1$ if $t_i \in T_j$ and $x_i = 0$ otherwise. This way, the odds of document j, represented by a binary vector v_j being relevant to query k can be computed as (Fuhr 1992):

$$O(R|k, v_j) = \frac{P(R|k, v_j)}{P(\bar{R}|k, v_j)} = \frac{P(R|k)}{P(\bar{R}|k)} \frac{P(v_j|R, k)}{P(v_j|\bar{R}, k)}$$

and based on the assumption of independence between the presence and absence of terms,

$$\frac{P(v_j|R, k)}{P(v_j|\bar{R}, k)} = \prod_{i=1}^{n} \frac{P(x_i|R, k)}{P(x_i|\bar{R}, k)}$$

Other more advanced models that take into considerations other factors such as term frequency, document frequency and document length have also been proposed (Jones, Walker & Robertson 1998).

(Basili, Moschitti, Pazienza & Zanzotto 2001) proposed a *TF-IDF* inspired measure for assigning terms with weights quantifying their specificity to the target domain. A *Contrastive Weight* is defined for a simple term candidate a in a target domain d as:

$$CW(a) = \log f_{ad} \left(\log \frac{\sum_j \sum_i f_{ij}}{\sum_j f_{aj}} \right) \quad (1)$$

where f_{ad} is the frequency of the simple term candidate a in the target domain d, $\sum_j \sum_i f_{ij}$ is the sum of the frequencies of all term candidates in all domain corpora, and $\sum_j f_{aj}$ is the sum of the frequencies of the term candidate a in all domain corpora. For complex term candidates, the frequencies of their heads are utilised to compute their weights:

$$CW(a) = f_{ad} CW(a^h) \quad (2)$$

where f_{ad} is the frequency of the complex term candidate a in the target domain d, and $CW(a^h)$ is the contrastive weight for the head, a^h of the complex term candidate.

There is also the use of contextual evidence to assist in the identification of terms. One of the works is *NCvalue* by (Frantzi & Ananiadou 1997). Given that TC is the set of all term candidates and c is a noun, verb or adjective (i.e. context words) appearing with the term candidates, $weight(c)$ is defined as:

$$weight(c) = 0.5 \left(\frac{|TC_c|}{|TC|} + \frac{\sum_{e \in TC_c} f_e}{f_c} \right)$$

where TC_c is the set of term candidates that have c as a context word, $\sum_{e \in TC_c} f_e$ is the sum of the frequencies of term candidates that appear with c, and f_c is the frequency of c in the corpus. After calculating the weights for all possible context words, the sum of the weights of context words appearing with each term candidate can be obtained. Formally, for each term candidate a that has a set of accompanying context words C_a, the cumulative context weight is defined as:

$$cweight(a) = \sum_{c \in C_a} weight(c) + 1$$

Finally, the *NCvalue* for a term candidate a is defined as:

$$NCvalue(a) = \frac{1}{\log F} Cvalue(a) cweight(a) \quad (3)$$

where F is the size of the corpus in terms of the number of words. $CValue(a)$ is given by:

$$Cvalue(a) = \begin{cases} \log_2 |a| f_a & \text{if } |a| = g \\ \log_2 |a| (f_a - \frac{\sum_{l \in L_a} f_l}{|L_a|}) & \text{otherwise} \end{cases}$$

where $|a|$ is the number of words in a, L_a is the set of potential longer term candidates that contain a, g is the longest n-gram considered, and f_a is frequency of occurrences of a in the corpus.

(Wong, Liu & Bennamoun 2007*a*) proposed a *Discriminative Weight (DW)* as part of a scoring and ranking scheme called *Termhood (TH)*. *DW* is a product of *Domain Prevalence (DP)* and *Domain Tendency (DT)*. If a is a simple term candidate, the domain prevalence DP is defined as:

$$DP(a) = \log_{10}(f_{ad} + 10) \log_{10} \left(\frac{F_{TC}}{f_{ad} + f_{a\bar{d}}} + 10 \right)$$

where $F_{TC} = \sum_j f_{jd} + \sum_j f_{j\bar{d}}$ is the sum of the frequencies of occurrences of all term candidates $j \in TC$ in both domain and contrastive corpora, while f_{ad} and $f_{a\bar{d}}$ are the frequencies of occurrences of a in the domain corpus and contrastive corpora, respectively. If the term candidate a is complex, DP is defined as:

$$DP(a) = \log_{10}(f_{ad} + 10) DP(a^h) MF(a)$$

DT is employed to determine the extent to which term candidate a is used for domain purposes. It is defined as:

$$DT(a) = \log_2 \left(\frac{f_{ad} + 1}{f_{a\bar{d}} + 1} + 1 \right)$$

where f_{ad} is the frequency of occurrences of a in the domain corpus, while $f_{a\bar{d}}$ is the frequency of occurrences of a in the contrastive corpora. The adjustment of contextual evidence is also introduced to ensure that the weights of non-terms are not inflated by domain-relevant context. The *Adjusted Contextual Contribution (ACC)* is defined as:

$$ACC(a) = ACDW(a) \frac{e^{\left(1 - \frac{ACDW(a)+1}{DW(a)+1}\right)} e^{\left(1 - \frac{DW(a)+1}{ACDW(a)+1}\right)}}{\log_2 \frac{ACDW(a)+1}{DW(a)+1} + 1}$$

$ACDW$ is simply the average DW of the context words of candidate a adjusted according to the context's relatedness to a:

$$ACDW(a) = \frac{\sum_{c \in C_a} DW(c) sim(a, c)}{|C_a|}$$

where $sim(a,c) = 1 - NGD(a,c)\theta$, $NGD(a,c)$ is the *Normalized Google Distance* (Cilibrasi & Vitanyi 2007) between term candidate a and c, and θ is a constant for scaling the distance value of NGD. The use of NGD overcomes the problems associated with the use of semantic information. The final weight of each term candidate a is given by:

$$TH(a) = DW(a) + ACC(a) \tag{4}$$

3 A Probabilistic Framework for Determining Termhood

Terms are lexical realisations of their abstract counterparts (i.e. concepts) that are relevant to some domains of interest. As such, the aim of determining termhood in term extraction is to identify terms that are relevant to the same domain as are the concepts they represent. As pointed out before, the only source of evidence in term extraction is the characteristics of terms embedded in the domain corpora. From here on, notation d is used to denote the domain corpus and the target domain it represents, and $\bar{d}$ denotes the contrastive corpora, that is, all corpora other than d. In probabilistic terms, we can describe our aim of termhood determination as:

Aim 1 *What is the probability of candidate a being relevant to domain d, given the evidence candidate a has?*

3.1 A General Probabilistic Model

Definition 1 outlines the primary characteristics of terms (Kageura & Umino 1996). These characteristics are considered as ideal because they rarely exist in real-world situations as we will discuss later.

Definition 1 *The primary characteristics of terms in ideal settings:*

1. A term should not have any synonyms.

2. The meaning of a term is independent of context.

3. The meaning of a term should be precise and related directly to a concept.

In addition, there are other characteristics that are equally important in the determination of termhood. Some of these characteristics are inherent general properties of words. This list is not a standard and by no means exhaustive. They are:

Definition 2 *Extended characteristics of terms*

1. Terms are properties of domains, not documents (Basili et al. 2001).

2. Terms tend to clump together (Bookstein, Klein & Raita 1998) the same way as content-bearing words do (Zipf 1949).

3. Terms of longer length are rare in a corpus since the usage of phrases of shorter length are more predominant (Zipf 1935).

4. Simple terms are often ambiguous and modifiers are required to reduce the number of possible interpretations. Hence complex terms are usually preferred in terminology (Frantzi & Ananiadou 1997).

Definition 1 states that a term is unambiguously relevant to a domain. For example, when one encounter the term *"bridge"*, there should be one and exactly one meaning: *"a device that connects multiple network segments at the data link layer"*. As such, an *ideal term* cannot be relevant to more than one domain. Therefore, determining termhood is simplified to just measuring the extent to which a term candidate is relevant to a domain regardless of its relevance to other domains. Aim 1 can thus be formulated as a conditional probability between two events using Bayes Theorem.

$$P(R_1|A) = \frac{P(A|R_1)P(R_1)}{P(A)} \tag{5}$$

where R_1 is the event that a is relevant to domain d and A is the event that a is a candidate term supported by an evidence vector $V =< E_1, ..., E_m >$. The details of the evidence vector will be presented in Section 3.2. $P(R_1|A)$ is the posterior probability of candidate a being relevant to d given the evidence vector V. $P(R_1)$ is the prior probability of candidate a being relevant without any evidence, and $P(A)$ is the prior probability of a being a candidate with evidence V. As we shall see later, these two prior probabilities will be immaterial in the final computation of the weights for the candidates. Since the reliability of the evidences of terms is dependent on the representativeness of the corpus, the following Assumptions 1 and 2 are also necessary:

Assumption 1 *Corpus d is a balanced, unbiased and randomised sample of the population text representing the corresponding domain.*

Determining the representativeness of a corpus is important but beyond the scope of this paper.

Assumption 2 *Contrastive corpora $\bar{d}$ is the set of balanced, unbiased and randomised sample of the population text representing approximately all major domains other than d.*

Given Assumption 2, let us define R_2 as the event that candidate a is relevant to other domains $\bar{d}$. Following this and based on Definition 1, we have $P(R_1 \cap R_2) = 0$. In other words, R_1 and R_2 are mutually exclusive in ideal settings. Ignoring the fact that a term may appear in certain domains by chance, any candidate a can either be relevant to d or to $\bar{d}$, but not both.

Unfortunately, according to (Loukachevitch & Dobrov 2004), *"An impregnable barrier between words of a general language and terminologies does not exist."*. For example, the word *"bridge"* has multiple meanings and is relevant to more than one domain. In other words, $P(R_1 \cap R_2)$ is rarely 0 in reality. However, words like *"bridge"* are regarded as poor choice of terms because they are simple terms and inherently ambiguous as defined in Definition 2. Instead, a better term for denoting the concept which the candidate *"bridge"* attempts to represent would be *"network bridge"*. This is usually the case in writings where authors first introduce new concepts using unambiguous complex terms and later, reiterating the same concepts with shorter terms. As such, we assume that:

Assumption 3 *Each concept represented using a polysemous simple term in corpus has a corresponding unambiguous complex term representation occurring in the same corpus.*

From Assumption 3, since all important concepts of a domain have unambiguous manifestation in the corpus, the possibility of the ambiguous counterparts being inappropriately ranked during our termhood measurement will have no effect on the overall term extraction output. As such, polysemous simple terms can be considered as insignificant in our determination of termhood. Based on Definition 1 and Assumption 3, the probability of relevance of candidate a to both d and $\bar{d}$ is approximately 0, i.e.

$P(R_1 \cap R_2) \approx 0$. Following this, we have $P(R_1 \cup R_2) = P(R_1)+P(R_2) \approx 1$. This approximation of the sum of the probability of relevance without evidence can be extended to the conditional probability of relevance given evidence vector V:

$$P(R_1|A) + P(R_2|A) \approx 1 \tag{6}$$

without violating the axioms of probability.

Since $P(R_1 \cap R_2)$ only approximates to 0 in reality, determining the probability of relevance of candidate a to d alone may not be enough. We need to calculate the odds of relevance to demonstrate that candidate a is more relevant to d than to $\bar{d}$:

Aim 2 What are the odds of candidate a being relevant to d given the evidence candidate a has?

Since $Odds = P/(1-P)$, we can obtain the odds of relevance given the evidence candidate a has by applying $(1 - P(R_1|A))^{-1}$ to Equation 5:

$$\frac{P(R_1|A)}{1 - P(R_1|A)} = \frac{P(A|R_1)P(R_1)}{P(A)(1 - P(R_1|A))} \tag{7}$$

and since $1 - P(R_1|A) \approx P(R_2|A)$ from Equation 6, and by applying the multiplication rule $P(R_2|A)P(A) = P(A|R_2)P(R_2)$ to the left side of Equation 7, we have:

$$\frac{P(R_1|A)}{P(R_2|A)} = \frac{P(A|R_1)}{P(A|R_2)} \frac{P(R_1)}{P(R_2)} \tag{8}$$

Equation 8 can also be called as the odds of relevance of candidate a to d given the evidence a has. This odds can be used to rank the term candidates. Taking the log of odds (i.e. logit) gives us

$$\log \frac{P(A|R_1)}{P(A|R_2)} = \log \frac{P(R_1|A)}{P(R_2|A)} - \log \frac{P(R_1)}{P(R_2)}$$

$P(A|R_1)$ and $P(A|R_2)$ are the class conditional probabilities for a being a candidate with evidence vector V given its different state of relevance. Since the chance of any candidate being relevant to d and to $\bar{d}$ without any evidence is the same (i.e. $P(R_1)/P(R_2) = 1$), we can safely ignore the second term (i.e. the odds of relevance without evidence) in Equation 8. This gives us

$$\log \frac{P(A|R_1)}{P(A|R_2)} = \log \frac{P(R_1|A)}{P(R_2|A)} \tag{9}$$

To score and rank the term candidates $a \in TC$ based on the evidences they have, we define the *Odds of Termhood (OT)* as

$$OT(a) = \log \frac{P(A|R_1)}{P(A|R_2)} \tag{10}$$

Since we are only interested in the relative ranking, ranking candidates using OT, according to Equation 9, is the same as ranking the candidates according to our Aim 2 as formulated in Equation 8. Obviously, from Equation 10, our initial predicament on not being able to empirically determine prior probabilities $P(A)$ and $P(R_1)$ is no longer a problem.

Assumption 4 *Independence between evidences in V.*

Next we can decompose the evidence vector V associated with each candidate a to enable the assessment of the class conditional probabilities $P(A|R_1)$ and $P(A|R_2)$. Given Assumption 4, $P(A|R_1)$ and $P(A|R_2)$ can be expanded as

$$P(A|R_1) = \prod_i P(E_i|R_1) \tag{11}$$
$$P(A|R_2) = \prod_i P(E_i|R_2)$$

where $P(E_i|R_1)$ and $P(E_i|R_2)$ is the probability of a as a candidate associated with evidence E_i given its different state of relevance. Substituting Equation 11 in 10 will give us

$$OT(a) = \sum_i \log \frac{P(E_i|R_1)}{P(E_i|R_2)} \tag{12}$$

Lastly, to simplify the notation, individual scores are defined for each evidence E_i, and we call them *evidential weights (O_i)*

$$O_i = \frac{P(E_i|R_1)}{P(E_i|R_2)} \tag{13}$$

and substituting Equation 13 in 12 gives us

$$OT(a) = \sum_i \log O_i \tag{14}$$

The purpose of OT is similar to many other functions for scoring and ranking term candidates such as those reviewed in Section 2. However, what differentiates our new function from the existing ones is the fact that OT is founded upon and derived in a probabilistic framework with explicit assumptions. Moreover, as shown in the following Section 3.2, the individual evidences themselves are formulated based on probability theory and the necessary term distributions are derived from formal distribution models.

3.2 Formalising Evidences in a Probabilistic Framework

Commonly adopted characteristics for determining the relevance of terms (Kageura & Umino 1996) are highlighted below in Definition 3.

Definition 3 *Characteristics of term relevance*

1. A term candidate is relevant to a domain if it appears relatively more frequent in that domain than in others.

2. A term candidate is relevant to a domain if it appears only in one domain.

3. A term candidate relevant to a domain may have biased occurrences in that domain.

4. A complex term candidate is relevant to a domain if its head is specific to that domain.

We propose seven evidences for the evidence vector V to capture the characteristics presented in Definition 2 and 3. They are as follow:

Evidence 1: Occurrence of candidate a

Evidence 2: Existence of candidate a

Evidence 3: Specificity of the head a^h of a

Evidence 4: Uniqueness of candidate a

Evidence 5: Exclusivity of candidate a

Evidence 6: Pervasiveness of candidate a

Evidence 7: Clumping tendency of candidate a

However, due to space limitations, here we present Evidence 3 and 4 as a proof of concept. It is worthwhile to note that evidences can always be introduced or removed depending on the goal or constraints imposed upon the applications implementing OT of Equation 14. The various evidences contribute to the computation of the corresponding evidential weights O_i, which in turn are summed to produce the final ranking of OT. Since OT serves as a probabilistically-derived formulaic realisation of our Aim 2, the various O_i can be considered as manifestations of sub-aims derivable from Aim 2. Each sub-aim is formulated into its corresponding evidential weight using the probability distributions of the occurrences of term candidates in d and in $\bar{d}$:

- P(occurrence of a in d)=$P(a, d)$ is the probability of occurrence of a in the domain corpus d.
- P(occurrence of a in $\bar{d}$)=$P(a, \bar{d})$ is the probability of occurrence of a in the contrastive corpora $\bar{d}$.

There are a few possible models for such distribution. One of them is the Zipf-Mandelbrot model which have been rigorously discussed by (Tullo & Hurford 2003). We use the Zipf-Mandelbrot model to obtain the distributions for both $P(a, d)$ and $P(a, \bar{d})$.

3.2.1 Odds of Specificity

The evidential weight O_3 focuses specifically on Definition 3.4 for complex term candidates. O_3 is meant for capturing the odds of whether the inherently ambiguous head a^h of a complex term a is specific to d. If the head a^h of a complex terms is found to occur individually without a in large numbers across different domains, then the specificity of the concept represented by a^h and a in d is doubtable. O_3 can be formally stated as:

> **Sub-Aim 3** What are the odds that the head a^h of a complex term candidate a is specific to d?

The head of a complex term candidate is considered as specific to a domain if the head and the candidate itself both have higher tendency of occurring together in that domain. The higher the chances of co-occurrence of a and a^h in a domain, the more specific is a^h to that domain. For example, if the event of both *"bridge"* and *"network bridge"* occurring together in the *"computer networking"* domain is very high, this means the possibly ambiguous head *"bridge"* is used in a very specific context in that domain. In such cases, when *"bridge"* is encountered in *"computer networking"*, one can safely deduce that it refers to the same domain-specific concept as *"network bridge"*. Consequently, the more specific the head a^h is with respect to d, the less ambiguous its occurrence is in d. From Definition 3.4, the less ambiguous a^h is, the higher are the chances of its complex counterpart a being relevant to d.

To proceed further, we assume that the occurrences of candidate a and its head a^h within the same domain (i.e. either d or $\bar{d}$) are independent. Even though the independence assumption may not always be the case in reality, it does remove many complications related to the non-trivial formulation of O_3 and other evidential weights not presented here. As such,

- P(occurrence of a in $d \cap$ occurrence of a^h in d) = $P(a, d)P(a^h, d)$

Based on the assumptions above, we define O_3 for complex term candidates as:

$$
\begin{aligned}
O_3 &= \frac{P(\text{specificity of } a|R_1)}{P(\text{specificity of } a|R_2)} \\
&= \frac{P(\text{specificity of } a \text{ to } d)}{P(\text{specificity of } a \text{ to } \bar{d})} \\
&= \frac{P(\text{occurrence of } a \text{ in } d \cap \text{occurrence of } a^h \text{ in } d)}{P(\text{occurrence of } a \text{ in } \bar{d} \cap \text{occurrence of } a^h \text{ in } \bar{d})} \\
&= \frac{P(a, d)P(a^h, d)}{P(a, \bar{d})P(a^h, \bar{d})}
\end{aligned}
$$

3.2.2 Odds of Uniqueness

This evidential weight O_4 attempts to realise Definition 3.2. O_4 captures the odds of whether a is unique to d or to $\bar{d}$. A term candidate is considered as unique if it occurs only in one domain and not the other. Formally, O_4 is described as

> **Sub-Aim 4** What are the odds of term candidate a being unique to d?

Based on the following intuitively reasonable independence and complementary assumption of the events of occurrence and non-occurrence of candidate a,

- P(non-occurrence of a in d) $= 1 - P(a, d)$
- P(occurrence of a in $d \cap$ non-occurrence of a in $\bar{d}$) $= P(a, d)(1 - P(a, \bar{d}))$

we can mathematically formulate O_4 as:

$$
\begin{aligned}
O_4 &= \frac{P(\text{uniqueness of } a|R_1)}{P(\text{uniqueness of } a|R_2)} \\
&= \frac{P(\text{uniqueness of } a \text{ to } d)}{P(\text{uniqueness of } a \text{ to } \bar{d})} \\
&= \frac{P(\text{occurrence of } a \text{ in } d \cap \text{non-occurrence of } a \text{ in } \bar{d})}{P(\text{occurrence of } a \text{ in } \bar{d} \cap \text{non-occurrence of } a \text{ in } d)} \\
&= \frac{P(a, d)(1 - P(a, \bar{d}))}{P(a, \bar{d})(1 - P(a, d))}
\end{aligned}
$$

4 Experiments

To evaluate the effectiveness of our probabilistic framework for determining termhood using *Odds of Termhood (OT)*, here we carry out a comparative study with three existing scoring and ranking scheme, namely, *Contrastive Weight (CW)*, *NCvalue (NCV)* and *Termhood (TH)*. The implementation of CW, NCV and TH are in accordance to Equation 1 and 2, 3, and 4 respectively. Although only two evidences are presented in this paper, we have included all seven evidences during our implementation to obtain the new measure OT in this experiment. The summary of the data sets is presented in Table 1. The data sets consist of three distinct domain corpora d and a collection of contrastive corpora $\bar{d}$. The domain corpora consist of three distinct collections of text gathered from `BioMedCentral.com` in the area of *"musculoskeletal disease"* (denoted by d_{MUS}), *"cancer"* (denoted by d_{CAN}), and *"cardiovascular disease"* (denoted by d_{CAR}). The contrastive corpora is a single collection of news articles across a wide range of genres gathered from various sources such as `Reuters.com`, `CNet.com` and `ABC.com` between the period of February 2006 and April 2007. The term candidates and context words are extracted as instantiated sub-categorisation frames (Wong 2005) from

Table 1. Summary of the datasets employed throughout this paper for experiments and evaluations. For simplicity reasons, two notations are adopted: d to represent the domain corpus, and $\bar{d}$ to represent the contrastive corpora.

	Notation	Source	Domain	No. of documents (N)	No. of words (F)	Average no. of words per document
Contrastive corpora	$\bar{d}$	Reuters	Business	2,691	987,305	
			Sports	2,306	792,902	
			Politics	2,187	871,788	
			United States local news	612	223,588	
			Entertainment	2,280	862,233	
			Health	1,039	387,905	
		CNet	Technology	3,375	1,626,849	
		ABC	Australian local news	1,394	349,301	
		Discovery	Travel	291	138,178	
			History	65	31,253	
			Wildlife	247	117,016	
		AllRecipes	Cooking recipe	552	89,644	
			Total	***17,039***	***6,477,962***	*380.18*
Domain corpus 1	d_{MUS}	BioMed Central	Musculoskeletal diseases	302	860,601	
			Total	***976***	***2,533,717***	*2,596.02*
Domain corpus 2	d_{CAN}	BioMed Central	Cancer	453	1,043,070	
			Total	***453***	***1,043,070***	*2,302.58*
Domain corpus 3	d_{CAR}	BioMed Central	Cardiovascular diseases	282	607,973	
			Total	***282***	***607,973***	*2,155.93*

the three domain corpora. As a result, three sets of term candidates are produced, one for each of the three domains (i.e. d_{MUS}, d_{CAN} and d_{CAR}). The experiments on three different sets of terms from three different domain corpora are necessary to demonstrate the consistency of any trends exhibited by the four measures.

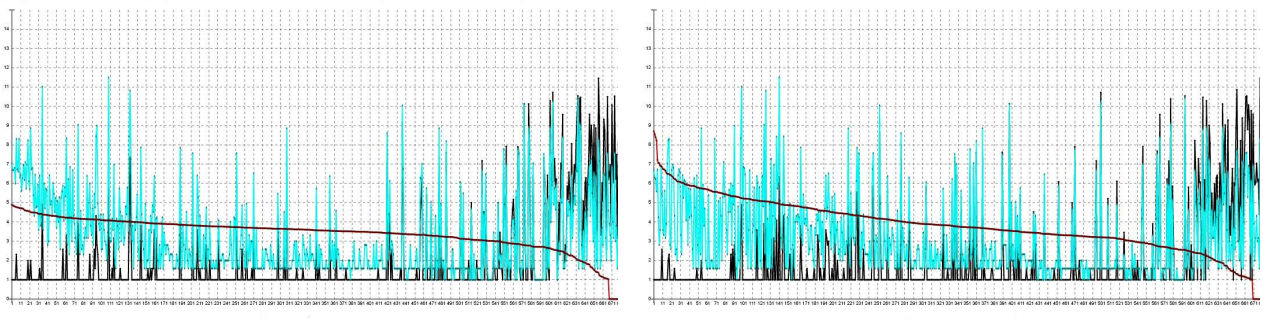
The four measures are used to score and rank the three sets of term candidates. The frequency distributions of the ranked candidates from d_{MUS}, d_{CAN} and d_{CAR} are shown in Figures 1, 2 and 3. The candidates are ranked in descending order according to their scores assigned by the respective measures. The first half of the graphs by CW, prior to the sudden surge of frequency consisted of only complex terms. Complex terms tend to have lower word counts compared to simple terms and hence, the disparity in the frequency distributions are clearly shown in Figures 1(c), 2(c) and 3(c). This is attributed to the biased treatment given to complex terms evident in Equation 2. However, priority is also given to complex terms by TH and OT, but as one can see from the distributions of candidates by TH in Figures 1(b), 2(b) and 3(b) and those by OT in Figures 1(a), 2(a) and 3(a), such undesirable trend does not occur. One of the explanation is CW relies heavily on frequency while TH and OT attempt to diversify the evidences. Even though frequency is a reliable source of evidence, the use of it alone is definitely inadequate (Cabre-Castellvi et al. 2001). As for the NCV measure, Figures 1(d), 2(d) and 3(d) show that scores for term candidates are calculated solely based on their domain frequencies. In other words, NCV is not suitable for performing contrastive analysis and hence, cannot be employed for term extraction to identify between domain-specific terms. Another advantage of TH and OT is their ability to assign higher weights to terms that occur relatively more frequent in d than in $\bar{d}$. This is evident through the gap between f_d and $f_{\bar{d}}$, especially at the beginning of the x-axis. Candidates along the end of the x-axis are those with $f_{\bar{d}} > f_d$. However, the discriminating power of OT is better since the gap between f_d and $f_{\bar{d}}$ is wider and lasted longer.

In addition to the absence of evaluation methods and datasets, the subjective nature of termhood assessment makes the tasks of objectively evaluating

Table 2. A summary of the mean (μ) and standard deviation (σ) of the scores assigned by the four measures (i.e. NCV, CW, TH and OT) for the three sets of term candidates extracted from three different domain corpora (i.e. d_{MUS}, d_{CAN} and d_{CAR}). The sum of the domain frequencies and of the contrastive frequencies of the three sets of term candidates are also shown in this table.

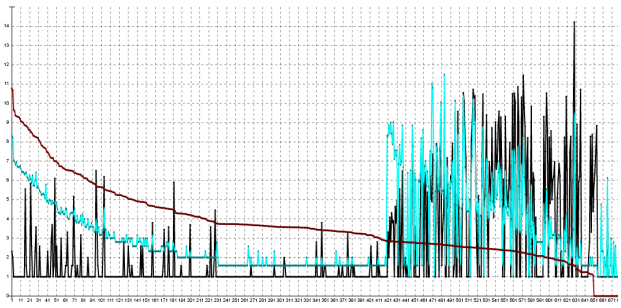
		μ *of weights*	σ *of weights*	$\sum f_d$	$\sum f_{\bar{d}}$	$\frac{\sum f_d}{\sum f_{\bar{d}}}$
d_{MUS}	***NCV***	2105.71	35435.75	33286	65396	0.51
	CW	43.10	132.91			
	TH	20.98	32.63			
	OT	10.44	5.93			
		μ *of weights*	σ *of weights*	$\sum f_d$	$\sum f_{\bar{d}}$	$\frac{\sum f_d}{\sum f_{\bar{d}}}$
d_{CAN}	***NCV***	112886.02	1209270.22	120820	134936	0.90
	CW	49.80	241.98			
	TH	25.71	38.05			
	OT	10.77	5.22			
		μ *of weights*	σ *of weights*	$\sum f_d$	$\sum f_{\bar{d}}$	$\frac{\sum f_d}{\sum f_{\bar{d}}}$
d_{CAR}	***NCV***	1778.82	29670.35	38740	88790	0.44
	CW	34.78	183.13			
	TH	15.90	17.06			
	OT	10.83	4.98			

and comparing our new measure with existing approaches a problem domain by itself. In the words of (Damle & Uren 2005), *"Unfortunately, there is no objective evaluation method reported in the literature for term extraction..."*. However, our subjective assessments of OT and three other existing measures offer promising insights into probabilistically-derived measures for termhood. Table 2 summarises the mean and standard deviation of the weights generated by the various measures. One can notice the extremely high dispersion from the mean of the weights generated by CW and also NCV. We speculate that such trends are due to the erratic assignments of weights, heavily influenced by domain frequencies. This is further enforced by the visible increase of the means and standard deviations of the weights produced by CW, NCV and even TH as the domain frequency f_d increases. On the other hand, our probabilistically-motivated measure OT appeared to be unaffected by

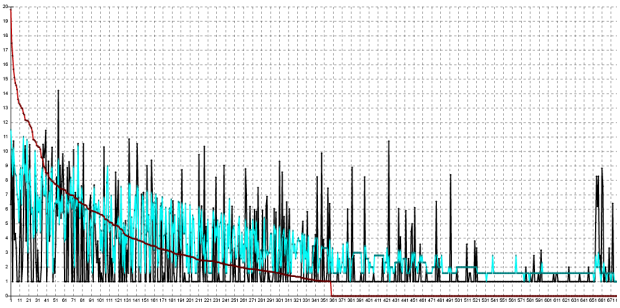


(a) Candidates ranked by OT

(b) Candidates ranked by TH

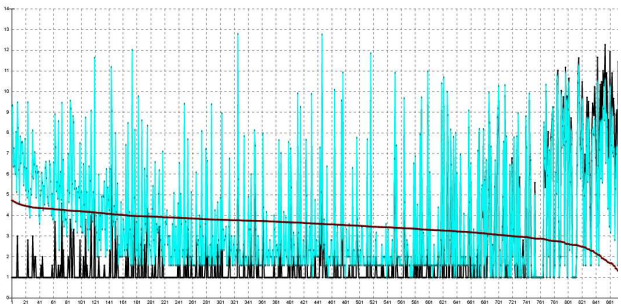


(c) Candidates ranked by CW

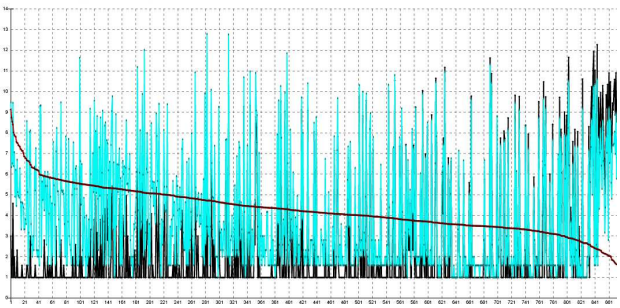


(d) Candidates ranked by NCV

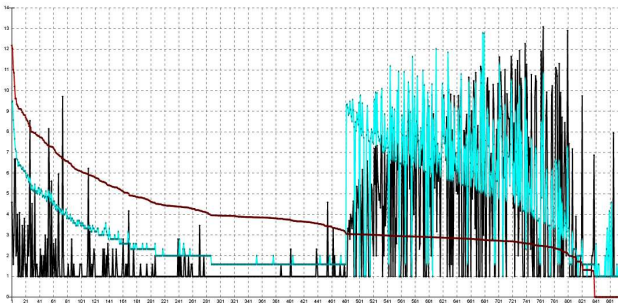
Figure 1: This graph shows the frequency distributions of 709 candidates extracted from d_{MUS} ranked in descending order according to the scores assigned by the respective measures. The black, oscillating line represents the frequency of the candidates in the contrastive corpora $f_{\bar{d}}$ while the greyish, oscillating line is the domain frequency f_d. The single dark line that spans from the left to the right of the graph in descending order represents the scores assigned by the respective measures.



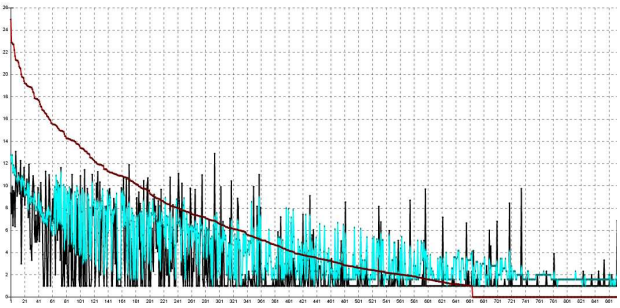
(a) Candidates ranked by OT



(b) Candidates ranked by TH

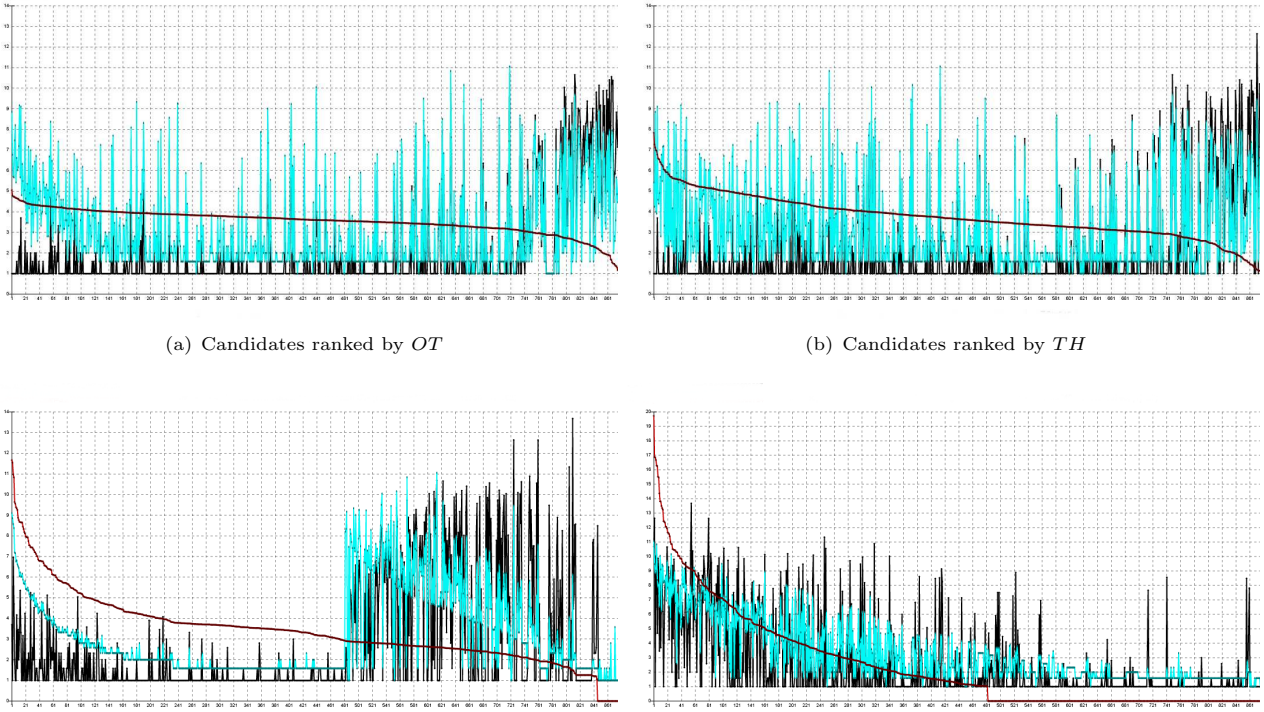


(c) Candidates ranked by CW



(d) Candidates ranked by NCV

Figure 2: This graph shows the frequency distributions of 709 candidates extracted from d_{CAN} ranked in descending order according to the scores assigned by the respective measures. The black, oscillating line represents the frequency of the candidates in the contrastive corpora $f_{\bar{d}}$ while the greyish, oscillating line is the domain frequency f_d. The single dark line that spans from the left to the right of the graph in descending order represents the scores assigned by the respective measures.



(a) Candidates ranked by OT

(b) Candidates ranked by TH

(c) Candidates ranked by CW

(d) Candidates ranked by NCV

Figure 3: This graph shows the frequency distributions of 709 candidates extracted from d_{CAR} ranked in descending order according to the scores assigned by the respective measures. The black, oscillating line represents the frequency of the candidates in the contrastive corpora $f_{\bar{d}}$ while the greyish, oscillating line is the domain frequency f_d. The single dark line that spans from the left to the right of the graph in descending order represents the scores assigned by the respective measures.

Table 3. The Spearman rank correlation coefficients ρ between all pairs of measures over the three sets of term candidates extracted from d_{MUS}, d_{CAN} and d_{CAR}.

	ρ	*NCV*	*CW*	*TH*	*OT*
d_{MUS}	*NCV*	1	0.2527	0.3421	0.3321
	CW	0.2527	1	0.5958	0.6333
	TH	0.3421	0.5958	1	0.8565
	OT	0.3321	0.6333	0.8565	1
	ρ	*NCV*	*CW*	*TH*	*OT*
d_{CAN}	*NCV*	1	0.0053	0.2138	0.1457
	CW	0.0053	1	0.4637	0.5313
	TH	0.2138	0.4637	1	0.8289
	OT	0.1457	0.5313	0.8289	1
	ρ	*NCV*	*CW*	*TH*	*OT*
d_{CAR}	*NCV*	1	0.1221	0.1985	0.1467
	CW	0.1221	1	0.4737	0.5129
	TH	0.1985	0.4737	1	0.8211
	OT	0.1467	0.5129	0.8211	1

the changes in frequencies. We also employ the Spearman rank correlation coefficient to study the possibility of any correlation between the four ranking schemes under evaluation. Table 3 summarises the coefficients between the various measures. Note that there is a strong correlation between the ranks produced by our new probabilistic measure OT and the ranks by the ad-hoc measure TH. This correlation is consistent throughout all the experiments using different sets of term candidates. The correlation of TH with OT reveals the possibility of providing mathematical justifications for the former's heuristically-motivated ad-hoc approach using a general probabilistic framework. We believe by adjusting the inclusion or exclusion of various evidences, other ad-hoc measures can be captured as well.

5 Conclusions

In this paper, we presented a probabilistically-derived measure termed as the *Odds of Termhood (OT)* for scoring and ranking term candidates for term extraction. We have also introduced seven evidences, founded on formal models of word distribution, to facilitate the calculation of OT. The evidences are motivated by characteristics of terms in a domain, which are made explicit. The fact that evidences can be added or removed makes OT a highly flexible framework that is adaptable to the applications' requirements and constraints. Our experiments comparing OT with three other existing ad-hoc measures, namely CW, NCV and TH have demonstrated the effectiveness of the new measure and the new framework.

More research is required for introducing new evidences to realise more term characteristics. Due to the difficulty in objectively determining the performance of termhood measures, we intend to assess OT within the scope of a larger application such as document retrieval to establish its precision, recall and accuracy.

Acknowledgement

This research was supported by the Australian Endeavour International Postgraduate Research Scholarship, and the Research Grant 2006 by the University of Western Australia.

References

Basili, R., Moschitti, A., Pazienza, M. & Zanzotto, F. (2001), A contrastive approach to term extrac-

tion, *in* 'Proceedings of the 4th Terminology and Artificial Intelligence Conference (TIA)', France.

Bookstein, A., Klein, S. & Raita, T. (1998), 'Clumping properties of content-bearing words', *Journal of the American Society of Information Science* **49**(2), 102–114.

Bookstein, A. & Swanson, D. (1974), 'Probabilistic models for automatic indexing', *Journal of the American Society for Information Science* **25**(5), 312–8.

Cabre-Castellvi, T., Estopa, R. & Vivaldi-Palatresi, J. (2001), Automatic term detection: A review of current systems, *in* D. Bourigault, C. Jacquemin & M. LHomme, eds, 'Recent Advances in Computational Terminology', John Benjamins.

Chinchor, N., Lewis, D. & Hirschman, L. (1993), 'Evaluating message understanding systems: An analysis of the third message understanding conference (muc-3)', *Computational Linguistics* **19**(3), 409–449.

Church, K. & Gale, W. (1995), Inverse document frequency (idf): A measure of deviations from poisson, *in* 'Proceedings of the ACL 3rd Workshop on Very Large Corpora'.

Cilibrasi, R. & Vitanyi, P. (2007), 'The google similarity distance', *IEEE Transactions on Knowledge and Data Engineering* **19**(3), 370–383.

Damle, D. & Uren, V. (2005), Extracting significant words from corpora for ontology extraction, *in* 'Proceedings of the 3rd International Conference on Knowledge Capture', Alberta, Canada.

Frantzi, K. & Ananiadou, S. (1997), Automatic term recognition using contextual cues, *in* 'Proceedings of the IJCAI Workshop on Multilinguality in Software Industry: the AI Contribution', Japan.

Fuhr, N. (1986), Two models of retrieval with probabilistic indexing, *in* 'Proceedings of the 9th ACM SIGIR International Conference on Research and Development in Information Retrieval'.

Fuhr, N. (1992), 'Probabilistic models in information retrieval', *The Computer Journal* **35**(3), 243–255.

Hirschman, L., Yeh, A., Blaschke, C. & Valencia, A. (2005), 'Overview of biocreative: Critical assessment of information', *BMC Bioinformatics* **6**(1), S1.

Jones, K., Walker, S. & Robertson, S. (1998), 'A probabilistic model of information retrieval: Development and status', *Information Processing and Management* **36**(6), 809–840.

Kageura, K. & Umino, B. (1996), 'Methods of automatic term recognition: A review', *Terminology* **3**(2), 259–289.

Kit, C. (2002), Corpus tools for retrieving and deriving termhood evidence, *in* 'Proceedings of the 5th East Asia Forum of Terminology', Haikou, China.

Lewis, D. (1998), Naive (bayes) at forty: The independence assumption in information retrieval, *in* 'Proceedings of the 10th European Conference on Machine Learning'.

Loukachevitch, N. & Dobrov, B. (2004), Sociopolitical domain as a bridge from general words to terms of specific domains, *in* 'Proceedings of the 2nd International Global Wordnet Conference'.

Manning, C. & Schutze, H. (1999), Foundations of statistical natural language processing, MIT Press, MA, USA.

Tullo, C. & Hurford, J. (2003), Modelling zipfian distributions in language, *in* 'Proceedings of the ESSLLI Workshop on Language Evolution and Computation', Vienna.

vanRijsbergen, C. (1979), Automatic text analysis, *in* 'Information Retrieval', University of Glasgow.

Wong, W. (2005), Practical approach to knowledge-based question answering with natural language understanding and advanced reasoning, Master's thesis, National Technical University College of Malaysia, arXiv:cs.CL/0707.3559.

Wong, W., Liu, W. & Bennamoun, M. (2007*a*), Determining termhood for learning domain ontologies using domain prevalence and tendency, *in* 'Proceedings of the 6th Australasian Conference on Data Mining (AusDM)', Gold Coast.

Wong, W., Liu, W. & Bennamoun, M. (2007*b*), Determining the unithood of word sequences using mutual information and independence measure, *in* 'Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics (PACLING)', Melbourne, Australia.

Zipf, G. (1935), The psycho-biology of language, Houghton Mifflin, Boston, MA.

Zipf, G. (1949), Human behaviour and the principle of least-effort, Addison-Wesley, Cambridge, MA.

Using Corpus Analysis to Inform Research into Opinion Detection in Blogs

Deanna Osman[1] **John Yearwood**[2] **Peter Vamplew**[3]

School of Information Technology and Mathematical Sciences
University of Ballarat,
P.O. Box 663, Ballarat Victoria 3353, Australia,

[1]Email: d.osman@ballarat.edu.au
[2] Email: j.yearwood@ballarat.edu.au
[3] Email: p.vamplew@ballarat.edu.au

Abstract

Opinion detection research relies on labeled documents for training data, either by assumptions based on the document's origin or by using human assessors to categorise the documents. In recent years, blogs have become a source for opinion identification research (TREC Blog06). This study analyses the part-of-speech proportion and the words used within various corpora, determining key differences and similarities useful when preparing for opinion identification research. The resulting comparisons between the characteristics of the various corpora is detailed and discussed. In particular, opinion-bearing and non-opinion Blog06 documents were found to display a high level of similarity, indicating that blog documents assessed at the document level cannot be used as training data in opinion identification research.

Keywords: Blogs, Weblogs, Blog06, TREC, Opinion detection, Opinion identification

1 Introduction

Weblogs (blogs) are a fast growing phenomenon on the World Wide Web as they allow people to publish their thoughts and opinions on any topic they choose. In September 2007 a blog tracking company, Technorati, Inc., reported that it was monitoring 104.9 million blogs worldwide (*About Technorati, Accessed September* 2007), up from 4.2 million in October 2004 (Rosenbloom 2004).

The majority of blog authors surveyed by Lenhart & Fox (2006) indicated that the reason they write blogs is to share their knowledge and skills, with a high proportion of the topics being about personal and life experiences. Blog authors are inspired by the things that happen to them and want to share these experiences. Often blog authors will express their opinions about products, event and people which impacts their lives. Automatically gathering and the analysis of these opinions could prove valuable in a number of applications.

Such a search engine could be used by manufacturers to access opinions on their products or a competitor's product. For example, negative opinions about a competitor's product may provide a competitive edge for a new design. Governments could search blogs for qualitative information to support quantitative research (opinion polls) regarding new policies or upcoming elections. Small businesses, who do not have a large 'market research' budget, could gain access to millions of people who potentially have an opinion relating to them.

Searching the blogosphere for opinions about life experience and other topics within blogs, is an arduous task using traditional search engines. A search engine that searches the blogosphere for opinions on a given topic requires the inclusion of an opinion identification module in the search engine architecture. The task of opinion identification has previously been investigated in a non-blog context. Newswire articles have been used in opinion identification research (Yu & Hatzivassiloglou 2003, Kim & Hovy 2005) to create training and testing data, by dividing the articles into opinion-bearing and non-opinion-bearing categories. Editorial and Letter to editor articles were assumed to be opinion-bearing, while Business and News articles were categorised as non-opinion-bearing (Yu & Hatzivassiloglou 2003, Kim & Hovy 2005).

These documents formed the training and testing data for Naive Bayes machine-learning (Yu & Hatzivassiloglou 2003), and were used to create a list of opinion-bearing and non-opinion-bearing words and opinion scores[1] (Yu & Hatzivassiloglou 2003, Kim & Hovy 2005). This list was expanded by adding synonyms and antonyms of opinion-bearing and non-opinion-bearing words (Kim & Hovy 2005). The original list (Yu & Hatzivassiloglou 2003) comprised of adjectives, adverbs, nouns and verbs.

The resulting list of words (adjectives, nouns, verbs & adverbs) was used to identify opinion-bearing sentences (Kim & Hovy 2005) by applying the scores to the words within the sentences. Sentences were assessed by three evaluators to enable precision and recall to be calculated. The Wall Street Journal articles in each category were not evaluated to determine the validity of the hypothesis that Editorial/Letter to editor articles are opinion-bearing and Business/News articles are non-opinion-bearing.

Opinion identification research was sponsored by the Text REtrieval Conference (TREC) within blogs for the first time in 2006. TREC collected blog posts and comments over an eleven week period to create a blog track (Blog06). One of the tasks for participants was to identify opinion-bearing blogs on a given topic. This task was made more

[1]Opinion scores indicate how strongly a word expresses an opinion.

difficult by the lack of an annotated blog corpus for training data (Yang, Si & Callan 2006, Zhang & Zhang 2006). Various other corpora were used by Blog06 participants as the training data, including the list created from the Wall Street Journal collection (Yu & Hatzivassiloglou 2003, Kim & Hovy 2005).

The results of the opinion identification task within Blog06 were varied (Ounis, de Rijke, Macdonald, Mishne & Soboroff 2006), with Mean Average Precision (MAP) ranging from 0.2983 to 0.0001. A question that arises from the Blog06 results is *'Does identifying opinions within blog posts and comments require different training data to identifying opinions within more traditional corpora?'*

Blogs are an informal form of communicating, where usually the audience of the blog is known to the author (Nardi, Schiano, Gumbrecht & Swartz 2004). The author of a blog is free to write informally and use any language required to express their thoughts and opinions.

On the other hand, newswire articles are written using a formal structure, using proper English without slang and word abbreviation. These articles have trained, experienced authors and an editor to ensure high quality writing techniques are used and words are not used out of context or with ambiguous meaning.

Therefore, it might be expected that blogs may exhibit different language usage and characteristics from other document corpora and training data developed from those corpora may not be applicable to a blog corpora. This study provides an analysis of various corpora and reports on the differences, with the view to gaining an insight into how blogs differ from traditional opinion identification corpora. A broad view of the characteristics of opinion-bearing versus non-opinion-bearing text within different corpora is also provided. The corpora analysed are listed in Table 1.

There are two main approaches to the corpora analysed:

1. The proportion of part-of-speech types within each corpus (Section 3.2).
2. The use of unique or 'weird' (Gillam & Ahmad 2005) words and slang (Section 3.3).

A similar pattern between the opinion and non-opinion corpora respectively in the above-mentioned approaches was not found, whilst the opinion and non-opinion blog corpora were found to display similar characteristics to each other. The lack of variation between BlogOp and BlogNop led to the non-relevant text being removed from these corpora and smaller sentence corpora being created, section 3.1 details the methodology applied. Further analysis of the opinion and non-opinion blog sentence corpora found a greater variation in the characteristics analysed in this study, indicating the need for analysis of the relevance of the text with blog documents prior to training and testing data being created for opinion detection research.

The remainder of this paper is organised as follows. Section 2 describes the various corpora that are listed in Table 1. Section 3 describes the methodology applied to each of the areas of analysis. The results of the analysis are discussed in section 4, with section 5 concluding the study and discussing future work.

2 Corpora

There are many document collections available for opinion identification and other research, with each one having different characteristics and features. Some are assessed to assist researchers when analysing their research, whilst others are generic document collections that need to be assessed according to the research area. This study includes the main corpora used by Blog06 participants and reports key differences between these and the analysis of blogs extracted from the assessed Blog06 data. This section describes each corpus used in this study and lists the total number of word types (distinct words) and the total number of tokens (words) in each corpus. A summary of the total number of word types and tokens in each corpus is detailed in Table 3.

Table 1: Corpora analysed in this study and the category each has been assigned to. Note: The Blog06 and Customer Review corpora has been assessed by human assessors. The Movie Review corpus is based on assumptions detailed in Section 2.8. An assumption has been made (for this study) on the category for the remaining corpora. * Indicates that the corpus is a subset of another corpus analysed in this study.

	Opinion Bearing	**Non Opinion Bearing**	**Mixed**
Formal writing/ News	WSJOp	Reuters NYT WSJNop	BNC MPQA WSJMix
Blogs	BlogOp OP3* OP5*	BlogNop NOP3* NOP5*	BlogMix
Webpages	MROp CRD	MRNop	

The corpora analysed in this study were divided into three categories: (1) Formal writing/News, (2) Blogs and (3) Webpages, with further distinctions between opinion-bearing (documents expressing an opinion), non-opinion-bearing (documents that do not express an opinion) and mixed (documents that have not been assessed). The corpora are listed in Table 1. Categories for the Wall Street Journal corpora were made using assumptions used by in the original research (Yu & Hatzivassiloglou 2003, Kim & Hovy 2005), which have not been verified by human assessors. Assumptions were made (by this researcher) for the remainder of the Formal writing/News corpora to enable them to be placed into categories. The categories for the Movie Review corpora were based on assumptions made by Pang & Lee (2004).

2.1 British National Corpus

The British National Corpus (BNC) was included in this study as a standard corpus for comparison purposes, and as a reference list of standard English words (Gillam & Ahmad 2005). BNC is made

up of written and spoken English (*What is the BNC?* 2007), and was collected over several years (1991–1994) with no new texts being added since completion. However, revisions were made in 2001 and 2007. BNC provides a general language corpus for this study. This collection is not divided into opinion-bearing and non-opinion-bearing. 4,050 documents contain a total number of 470,821 word types and the token count is 96,353,012. The mean length of each document is 23,797 tokens.

2.2 Reuters

Reuters was included in this study as an example of news articles. Reuters is a collection of 7,190 news articles dating between August 1996 and October 1996. As the articles are reporting news events, they are assumed to be non-opinion-bearing in this study. This collection contains 43,963 word types and 1,565,380 tokens. The mean length of each document is 218 tokens.

2.3 New York Times

The New York Times (NYT) corpus is a subset of the AQUAINT[2] document collection. It was included in this study as a further example of news articles. The articles range in date from June 1998 to September 2000, totaling 820 days. The corpus contains 314,452 news articles, totaling 830,075 word types and 231,856,086 tokens. This corpus contains news articles and has been categorised as non-opinion. The mean length of each document is 737 tokens.

2.4 Wall Street Journal

The Wall Street Journal (WSJ) corpus is a subset of the TIPSTER[2] document collection. The news articles have been collected in the years 1987 to 1992. Some of these articles have headings indicating which category the article originates from (*Editorial, Letter to editor, Business* or *News*). These categories were used to divide the corpus into opinion-bearing (Editorial and Letter to editor) and non-opinion-bearing (Business and News) (Yu & Hatzivassiloglou 2003, Kim & Hovy 2005). This corpus is divided into three subset corpora:

- WSJOp – 4,190 articles, 47,939 word types, 1,364,326 tokens, document mean length: 326 tokens
- WSJNop – 19,731 articles, 58,509 word types, 4,625,526 tokens, document mean length: 234 tokens
- WSJMix[3] – 29,324 articles, 288,242 word types, 60,402,701 tokens, document mean length: 2,060

The WSJ collections were included in this study to allow the comparison of opinion-bearing and non-opinion-bearing news articles to opinion-bearing and non-opinion-bearing blogs. Opinion identification research (Yu & Hatzivassiloglou 2003, Kim & Hovy 2005) used WSJ articles to develop training and testing data. A Blog06 participant (Eguchi & Shah 2006) used this word list for training data in the opinion identification task.

[2] http://www.ldc.upenn.edu/

[3] The remainder of the documents where the title did not indicate into which category the document should be placed.

2.5 MPQA

The MPQA corpus contains 535 news articles collected from various sources (*MPQA Releases* 2007, Wiebe 2002). It contains 6,867 word types, and totals 50,502 tokens. A Blog06 participant (Eguchi & Shah 2006) used this corpus for training data in the opinion identification task. The mean length of each document is 94 tokens.

2.6 Blog06

100,649 blogs were crawled over an eleven week period, made up of 70,701 'top blogs'[4], 17,969 splogs and 11,979 'other blogs'[5]. The resulting corpus contains 3,215,171 blog documents (MacDonald & Ounis 2006).

Judgements by human assessors Ounis et al. (2006) on 67,382 documents placed them into one of five categories (detailed in Table 2). These assessed documents were divided into three document collections for this study:

1. BlogOp – 10,446 documents, 404,131 word types, 28,713,436 tokens, blog mean length: 2,749 tokens
2. BlogNop – 8,281 documents, 338,895 word types, 19,438,021 tokens, blog mean length: 2,347 tokens
3. BlogMix[6] – 42,663 documents, 866,570 word types, 105,824,131 tokens, blog mean length: 2,480 tokens

Table 2: Number of documents allocated, by NIST assessors, to each assessment category (Ounis et al. 2006)

Relevance Scale	Label	No. of Documents
Not Judged	-1	0
Not Relevant	0	47,491
Adhoc-Relevant	1	8,361
Negative Opinion	2	3,707
Mixed Opinion	3	3,664
Positive Opinion	4	4,159
(Total)	-	67,382

2.7 BlogOpSent and BlogNopSent

The NIST assessments on the blog documents that place them into 'opinion-bearing' and 'non-opinion-bearing' categories were done at the document level, meaning that assessed documents contained text relevant to the topic (Op/Nop) and text not relevant to the topic (Op/Nop). This results in the characteristics of BlogOp and BlogNop being very similar. To enable analysis and reporting on the differences between these two corpora, they were divided into subsets by removing the text not relevant to the

[4] Top blogs were selected by Nielsen BuzzMetrics and the University of Amsterdam (Ounis et al. 2006)

[5] Blogs from a mix of genre

[6] Blogs judged as 'not relevant' were blogs retrieved in the information retrieval process and judged as not being relevant to the query topic by the NIST human assessor

given topic.

The documents within each corpus (BlogOp and BlogNop) were divided into single sentences and indexed using the Lucene Search Engine[7]. Lucene was used to create a list of individual sentences relevant to the given topic and these sentences were used as the first sentence in a block of text extracted from each blog. The blocks size was three sentences and five sentences. Full details of the methodology applied is explained in Section 3.1.

The resulting subsets are entitled:

- OP3 – 58,277 sentences, 72,018 word types, 1,867,411 tokens, sentence mean length: 32
- OP5 – 86,869 sentences, 83,231 word types, 2,305,358 tokens, sentence mean length: 26
- NOP3 – 45,821 sentences, 65,025 word types, 1,354,630 tokens, sentence mean length: 29
- NOP5 – 66,534 sentences, 72,584 word types, 1,615,991 tokens, sentence mean length: 24

2.8 Movie Review Data

This corpus[8] is made up of 5,000 subjective (MROp 13,765 word types, 100,136 tokens and sentence mean length: 20 tokens) and 5,000 objective (MRNop 14,325 word types, 110,283 tokens and sentence mean length: 22 tokens) sentences (Pang & Lee 2004). Two websites were used as the source for these sentences:

- Rotten Tomatoes[9] – these were assumed to be subjective (Pang & Lee 2004)
- Internet Movie Database[10] – these were assumed to be objective (Pang & Lee 2004)

A Blog06 participant (Yang, Yu, Valerio & Zhang 2006) used this corpus for training data in the opinion identification task.

2.9 Customer Review Data (CRD)

This corpus contains customer reviews on digital cameras, cellular phones, mp3 players and dvd players, which were collected from amazon.com, and annotated by Minqing Hu and Bing Liu[11]. There is 5,015 word types and 59,317 tokens in this corpus. The mean length of the 4,256 sentences is 14 tokens. A Blog06 participant (Yang, Yu, Valerio & Zhang 2006) used this corpus for training data in the opinion identification task.

3 Methodology

This section discusses the methodology applied in the analysis of key differences between corpora originating from different sources and the results are reported in the Results and Discussion section (Section 4). Two features of the corpora were analysed: (1) Part-of-speech (POS) Proportions and (2) Unique/Weird Words and Slang. All corpora were analysed using the above-mentioned methods and reported in results section, excluding the blog sentence corpora. The analysis on OP3, OP5, NOP3 and NOP5 is reported separately in Section 4.3.

[7] http://lucene.apache.org/java/docs/
[8] http://www.cs.cornell.edu/people/pabo/movie-review-data
[9] http://www.rottentomatoes.com/
[10] http://www.imdb.com
[11] http://www.cs.uic.edu/ liub/FBS/FBS.html

Table 3: Total word types, tokens and mean document length in the corpora analysed in this study. *Mean length is at the sentence level. ♢ Indicates that the corpus is a subset of another corpus.

Corpus	Word Types	Tokens	Mean Document Length
BNC	470,821	96,353,012	23,797
Reuters	43,963	1,565,380	218
NYT	830,075	231,856,088	737
WSJOp	47,939	1,364,326	326
WSJNop	58,509	4,625,526	234
WSJMix	288,242	60,402,701	2,060
MPQA	6,867	50,502	94
BlogOp	404,131	28,713,436	2,749
BlogNop	338,895	19,438,021	2,347
BlogMix	866,570	105,824,131	2,480
BlogOp			
OP3♢	72,018	1,867,411	32*
OP5♢	83,231	2,305,358	26*
BlogNop			
NOP3♢	65,025	1,354,630	29*
NOP5♢	72,581	1,615,991	24*
MROp	13,765	100,136	20*
MRNop	14,325	110,283	22*
CRD	5,015	59,317	14*

3.1 Removing Non-Relevant Text from the Blog Corpora

Due to the similarity between BlogOp and BlogNop, further analysis was done on these corpora. The content of blogs contained four types of text: (1) Opinion-bearing – off topic, (2) Non-opinion-bearing – off topic, (3) Opinion-bearing – on topic (BlogOp only[12]), and (4) Non-opinion-bearing – on topic.

The text within each document in the BlogOp and BlogNop respectively, was separated into single sentence blocks and indexed using The Lucene Search engine. The sentences relevant to the given topic were retrieved and placed into a list.

The structure of the text within the blogs led to some relevant text not being retrieved by the Lucene Search engine. An example of this was for the given topic 'March of the Penguins', where the title of the documentary was in one sentence and the opinion on the documentary (not mentioning the query term) was in the next sentence.

To reduce the impact of this, the text within each relevant sentence was retrieved, along with the following two/four[13] sentences. Sentences were included once only in the resulting subset of text, any duplications were removed prior to the collation of the text. The analysis methods described in the remainder

[12] The assumption was made that all relevant text within the BlogNop corpus was non-opinion-bearing.
[13] Depending on the sentence block size.

of this section were applied to these four subsets of text, similarly to the other corpora listed in Section 2.

3.2 Part-of-Speech Proportions

An area of interest is whether one type of corpus has a higher proportion of a particular POS type. Three part-of-speech taggers were tested for speed and robustness by Johnson, Malhotra & Vamplew (2006): *The Stanford NLP Group Loglinear Part-Of-Speech Tagger* (2006), *The MontyLingua natural language package* (2006) and *QTag probabilistic parts-of-speech tagger* (2006). QTag was found to be the fastest and most robust of these three (Johnson, Malhotra & Vamplew 2006).

Each corpus was tagged using QTag and the proportions of the following categories[14] were summarised:

- Adjectives – general, comparative and superlative
- Nouns – common singular, common plural, proper singular and proper plural
- Pronouns – indefinite, personal, possessive (my, his), reflexive, 'wh-' (who, that) and possessive (whose)
- Adverbs – general, comparative and superlative
- Verbs – base, past tense, '-ing' (believing), past participle and '-s' (believes)
- Unclassified – words that QTag could not classify

The proportions were used as a vector and the similarity between each vector calculated using the following formula, where v is the vector, p is the position within the vector and n is the vector length. The 'norm' of the vector is calculated:

$$\parallel v1 \parallel = \sqrt{\sum_{k=1}^{n} p_k^2}$$

and the similarity $(\widehat{v1} \cdot \widehat{v2})$ is calculated:

$$\widehat{v1} \cdot \widehat{v2} = \frac{\sum_{i=1}^{n}(v1_i \cdot v2_i)}{\parallel v1 \parallel \cdot \parallel v2 \parallel}$$

The results for each corpus are detailed and discussed in the results section (4.1).

3.3 Unique/Weird Words and Slang

Another area of interest is whether blogs use a higher proportion of unique or weird words and slang. More than half of bloggers are under the age of 30 with an even split of men and women (Lenhart & Fox 2006). Bloggers form communities of common interest and link to other members of the community. These communities often create an language specific to their particular interests.

The SC reference collection of words used in the spell checking section of this research includes a wide range of words, including American, English and Canadian spelling and jargon. The list was compiled from various sources[15], on the World Wide Web. BNC is used as a reference collection for general English language when calculating weirdness values in this study, as has been done in other research (Gillam & Ahmad 2005).

3.3.1 Spell Checking

The words within each corpus were compared to the SC reference list (described above) of English words to extract uncommon words. These words were placed into a list of 'non-standard' words, they could be words that are specific to a particular community, slang or simply misspelt. The proportion of uncommon words were compared to determine whether a particular corpus is more likely to contain 'non-standard' English words.

3.3.2 Weirdness Values

'Weird' words are either not found in the reference list of words or rarely appear. Words with high frequency and weirdness values are considered high in domain specificity (Gillam & Ahmad 2005). The weirdness values were calculated for each term within each corpus, using the following formula (Gillam & Ahmad 2005), and the results are discussed in section 4.2.

$$weirdness = \frac{N_{GL} f_{SL}}{(1 + f_{GL}) N_{SL}} (Gillam\&Ahmad(2005))$$

where f_{SL} is the word frequency in the corpus, f_{GL} is the word frequency in the reference list and N_{SL} and N_{GL} are the total number of tokens in the corpus and reference list respectively.

4 Results and Discussion

The thirteen corpora analysed in this study were divided into one of three general categories and eight sub-categories:

- Formal writing/News
  - Opinion-Bearing – WSJOp
  - Non-Opinion-Bearing – WSJNop, Reuters, NYT
  - Mixed – WSJMix, BNC, MPQA
- Blogs
  - Opinion-Bearing – BlogOp
  - Non-Opinion-Bearing – BlogNop
  - Mixed – BlogMix
- Webpages
  - Opinion-Bearing – MROp, CRD
  - Non-Opinion-Bearing – MRNop

The indicators analysed in the study show a high level of similarity between the BlogOp and BlogNop corpora. However, the indicators show the BlogMix is different in many areas (detailed throughout this section). This is partly due to the existence of

[14] The Yu & Hatzivassoglou (2003) list comprised of adjectives, nouns, adverbs and verbs. Pronouns has been added to these categories for this research.

[15] http://wordlist.sourceforge.net/, http://www.mieliestronk.com/worklist.html, http://www.outpost9.com/files/WordList.html

Table 4: Mean proportion of part-of-speech categories in the corpora categories

Part-of-speech	Formal			Blog			Webpages		
Category	op	nop	mix	op	nop	mix	op	nop	mix
Adjectives	8.4	7.7	7.9	6.8	7.0	7.6	10.0	8.8	-
Nouns	31.2	34.1	31.3	31.2	32.1	40.1	27.5	32.7	-
Pronouns	4.2	2.9	4.1	5.5	4.9	4.2	5.4	6.6	-
Adverbs	3.5	2.5	3.4	4.4	4.3	3.6	5.6	3.2	-
Verbs	8.5	9.5	9.5	9.9	9.9	8.8	9.1	9.8	-

spam blogs within this corpus. These blogs contain repeated text that artificially inflates the various characteristics. This section discusses the POS proportions and Unique/Wierd words used within each individual corpus, and the characteristics found in the blog sentence corpora are discussed at the end of this section.

4.1 Part-of-Speech Proportions

The QTag part-of-speech tagger was used to tag the content of corpora analysed in this study. The sum of each part-of-speech tag was compared to determine similarities and differences between the various types of corpus. The mean of the proportions for each category (detailed above) was calculated with the following results (detailed in table 4):

- Adjectives – Opinion-bearing webpages recorded the highest mean proportion (10.0%), followed by Non-opinion-bearing webpages (8.8%). The lowest mean proportion recorded was Opinion-bearing blogs (6.8%) and Non-opinion-bearing blogs (7.0%).
- Nouns – Mixed blogs recorded the highest mean proportion (40.1%), followed by Non-opinion-bearing formal writing/news. The lowest mean proportion recorded was Opinion-bearing webpages (27.5%) and Non-opinion-bearing blogs (32.1%).
- Pronouns – Non-opinion-bearing webpages recorded the highest mean proportion (6.6%), followed by Opinion-bearing blogs (5.5%). The lowest mean proportion recorded was Non-opinion-bearing formal writing/news (2.9%) and Mixed formal writing/news (4.1%).
- Adverbs – Opinion-bearing webpages recorded the highest mean proportion (5.6%), followed by Opinion-bearing blogs (4.4%). The lowest mean proportion recorded was Non-opinion-bearing formal writing/news (2.5%) and Non-opinion-bearing webpages (3.2%).
- Verbs – Opinion-bearing and Non-opinion-bearing blogs recorded the highest mean proportion (9.9%) with the lowest mean proportion being recorded by Opinion-bearing formal writing/news (8.5%) and Mixed blogs (8.8%).
- Unclassified – Of the 13 corpora analysed in this study all recorded 0.1% of words that could not be classified, except MPQA and MRNop which recorded 0.0%.

The POS proportions for each corpus was entered into a part-of-speech vector, which was used to calculate a similarity score between the various corpora. When determining similarities between different types of text documents, it is interesting to note that of the individual corpora, BlogOp and BlogNop show very little difference between the POS proportions (0.9997 where 1.0 is exactly the same), while CRD and Reuters show the highest level of difference (0.9307).

When the mean proportions for each corpus category (detailed at the start of this section) are compared, the least similar categories are Blogs Mixed and Webpages Opinion-bearing (0.9386), followed by Webpages Opinion-bearing and Formal writing/news Non-opinion-bearing (0.9507). The most similar is once again Blogs Opinion and Non-opinion (0.9997), followed by Formal writing/new Opinion and Mixed (0.9967). The mean similarity scores are detailed in Table 5.

As the POS proportions do not indicate a pattern over the various types of corpora, each corpus was analysed at an individual word level.

4.2 Unique/Weird Words and Slang

Two collections of words were used as reference collections for this analysis: (1) A collection of words including American, Canadian and English spelling and slang that was compiled from various sources on the World Wide Web[15] (SC reference list) and (2) BNC is used as the reference collection when calculating 'weirdness' (Gillam & Ahmad 2005) scores for the various corpora.

4.2.1 Spell Checking

The word types in each corpus were compared to the SC reference list, to create a list of 'non-standard' words. The proportion of word types appearing in each corpus that do not appear in the reference list is detailed in table 6. The table shows the percentage of word types not in the reference list, the percentage of tokens (word frequency) that the previous figure represents within each corpus and the percentage of those tokens that have a frequency of one within each corpus.

The blog corpora recorded the highest percentage of word types not appearing in the reference list (BlogOp 65%, BlogNop 63%, BlogMix 63%), which represents 4% (BlogMix 6%) of the tokens within each corpus, indicating that blogs use a higher proportion of unique/slang words compared to the other corpora analysed in this study. The low percentage (16%, 5%) of single frequency terms appearing in the blog corpora and not in the SC reference list indicates that unique/slang words are less likely to be 'one-off' uses compared to the other corpora. BlogMix recorded a higher proportion of tokens that were not found in the SC reference list, only 5% of

Table 5: The mean similarity between the part-of-speech vectors. Vectors that are exactly the same would score 1.0000

Corpus		Formal writing/News			Blogs			Webpages	
		Op	Nop	Mix	Op	Nop	Mix	Op	Nop
Formal writing/ News	Op		0.9942	**0.9967**	0.9956	0.9961	0.9831	0.9598	0.9667
	Nop			0.9950	0.9922	0.9936	0.9893	**0.9507**	0.9651
	Mix				0.9937	0.9934	0.9816	0.9718	0.9803
Blogs	Op					**0.9997**	0.9873	0.9563	0.9647
	Nop						0.9898	0.9529	0.9621
	Mix							**0.9386**	0.9566
Webpages	Op								0.9884
	Nop								

the of these tokens are single frequency tokens. The low proportion of single frequency tokens indicates the multiple use of unique words. This is partly due to the inclusion of spam which repeats text multiple times in the same document. To alleviate the problems found in the BlogMix corpus, spam and other repeating text will need to be removed.

Examples of the words and frequency within corpora not found in the reference list are listed in table 7. While many of the words appearing in the corpus and not appearing in the SC reference list are variations of spelling (Eg: 'heeelllooo') or words that run together (Eg: 'ofconservingcanada') other possible explanations for some of the words are listed below:

- abramoff – Jack Abramoff is a former American political lobbyist
- quicklink – QuickLink allows users to manage a set of words for which they would like links to be automatically generated[16]
- usefulrate – a term used when rating something on the Web
- alito – to overcome large amounts of adversity with ease[17]
- zinat – the name of a movie about a woman named Zinat
- korinna – ancient Greek poet or current model

4.2.2 Weirdness Values

Each corpus (other than BNC) was compared to the BNC reference collection and 'weirdness' scores were calculated for each token within the corpus. The tokens of most interest have high frequency and 'weirdness' values. Table 8 details the proportion of word types found in each corpus that are not found in the BNC reference collection. The corpora with less than 1,000,000 tokens have a low proportion of tokens ($< 10\%$) not found in the BNC reference list, with the proportion growing larger as the corpus size increases.

Word types of particular interest have a high score in frequency and weirdness. Table 9 includes an example of some of the word types with high frequency and weirdness, selected from the corpora scoring the highest in weirdness.

[16] http://www.majordojo.com/projects/QuickLink/

[17] http://www.urbandictionary.com/define.php?term=alito

Table 6: Percentage of word types and tokens appearing in the corpus and not appearing in SC reference list. The number of single frequency tokens, and the percentage they represent, appearing in the corpus and not appearing in the SC reference is shown in the third and fourth columns respectively.

Corpus	% types not in ref coll	% tokens not in ref coll	Num single freq tokens	% tokens with single freq.
BNC	40	1	104,737	12
Reuters	24	2	5,327	15
NYT	35	1	122,747	4
MPQA	3	1	125	27
WSJOp	10	1	3,664	40
WSJNop	21	1	6,021	10
WSJMix	33	1	47,724	7
BlogOp	65	4	171,706	16
BlogNop	63	4	134,397	16
BlogMix	63	6	277,195	5
MROp	6	1	574	49
MRNop	8	1	874	53
CRD	7	1	267	36

One problem that becomes evident when looking at word types with high frequency and high weirdness values, is the use of hyphenated words within the corpora (Eg. NYT – star-telegram and WSJMix – year-earlier). This problem is not only evident at the top end of the weirdness list (high frequency/high weirdness), it is spread throughout the list with a high concentration of single frequency word types.

In the small sample shown in table 9, there are words that could be included in a stop-word list (Eg. blog, trackback, nyt). However, simply adding high frequency/high weirdness words to the stop-word list would remove words such as 'lewinsky' and 'netflix', which is problematic as both of these words are possible query terms.

Table 7: Examples of word types appearing in each corpus, and not appearing in the reference list. Table shows the word frequency within the corpus and the proportion of the word within the total tokens in the corpus that were not found in the reference list.

Corpus	Word Type	Corpus Freq.	% of Tokens
BlogOp	abramoff	7221	0.68
	quicklink	3746	0.35
	usefulrate	3135	0.29
	janeane	2552	0.24
	ofconservingcanada	1	0.00
	zweng	1	0.00
BlogNop	usefulrate	3643	0.42
	engadget	3010	0.35
	alito	1598	0.19
	myyahoooorbloglines	1562	0.18
	heeelllooo	1	0.00
	zinat	1	0.00
BlogMix			
	phentermine	85133	1.44
	spyware	76162	1.29
	holdem	60179	1.02
	zzzzzzzzippy	1	0.00
	korinna	2	0.00

Table 8: Word types not appearing in BNC reference corpus, detailing the total number of word types and tokens in each corpus, the number of word types found in the corpus and not found in the BNC reference corpus and the proportion of word types in the corpus that it represents. *The NYT proportion of types not in BNC is high due to the use of 'American' spelling within the corpus.

Corpus	Types	Tokens	Num types not in BNC	% types not in BNC
Reuters	43,963	1,565,380	9,663	22
NYT	830,075	231,856,086	641,395	77*
MPQA	6,867	50,502	213	3
WSJOp	47,939	1,364,326	7,666	16
WSJNop	58,509	4,625,526	15,338	26
WSJMix	288,242	60,402,701	154,727	54
BlogOp	404,131	28,713,436	292,365	72
BlogNop	338,895	19,438,021	236,700	70
BlogMix	866,570	105,824,131	695,515	80
MROp	13,765	100,136	1,150	8
MRNop	14,326	110,283	1,143	8
CRD	5,015	59,317	399	8

4.3 Blog Sentence Corpora

The blog sentence corpora were analysed using the methodology described in Section 3 for POS proportions and Unique words/slang. The OP3 and OP5 were found to be very similar in all characteristics, with the exception that OP5 was a larger corpus, the same was found for NOP3 and NOP5.

The total number of tokens in each corpus is OP3 1,867,411, OP5 2,305,358 (OP5 comprises of 23% more tokens), and NOP3 1,354,630 and NOP5 1,615,991 (NOP5 comprises of 19% more tokens). It may be expected that the difference between a corpus comprised of three sentence blocks compared to five sentence blocks would be approximately 167%[18]. The OP5 corpus comprises of 115% of the word types in to OP3, while NOP5 corpus comprises of 158% of the word types in NOP3.

Table 10 details the results of the analysis of OP3, OP5, NOP3 and NOP5 and compares them to the original blog corpora results. The figures are similar in all categories for the blog sentence except the percentage of word types found in NOP5 and not found in the SC reference list. The proportion is lower than the other corpora proportions, however this is due to there being more word types in NOP5 compared to NOP3.

The distance between the blog opinion and non-opinion corpora in the POS proportions increased in the 'pronouns', 'adverbs', 'adjectives' and 'verbs', while the distance did not change in the 'nouns' category. The opinion-bearing Yu & Hatzivassiloglou (2003) word list created from documents from within the Wall Street Journal corpus, did not include 'pronouns' as a category.

[18] 167% is calculated $\frac{5}{3}$.

The proportion of word types in the various blog corpora and not found in the SC reference list reduced substantially in the blog sentence corpora, while the proportion of tokens found in the corpus and not found in the SC reference list only changed slightly. The proportion of word types with frequency one found in each corpus and not found in the SC reference list doubled (approximately). The weirdness score also decreased dramatically. Along with the proportions changing, the distance between BlogOP and BlogNOP compared to the distance between OP3 and NOP3 increased substantially in these categories with the exception of the weirdness score which did not record a change in the distance between the opinion/non-opinion corpora. Table 10 details the percentage change to the various characteristics measured in this study.

In general the variance between the opinion and non-opinion blog corpora increased, however the difference between the corpus having three sentence blocks and five sentence blocks was not shown in these results. The remainder of this section compares the characteristics of the OP3 and NOP3 to the corpora assessed as being either Opinion or Non-opinion in this research[19]

OP3 is most similar to the WSJOp corpus in the POS proportions (0.993) with the similarity scores being 0.934 (MROp) and 0.942 (CRD) for the other opinion corpora. A major difference between the OP3 corpus and the other opinion corpora is within the 'adjective' category. The MROp corpus recorded the highest proportion of adjectives (11.9%) with CRD (8.1%) and WSJOp (8.4%) both recording a higher proportion compared to OP3 (6.5%). The 'verbs' category also showed a large variation between OP3

[19] Reuters and NYT corpora are not included in this analysis as they have not been assessed as being non-opinion, instead they were assumed to be non-opinion in this study.

Table 9: Word types with high frequency and weirdness

Corpus	Word Type	Corpus Freq.	BNC Freq.	Weirdness
NYT	nyt	111,517	1	23,171
	nytimes	27,403	0	11,387
	coxnet	26,387	0	10,965
	star-telegram	24,938	0	10,363
	lewinsky	24,718	0	10,272
Blog Op	blog	30,088	0	100,965
	trackback	16,083	0	53,969
	permalink	10,417	0	34,956
	netflix	6,437	0	21,600
	google	5,498	0	18,449
Blog Nop	blog	22,015	0	109,126
	permalink	12,615	0	62,531
	google	4,868	0	24,130
	usefulrate	3,643	0	18,058
	url	3,498	0	17,339
Blog Mix	blog	147269	0	134088
	phentermine	82953	0	75528
	spyware	76162	0	69345
	holdem	60179	0	54793
	permalink	50474	0	45956
WSJ Mix	totaling	3637	0	5801
	calif	14251	4	4546
	year-earlier	8272	2	4398
	totaled	5381	1	4291
	bankruptcy-law	2218	0	3538

and the remaining corpora. OP3 recorded 10.3% which is higher than MROp (8.4%), CRD (9.8%) and WSJOp (8.5%).

The proportion of word types in the SC reference list and not in OP3 (36%) is much higher than the other opinion corpora (MROp 6%, CRD 7% & WSJOp 10%), this represents 3% of the tokens within OP3 corpus and 1% of the remaining corpora. The proportion of these word types with a frequency of one ranges between 34–49%, with OP3 recording the lowest proportion.

OP3 recorded the highest level of weirdness (39%), with the remaining corpora recording 16% (WSJOp) and 8% for MROp and CRD. This indicates a high level of domain specific word types being used within the OP3 corpus. Table 11 details the results of the analysis using POS proportions, Spell checking and Weirdness.

When analysing the non-opinion corpora, it was found that once again the NOP3 corpus was more similar to the WSJNOP corpus (0.991) compared to the MRNop corpus (0.955). Similar to the opinion corpora, the proportion of adjectives in the NOP3 corpus (7.1%) was lower than the other corpora (MRNop 8.8% & WSJNop 7.7%), however the proportion of verbs was similar across the non-opinion corpora. The proportion of adverbs was higher in the NOP3 corpus compared to the remaining corpora (MRNOP 3.2% & WSJNop 2%).

The proportion of 49% of word types found in NOP3 and not in the SC reference collection was much higher compared to MRNop (8%) and WSJNop (21%). The proportion of tokens found in the non-opinion corpora and not found in the SC reference list was slightly higher in NOP3 (4%) compared to MRNop (2%) and WSJNop (1%). MRNop recorded the highest proportion of word types with a frequency one (53%) compared to NOP3 (33%) and WSJNop (10%).

As was found in the opinion corpora, NOP3 was much higher in weirdness (38%) compared to WSJNop (26%) and MRNop (8%). This reinforces the belief that blogs are more likely to contain domain specific word types compared to other corpora. Table 11 details the results of the analysis using POS proportions, Spell checking and Weirdness.

5 Conclusions and Future Work

The indicators analysed in this study reveal the opinion and non-opinion blog corpora to be different in their characteristics to the other corpora analysed in this study, especially in the use of non-standard words where a high proportion of words used in the blog documents were not found in standard English word lists. Contrasting this is the corpora collected from sources other than blogs that recorded a much lower proportion of non-standard words. It is expected that opinion identification research training data collected from outside the Blogosphere will not contain the same high level of non-standard words, making it unlikely to produce accurate results when attempting to identify opinions within the blogosphere.

Comparing the results of analysis for the opinion and non-opinion blog corpora indicates very little variation between the two corpora. The similarity of the POS proportions in the two corpora was close to the perfect score (for corpora that is exactly the same). While the distance between the remaining indicators was not substantial. The similarity between the opinion and non-opinion blog corpora is not mirrored by the opinion and non-opinion Movie Review corpora where there was greater distance between the various indicators analysed. One major difference between the blog and Movie Review corpora is that the Movie Review corpora contains opinion or non-opinion sentences. The non-opinion text has been removed from the opinion documents in the MROp corpus.

To determine if a case exists for separating the blog corpora into sentence blocks, the opinion and non-opinion blog corpora were divided into subsets comprising sentences relevant to their given topic. These corpora were compared to the original blog corpora to determine whether the distance between the opinion and non-opinion corpora increased. The distance between the opinion and non-opinion blog sentence corpora generally increased compared to the distance between the original opinion/non-opinion blog corpora. This was particularly evident in the Spell categories which recorded large distance increases in the percentage of word types and the percentage of tokens found in the blogs and not in the SC reference list. This, coupled with the distance increases in the POS categories will lead to more detailed research at the 'word' level in future research.

Comparing the OP3 corpus to other opinion corpora in this study (MROp, CRD & WSJOp)

Table 10: Characteristics of various blog corpora. The 'Variance Change' column indicates the % change in the distance between BlogOP/BlogNOP and OP3/NOP3 in the indicators within this study. *indicates mean document length.

	OP3	OP5	BlogOP	BlogNOP	NOP3	NOP5	Variance Change %
Word types	72,018	83,231	404,131	338,895	65,025	72,581	
Tokens	1,867,411	2,305,358	28,713,436	19,438,021	1,354,630	1,615,991	
Mean Sentence Length	32	26	2,749*	2,347*	29	24	
% Adjectives	6.5	6.5	6.8	7.0	7.1	7.1	6
% Nouns	32.7	32.3	31.2	32.1	33.6	33.6	0
% Pronouns	5.3	5.3	5.5	4.9	4.3	4.3	8
% Adverbs	4.1	4.2	4.4	4.3	3.8	3.8	5
% Verbs	10.3	10.2	9.9	9.9	10.0	10.0	3
Spell							
% Types	36	39	65	63	49	37	33
% Tokens	3	3	4	4	4	4	33
% Single Freq. Word Types	34	33	16	16	33	33	3
Weirdness	39	43	72	70	38	40	0

showed that OP3 was most similar to WSJOp in POS proportions, whilst the Spell and Weirdness categories recorded a large variation between OP3 and the remaining opinion corpora. Of the three opinion corpora in this section of the study, WSJOp content has been assumed to be opinion-bearing text in other research (Yu & Hatzivassiloglou 2003, Kim & Hovy 2005) at the document level, whilst MROp and CRD were assessed at the sentence level. These similarities and differences are repeated in the comparison of NOP3 to the non-opinion corpora in this study (MRNop & WSJNop).

There was however, a lack of variation[20] between the blog sentence corpora with three sentence blocks and five sentence blocks. Whether there is a difference when using the corpora as training data will be determined in future research into opinion identification with blog documents.

Blogs contain a high level of 'non-standard' word types when comparing them to a reference list of either standard English words (BNC) or an expanded list of words containing various spellings of words (English, American, Canadian, etc.), proper names and abbreviations, with a low percentage of these being a singular use of the word. This dramatic variation indicates that blogs use a higher proportion of specific words, demonstrating a substantial variation in the words used within blogs compared to other corpora, and that training data for blog opinion identification should not be extracted from the other corpora.

When asking the question *'Does identifying opinions within blog posts and comments require different training data to identifying opinions within more traditional corpora?'*, it is clear that there is no simple approach to dealing with Blogs. The difference between opinion-bearing and non-opinion-bearing blog documents is not great enough to warrant using blogs assessed at the document level. It cannot be assumed that an entire blog document will contain opinion-bearing words as has been assumed in other research (Yu & Hatzivassiloglou 2003, Kim & Hovy 2005). The high level of 'non-standard' words found within blogs indicates that specific blog training data is needed when attempting to identify opinions within blogs.

[20] Excluding the percentage of word types found in NOP3 and NOP5 and not in the SC reference list.

References

About Technorati, Accessed September (2007). Available: http://www.technorati.com/about/.

Eguchi, K. & Shah, C. (2006), Opinion retrieval experiments using generative models: Experiments for the trec 2006 blog track, *in* E. M. Voorhees & L. P. Buckland, eds, 'The Fifteenth Text REtrieval Conference Proceedings (TREC 2006)', Gaithersburg, Maryland.

Gillam, L. & Ahmad, K. (2005), Pattern mining across domain-specific text collections., *in* P. Perner & A. Imiya, eds, 'Machine Learning and Data Mining in Pattern Recognition', pp. 570–579.

Johnson, D., Malhotra, V. & Vamplew, P. (2006), 'More effective web search using bigrams and trigrams', *Webology* **3**(4).

Kim, S.-M. & Hovy, E. (2005), Automatic detection of opinion bearing words and sentences, *in* 'Natural Language Processing - IJCNLP 2005', Springer, New York.

Lenhart, A. & Fox, S. (2006), Bloggers: A portrait of the internet's new storytellers, Technical report, Pew Internet & American Life Project.

MacDonald, C. & Ounis, I. (2006), 'The trec blogs06 collection : Creating and analysing a blog test collection', *DCS Technical Report Series* p. 8.

MPQA Releases (2007). Available: http://www.cs.pitt.edu/mpqa/.

Table 11: Characteristics of Opinion and Non-opinion-bearing corpora. Note: Similarity scores closest to 1.000 are the most similar.

Category	OP3	MROp	CRD	WSJOp	NOP3	MRNop	WSJNop
Adjectives	6.5	11.9	8.1	8.4	7.1	8.8	7.7
Nouns	32.7	28.7	26.3	31.2	33.6	32.7	35.1
Pronouns	5.3	4.2	6.7	4.2	4.3	6.6	2.1
Adverbs	4.1	5.7	5.4	3.5	3.8	3.2	2
Verbs	10.3	8.4	9.8	8.5	10	9.8	10
Similarity		0.934	0.942	0.993		0.955	0.991
Spell							
Types	36	6	7	10	49	8	21
Tokens	3	1	1	1	4	2	1
Single Freq.	34	49	36	40	33	53	10
Weirdness	39	8	8	16	38	8	26

Nardi, B. A., Schiano, D. J., Gumbrecht, M. & Swartz, L. (2004), 'Why we blog', *Commun. ACM* **47**(12), 41–46.

Ounis, I., de Rijke, M., Macdonald, C., Mishne, G. & Soboroff, I. (2006), Overview of the trec-2006 blog track, *in* E. M. Voorhees & L. P. Buckland, eds, 'The Fifteenth Text REtrieval Conference Proceedings (TREC 2006)', Gaithersburg, Maryland.

Pang, B. & Lee, L. (2004), A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts, *in* 'Proceeding of the ACL'.

QTag probabilistic parts-of-speech tagger (2006). http://www.english.bham.ac.uk/staff/ omason/software/qtag.html.

Rosenbloom, A. (2004), 'Introduction: The blogosphere', *Commun. ACM* **47**(12), 30–33.

The MontyLingua natural language package (2006). http://web.media.mit.edu/ hugo/ montylingua/index.html.

The Stanford NLP Group Loglinear Part-Of-Speech Tagger (2006). http://nlp.stanford.edu/ software/tagger.shtml.

What is the BNC? (2007). Available: http://www.natcorp.ox.ac.uk/corpus/index.xml.

Wiebe, J. (2002), Instructions for annotating opinions in newspaper articles, Technical Report TR-02-101, Department of Computer Science, University of Pittsburgh, Pittsburgh, PA.

Yang, H., Si, L. & Callan, J. (2006), Knowledge transfer and opinion detection in the trec2006 blog track, *in* E. M. Voorhees & L. P. Buckland, eds, 'The Fifteenth Text REtrieval Conference Proceedings (TREC 2006)', Gaithersburg, Maryland.

Yang, K., Yu, N., Valerio, A. & Zhang, H. (2006), Widit in trec-2006 blog track, *in* E. M. Voorhees & L. P. Buckland, eds, 'The Fifteenth Text REtrieval Conference Proceedings (TREC 2006)', Gaithersburg, Maryland.

Yu, H. & Hatzivassiloglou, V. (2003), Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences, *in* 'Proceedings of the 2003 conference on Empirical methods in natural language processing', Association for Computational Linguistics, Morristown, NJ, USA, pp. 129–136.

Zhang, E. & Zhang, Y. (2006), Ucsc on trec 2006 blog opinion mining, *in* E. M. Voorhees & L. P. Buckland, eds, 'The Fifteenth Text REtrieval Conference Proceedings (TREC 2006)', Gaithersburg, Maryland.

Exploratory Multilevel Hot Spot Analysis: Australian Taxation Office Case Study

Denny[1,2] **Graham J. Williams**[3,1] **Peter Christen**[1]

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

[2] Faculty of Computer Science,
University of Indonesia

[3] The Australian Taxation Office,
Email: graham.williams@ato.gov.au

Abstract

Population based real-life datasets often contain smaller clusters of unusual sub-populations. While these clusters, called 'hot spots', are small and sparse, they are usually of special interest to an analyst. In this paper we introduce a visual drill-down Self-Organizing Map (SOM)-based approach to explore such hot spots characteristics in real-life datasets. Iterative clustering algorithms (such as k-means) and SOM are not designed to show these small and sparse clusters in detail. The feasibility of our approach is demonstrated using a large real life dataset from the Australian Taxation Office.

Keywords: self-organizing maps, cluster analysis, neural network, imbalanced data, drill-down, visualization.

1 Introduction

Cluster analysis is often used to help in understanding and dealing with the complexities of large datasets. For example, it may be easier to devise marketing strategies based on groupings of customers sharing similar characteristics because the number of groupings/clusters can be small enough to make the task manageable.

Self-Organizing Map (SOM) (Kohonen 1982) is a popular tool for cluster analysis for several reasons. First, SOM performs topological mapping from high-dimensional data into a two-dimensional map where similar entities are placed nearby. Second, SOM performs vector quantization which produces a smaller representative dataset that follows the distribution of the original dataset. Third, SOM offers various visualizations which are relatively easy to interpret for non-technical users when exploring a dataset. Applications of SOM for cluster analysis can be found in many domains, such as health (Markey et al. 2003, Viveros et al. 1996) or marketing (Dolnicar 1997).

In real life, cluster sizes are normally not equal and clusters do not have the same interestingness. Distribution of clusters is often very skewed as captured by the Pareto distribution (Pareto 1972) also known as the "80:20 rule". Thus, the interesting clusters are usually only a small fraction of a dataset. Furthermore, the variance of items at the tail or margin of the normal distribution of a population is also larger compared to the center of the normal distribution. In other words, in real life it is common to find large dense clusters for common sub-populations and small sparse clusters for interesting sub-populations. In a taxation context this could be a group of tax entities who have a tax debt, while in an insurance context this could be a group of high claiming clients. Williams (1999) proposed the hot spots methodology that aims to identify important or interesting groups in a very large dataset. The methodology uses a combination of clustering and rule induction. As a result, business organizations can make improvements on their strategies, such as treatment strategies to improve tax compliance, by understanding these small and interesting clusters that are called hot spots. It can be interesting to analyze these hot spots in relation to the whole population.

However, iterative clustering algorithms (such as k-means) and SOM tend to merge these small sparse clusters, thus reducing the ability to analyze them in detail. The k-means algorithm tries to generate a relatively uniform distribution on the cluster sizes as shown by Xiong et al. (2006). As a result, k-means is unsuitable for highly skewed datasets.

When SOM is used for cluster analysis, it also has similar issues. Increasing the map size of a SOM only gives a better resolution map (in terms of lower quantization error and finer cluster borders) but with significant additional computational cost. However, an increased map size does not provide extra information about these small and sparse clusters. Small sparse clusters are represented as a few nodes in a SOM, which reduces the capability to characterize them.

Hierarchical clustering algorithms (Han & Kamber 2006), on the other hand, require high computational resources, thus making them impractical for very large datasets. Furthermore, different definitions of between cluster distances (such as minimum, maximum, or average distance) will often produce different clustering results. Moreover, the definition of the between cluster distance has to be determined beforehand.

Therefore, the approach presented in this paper is aimed to help analysts to identify and understand hot spots behaviour. The main contribution of our approach is drill-down hot spot exploration using SOM-based visualizations that capable in handling imbalanced data.

The rest of the paper is organized as follows. Section 2 briefly introduces SOMs and explain their limi-

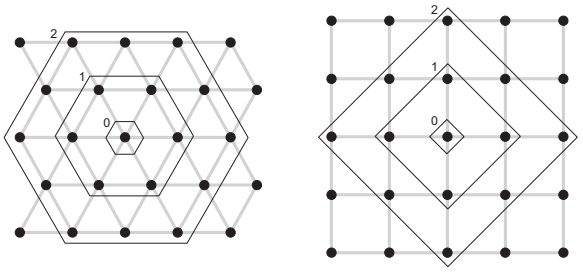


Figure 1: Local lattice structure: hexagonal topology (left) and rectangular topology (right) and its neighbourhood radius in the map space (adapted from Vesanto et al. (2000)).

tation for analyzing hot spots. Section 3 reviews current SOM-based clustering techniques. Our approach is discussed in Section 4 and Section 5 discusses the results of our experiments with a real life dataset from the taxation domain.

2 Self-Organizing Maps

A SOM is an artificial neural network that performs unsupervised competitive learning (Kohonen 1982). Importantly, SOMs allow the visualization and exploration of a high-dimensional data space by non-linearly projecting it onto a lower-dimensional manifold, most commonly a 2-D plane (Kohonen 2001). Artificial neurons are arranged on a low-dimensional grid. Each neuron i has an n-dimensional prototype vector, m_i, also known as a weight or codebook vector, where n is the dimensionality of the input data. Each neuron is connected to neighbouring neurons, determining the topology of the map. In a hexagonal grid, each neuron is connected to six neighbours, while in a rectangular grid each neuron is connected to four neighbours, as shown in Figure 1. In the map space, neighbours are equidistant.

SOMs are trained by presenting data vectors to the map and adjusting the prototype vectors accordingly. These prototype vectors are initialized to different values. There are two approaches to training a SOM: sequential training and batch training. In sequential training, one data vector is presented to the map at a time and the prototype vectors are updated. On the other hand, in batch training, the whole dataset is presented to the map and all prototype vectors are updated at once.

In sequential training, the training vectors can be taken from the dataset in random order, or cyclically. At each training step t, the *Best Matching Unit* (BMU) b_i for training data vector x_i, i.e. the prototype vector m_j closest to the training data vector x_i, is selected from the map according to Equation 1:

$$\forall j, \qquad \|x_i - m_{b_i}(t)\| \leq \|x_i - m_j(t)\|, \qquad (1)$$

where only non-missing values are used in the distance calculation. Then, the prototype vectors of node b_i and its neighbours are moved closer to x_i:

$$m_j(t+1) = m_j(t) + \alpha(t) h_{b_i j}(t)[x_i - m_j(t)], \qquad (2)$$

where $\alpha(t)$ is the learning rate (a tuning parameter) and $h_{b_i j}(t)$ is the neighbourhood function (often Gaussian) centered on b_i. This process of updating the prototype vectors is repeated until a predefined number of iteration or epochs is completed. Both $\alpha(t)$ and the radius of $h_{b_i j}(t)$ are decreased after each iteration. Since the time complexity of SOMs is linear in the number of prototype vectors, number of data vectors, and number of iteration, SOMs are able to cope with large and high-dimensional datasets.

In the batch algorithm, the values of new prototype vectors are the weighted averages of the training data vectors that are mapped to m_j and its neighbours, where the weight is the neighbourhood kernel value $h_{b_i j}$ centered on unit b_i (Kohonen 2001). The new prototype vectors are calculated using Equation 3.

$$m_j(t+1) = \frac{\sum_{i=1}^{N} h_{b_i j}(t) x_i}{\sum_{i=1}^{N} h_{b_i j}(t)}, \qquad (3)$$

where N is the number of training data vectors. SOM is capable in handling missing values, as Equation 3 only performs summation and counting of the non-missing values.

The batch algorithm is similar to k-means. The difference is that the batch algorithm uses weights in calculating the new 'centroids' that are based on the chosen neighbourhood kernel function, while k-means assigns the same weight (weight of one for data vectors assigned to a cluster, weight of zero for the rest) when calculating the centroids.

The map is usually trained in two phases: rough training phase and fine tuning phase. The rough training phase usually has shorter training length and wider initial radius compared to fine tuning phase. In the rough phase, the learning rate $\alpha(t)$ and the radius of $h_{b_i j}(t)$ decrease in a faster rate compared to the fine tuning phase.

After a SOM is trained using a real life dataset, the common population is usually located in the center of the map and the remainder at the border, because of the topologically ordering property and the neighbourhood kernel function used in the training. In real life datasets, the remainder of a population usually has a few different characteristics compared to the common population. For example, in a taxation context, entities who rely mainly on salary and wages for income are mapped onto the center of the map since they are the common population. Other entities might have a few variations, such as having salary and wages and interest income; or having salary and wages, interest, and dividend income.

Since we are interested in the hot spots or 'uncommon but interesting clusters', these clusters are usually located at the border of the map. However, SOMs have a problem with an issue called the *border effect* (Kohonen 2001). The neighbourhood definition is not symmetric at the borders of the map. As shown in Figure 1, the number of neighbours per unit on the border and corner of the map is not equal to the number of neighbours in the middle of the map. Therefore, the density estimation for the border units is different to the units in the middle of the map (Kohonen 2001). As a result, the tails of the marginal distributions of variables (normally located at border units) are less well represented than their centers. As we are interested in hot spots, and these hot spots are usually located at the borders of the map, there is a need to address this problem.

Besides the single level SOM proposed originally by Kohonen (1982), there are SOMs with hierarchical structure, such as Hierarchical SOM (Koikkalainen & Oja 1990) and Growing-Hierarchical SOM (Dittenbach et al. 2000). In these approaches, only one node can be drilled down to the next level. The problem of drilling down only one node at a time is that the Voronoi border of the prototype vector in a sparse area might not be a good cut of the entities in a hot spot area. Furthermore, the goal of Hierarchical SOM is to achieve lower computational cost by using a Tree-Structured SOM to find a BMU faster.

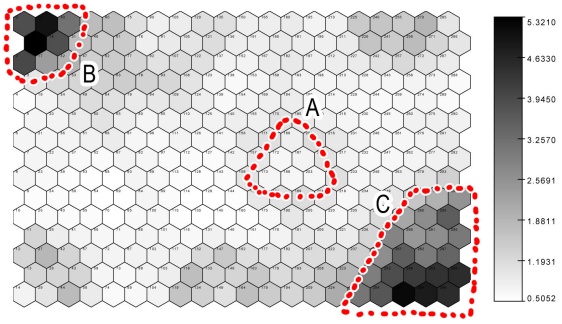


Figure 2: The distance matrix visualization of the whole population dataset, where distance is the median of distances a node to its neighbours.

In our approach, several nodes can be selected to be drilled down interactively by feedback from the user.

3 SOM-based clustering

As mentioned earlier, SOMs perform vector quantization and projection to a 2-D map, and have a topology-retaining property. This makes SOMs suitable for clustering data based on their spatial relationships on the map using visualizations. Existing SOM-based clustering methods can be categorized into visualization based clustering, direct clustering, and two-level clustering (hybrid) as discussed below.

A rough cluster structure can be observed using a distance-matrix based visualization. The distance-matrix based visualization, such as u-matrix visualization (Iivarinen et al. 1994), shows distances between neighbouring nodes using a colour scale representation on a map grid, as shown in Figure 2[1]. As shown in the colour bar, white indicates a short distance between a node and its neighbouring nodes, while black indicates a long distance between the node and its neighbours. The distance matrix visualization methods can be used to show borders between clusters. Long distances that show highly dissimilar features between neighbouring nodes divide clusters, i.e. the dense parts of the maps with similar features (white regions) (Iivarinen et al. 1994). In other words, the distances of the neighbouring units in the data space are represented using shades of colour in the map space.

By using this visualization, users can see the cluster structure of the dense part of the map, for example the center of the map (region marked 'A') in Figure 2. However, it is difficult to see the cluster structure of the sparse parts at the lower-right and the upper-left corners of the map (regions marked 'B' and 'C').

Another method to analyze a hierarchical cluster structure is by using a variant of the data hit histogram that shows how many data vectors are mapped to each node. This is called "Smoothed Data Histogram" (SDH) and proposed by Pampalk et al. (2002). In this visualization technique, each data vector is mapped to its s closest units (BMU) with a linearly decreasing membership degree. The first BMU has a s/c_s degree of membership, the second BMU has a $(s-1)/c_s$, and so forth for the s closest units. The remainder units have zero degree of membership. Pampalk et al. (2002) define $c_s = \sum_{i=0}^{s-1}(s-i)$ to ensure the total membership of each data item adds up to 1. They argue that a hierarchical cluster structure in the data can be observed by changing the value of s. The drawback of this visualization technique is sensitive to the parameter s. The authors did not give any heuristics to choose a suitable value of s. They argued that the optimal value of the smoothing parameter depends on an application. Furthermore, large values of s will give more value to the units at the center of the map due to the topological ordering property of a SOM.

This technique might be able to visualize cluster structure of the dense parts of the map. However, this approach cannot show the hierarchical structure of a sparse part (hot spot) of a map due to the limitation of SOM as described in Section 2.

In direct clustering, each map unit is treated as a cluster, its members being the data vectors for which it is the BMU. This approach has been applied to a breast cancer database (Markey et al. 2003), to a health insurance industry (Viveros et al. 1996) and for market segmentation (Dolnicar 1997).

A disadvantage is that the map resolution must match the desired number of clusters, which must be determined in advance. Furthermore, taking each map unit as a cluster centroid does not guarantee that the clustering result will minimize within-cluster distances and maximize between-cluster distances since SOMs will produce more units for large clusters. Again, this technique cannot show the cluster structure of the sparse part of a map due to the limitation of SOM.

In contrast to direct clustering, in two-level clustering, the units of a trained SOM are treated as 'proto-clusters' serving as an abstraction of the dataset (Vesanto & Alhoniemi 2000). Their prototype vectors are clustered using a traditional clustering technique, such as k-means or agglomerative hierarchical clustering, to form the final clusters. Each data vector belongs to the same cluster as its BMU.

When a SOM is used in the first level of the procedure, it leads to two advantages. Firstly, the original data vectors are characterized by a considerably smaller-sized set of prototype vectors, allowing efficient use of clustering algorithms to divide the prototypes into groups, as shown by Vesanto & Alhoniemi (2000). As a result, this approach is suitable for large or high-dimensional datasets, such as genome data, and for obtaining an initial understanding of possible clusters. For example, after the optimal number of clusters is decided, based on data exploration of the clustering of the maps, clustering with that number of clusters can be performed directly on the data vectors instead of on the prototype vectors, if desired. Furthermore, it allows a visual presentation and interpretation of the clusters via the 2-D grid.

The two-level clustering method also has the same drawback as the previously mentioned methods, as it also uses SOM as the abstraction layer. It is not possible to see the cluster structure of the sparse part of the map, even when using an agglomerative hierarchical clustering on top of the map.

In detecting changes in cluster structure using SOM, Denny & Squire (2005) used two level clustering as described previously and multiple visualization linking to show how clusters change over time, such as emerging clusters, missing clusters, enlarging clusters, and shrinking clusters. Their method were tested using synthetic and real-life datasets using the World Development Indicator data published by the World Bank (World Bank 2003). The results verify that the methods are capable of revealing changes in cluster structure, corresponding to known changes in economic fortunes of countries.

[1]All the SOM figures were originally in colour. For printing purposes, they were converted into gray scale and therefore some details are lost. In the original version, for example, low values are represented as shades of blue and high values are represented as shades of reds.

4 Our Visual Drill-Down Approach

Our visual SOM drill-down approach is applied to the task of exploring taxpayer compliance, in the context of a project with the Australian Taxation Office (ATO) and using a de-identified client dataset. In this section, we discuss data pre-processing, map training, identifying hot spots, and drilling-down the hot spots.

4.1 Dataset

Due to data confidentiality, the complete data description and results cannot be shown in this paper. However, we do provide aggregate indicative results that demonstrate the effectiveness of our approach.

The motivation of the analysis is to understand the logic and structures that drive tax payers' compliance behaviour (behavioural archetypes). The idea is to construct 'psychographic groups' (Wells 1975) by using data mining. Understanding the difference between low and high risk tax payers will be valuable for the ATO.

The archetype dataset consists of about 6.5 million entities with 89 numerical attributes which reflect tax payers behaviour. In general, these attributes can be categorized into: income profile (amount and proportion of each income source), propensity to lodge correctly and on time (lodgement profile), propensity/capacity to pay (debt profile), market segments, demographics, Socio-Economic Indicators for Areas (SEIFA) (Trewin 2003), and involvement in tax avoidance schemes. These attributes were manually selected by the ATO's analysts.

4.2 Data Preprocessing

In distance-based clustering methods, it is important to perform normalization prior to clustering since attributes might have different scale/range (Han & Kamber 2006). Without normalization, attributes with larger ranges will have more influence on the distance measurement. Common normalization technique are: z-score normalization, min-max normalization, and decimal scaling.

In the dataset, we found that some attributes have a large range to variance ratio. When all of the attributes in the dataset are normalized using z-score, the normalized values of these attributes will still have larger ranges.

The range of the z-score normalized value ($range_{A'}$) can be calculated as the range in the original dataset ($range_A$) divided by the standard deviation of the original dataset (σ_A) as shown below. The normalized value v' of attribute A can be calculated by: $v' = \frac{v-\overline{A}}{\sigma_A}$.

$$\begin{aligned} range_{A'} &= max_{A'} - min_{A'} \\ &= \frac{max_A - \overline{A}}{\sigma_A} - \frac{min - \overline{A}}{\sigma_A} \\ &= \frac{max_A - min_A}{\sigma_A} = \frac{range_A}{\sigma_A} \end{aligned}$$

where $\overline{A}$ is the mean, min_A and max_A are the minimum and maximum value of the original attribute values, and $min_{A'}$ and $max_{A'}$ are the minimum and maximum of the normalized values. Therefore, when an attribute has a large range to variance ratio, the range of the normalized value would be high, outweighing other attributes in the distance calculation. Therefore, it is suggested to use a mixed normalization method, such as z-score and min-max normalization, or use weight coefficients in the distance calculation.

As SOMs can only handle numerical attributes, all non-numerical attributes have to be transformed into numerical attributes. Categorical attributes, such as market segmentation and lodgement channel, are converted into numerical attributes by encoding each categorical value into a binary attribute. Furthermore, some numerical attributes that can have negative and positive values are split into two new variables that only contain the positive values or only the negative values to make it easier to interpret the result.

4.3 Map Training

The map is initialized using linear initialization (Kohonen 2001), and trained in two phases using batch training. In linear initialization, the prototype vectors are initialized based on the two largest principal components. Linear initialization is chosen over random initialization because it speeds up the learning process by an order of magnitude by having shorter training lengths (Kaski & Kohonen 1998). Furthermore, linear initialization combined with batch training will produce the same map if the learning process were redone. Random initialization might produce different orientations of the map.

Batch training is chosen because it produces more stable asymptotic values for the prototype vectors and it does not have the convergence problem of sequential training (Kohonen 2001). Furthermore, with a batch training algorithm, it is possible to utilize multi-processor environments to speed up the training process.

The map size, training length, initial and final radius are chosen by considering the best practice, as suggested by Vesanto et al. (2000).

4.4 Identifying Hot Spots in Self-Organizing Maps

Generally, in business, users are more interested in "abnormal clusters" or hot spots (e.g. clusters of entities who have debts) than "normal clusters". Hot spots in SOMs can be identified by two approaches, by using the distance matrix visualizations as well as analysts' feedback based on component plane visualizations.

With the idea that entities in hot spots are usually less homogenous because they are often located at the tail of distributions compared to the common/regular entities, these regions can be identified by using the distance matrix. Using distance matrix visualizations, homogenous groups (low variation) will have shorter neighbour distances (the white regions) compared to high variation groups (the dark regions) as shown in Figure 2. Then, regions that have longer distances should be investigated further by using component plane visualizations.

Component planes show the spread of values of a certain component of all prototype vectors in a SOM (Tryba et al. 1989). The value of a component in a node is the 'average' value of entities in the node and its neighbours according to the neighbourhood function and the final radius used in the final training (Equations 2 and 3). The colour coding of the map is created based on the maximum and the minimum values of the component of the map. In this paper, we use the 'gray' colour map where the maximum value is assigned black and the minimum value is assigned white. Component planes can be used to see interesting cluster patterns and correlations between variables (Himberg 1998, Vesanto 1999)

In Figure 2, there are two hot spots according to the aforementioned criteria, one in the top-left corner (region marked 'B') and another one in the bottom-right corner (region marked 'C'). According the component planes, such as the component plane of the

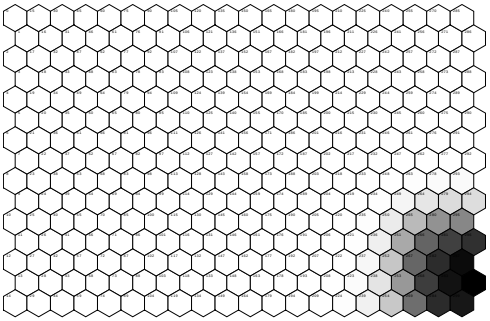


Figure 3: Component plane of 'number of debt cases' of the whole population.

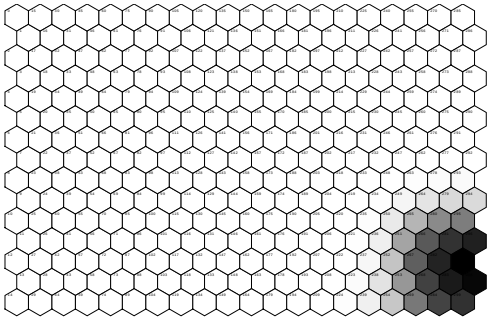


Figure 4: Component plane of 'number of debt cases paid' of the whole population.

number of debt cases as shown in Figure 3, and domain expertise, the hot spot in the bottom-right corner is more interesting than the one in the top-left corner. The bottom-right corner region consists of entities who have debt, have high taxable income, are involved in tax avoidance schemes, and have high risk scores. The top-left corner, on the other hand, consists of entities who received allowances and have more amendments.

The entities in the bottom-right region have highly dissimilar characteristics. However, at this level, it is difficult to differentiate the debt behaviour as shown in Figures 3 and 4. Therefore, it is a good idea to drill down into this region as discussed in the next section.

In identifying hot spots, the domain knowledge of analysts is invaluable because some attributes are more interesting compared to others. In this case, for example: involvement in tax avoidance schemes, lodgement behaviours, number of debt cases, and taxable income, are more interesting in identifying hot spots compared to market segmentation.

4.5 Drill Down and Visualizing Hot Spots

After analysts choose a part of the top level map (distinguish this group as a hot spot) that is interesting to be explored, a sub-map of the region is trained using entities that are mapped to the chosen region. Some issues that need to be taken care of in training the sub-map are: consistency of interpretation of the visualization of the sub-map, and maintaining the sub-map quality with respect to the sub-population.

In order to make interpretation of the visualization of the sub-map consistent to the analysts, the orientation of the map should be preserved and the colour coding should be consistent. The drawback of using linear initialization for the sub-map based on the entities in the sub-map is that the orientation of the sub-map might be different to the orientation of the top level map. For example, the debt entities were located at the bottom-right corner of the top level map but they might be located at the top-left corner as we drill down. This might confuse the user. This could happen when the two largest principal components of the whole population and the sub-population are different.

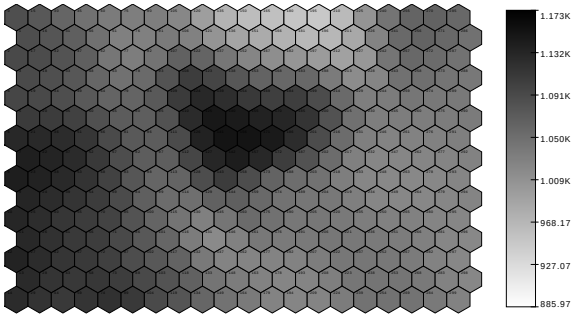


Figure 5: Component plane of SEIFA of the sub-map of region marked 'C' in Figure 2.

Therefore, it is suggested that the top level map is used as the initial map of the sub-map. The radius of the rough phase training should be wide enough, otherwise parts of the map might be empty (no entities mapped to particular nodes). Therefore, as a guide, the initial radius of the rough phase can be half of the longest side and the initial radius of the fine tune phase can be a quarter of the longest side.

The sub-map can be visualized using distance matrix visualization and component plane visualization. In order to show the distribution of values of the sub-map with respect to the whole population, it is suggested that when showing the component planes of the sub-map, the colour map used for the whole population, as described in Section 4.4, is used to visualize the component planes of the sub-map. In other words, black colour in the sub-map visualizations is used for the maximum value of the component of the top level map, not the maximum value of the component of the sub map. For example, Figure 5 shows the distribution of Socio-Economic Indicator for Areas of the bottom-right corner of the whole map. As the sub-map has better quality in terms of quantization error (more homogenous/less variation of the entities mapped to a node), the component value in the sub-map might exceed the maximum value of the whole map. The colour for values more than the maximum value of the whole map would be black as well. Therefore, when a cluster of black nodes appears in the visualization, it is possible that the values are actually exceeding the values of black in the colour bar.

The training of the sub-map will be considerably faster than training of the whole population as the number of data vectors mapped to the region are considerably smaller. Therefore, it is possible for users to explore hot spots interactively.

5 Results and Discussion

To interpret multiple visualizations, analysts need to understand that these visualizations are linked by position or by colour. Visualization of the same map is linked by position, which means that the position of each entity remains the same in each visualization. For example, Figures 2, 3, and 4 are linked by position. Visualization of the whole map and the sub map is linked by colour as described previously. The colour map of the top level map is used as the colour map in the sub-map.

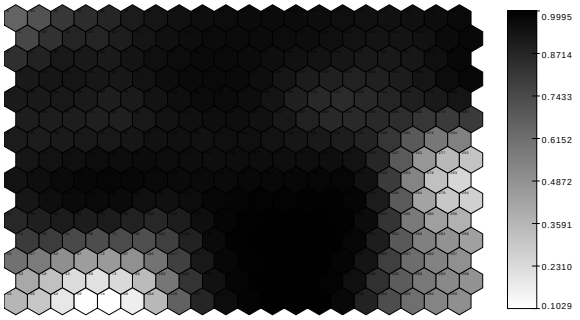


Figure 6: Component plane of 'employee market' of the whole population. Value of 1.0 means that the node consists of 100% employees.

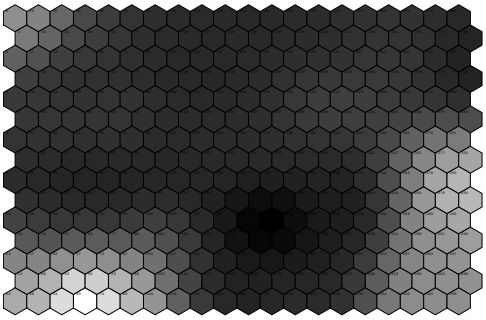


Figure 7: Component plane of percentage of salary and wages to total income of the whole population.

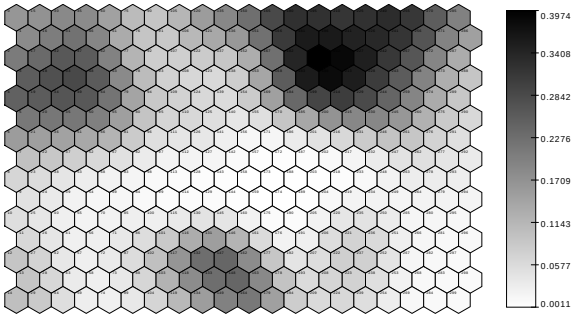


Figure 8: Component plane of 'usage of e-tax lodgement channel' of the whole population.

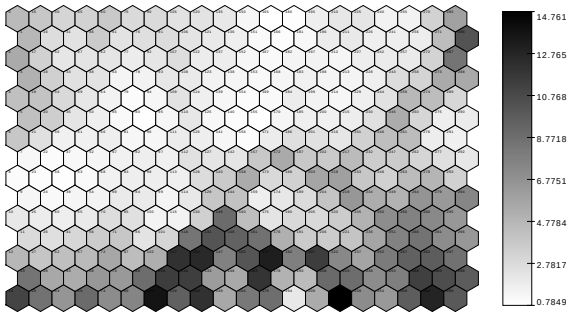


Figure 9: Distance matrix visualization of the sub-map of region marked 'C' in Figure 2.

In our experiments, the map size is 15x30, with hexagonal lattice structure. The initial radius of the rough phase and the fine tune phase are 8 and 4 respectively. The training length for the rough phase and the fine tuning phase are 6 and 10 epochs, respectively. The training processes took about 5 hours on a Debian GNU/Linux machine with two 64-bit AMD dual-core 3 GHz processors and 16 GB memory using our Java SOM Toolbox[2].

As discussed in Section 2, the common population in a real life dataset are usually located in the center of the map. The entities in the center of the map of the whole population are relatively homogenous as shown in Figure 2. Based on the component plane visualizations, this common population mainly consists of employees (Figure 6) with salary and wages as the main source of income (Figure 7).

At this level, we can see that e-tax[3] is an income tax return lodgement channel that is commonly used by employees, as shown in Figure 8. This is as expected since their tax returns generally tend to be simpler. The usage of the e-tax lodgement channel can be further optimized since, as a group, only 40% of the entities mapped to the darkest nodes of the map were using this channel. The information can be useful, for example, in deciding whether to promote e-tax directly to groups of other (similar) tax payers who may benefit from using this lodgement channel.

At the whole population level, it is not possible to differentiate debt behaviours because these entities are mapped to a small number of units at the lower-right corner of the map, as shown in Figures 3 and 4. Debt behaviour can be differentiated by observing debt-related attributes of this sub-population, such as total payment arrangements made, total default payment arrangements, total finalized payment arrangements, and age of debt.

In order to see the debt behaviour in detail, we drill down the lower-right corner of the top level map as explained in the previous section. At this level, we can also use a distance matrix (Figure 9) visualization to highlight the hot spot at this sub-map. In Figure 9, they are located at the bottom of the map.

In the sub-map, we are able to identify a group with characteristics of nearly all of the debt cases paid (Figures 10 and 11) but with a higher stage of compliance enforcement taken by the ATO. It is interesting to note that these entities also live in areas with slightly above average Social-Economic Indicator for Areas (Figure 5) which could mean that they might have the capacity to pay. This kind of analysis is not possible at the whole population level, as these entities are squeezed into a few nodes over the whole map which makes it difficult to differentiate.

It is also interesting to note that the hot spot of the sub-map consists of entities that are involved in

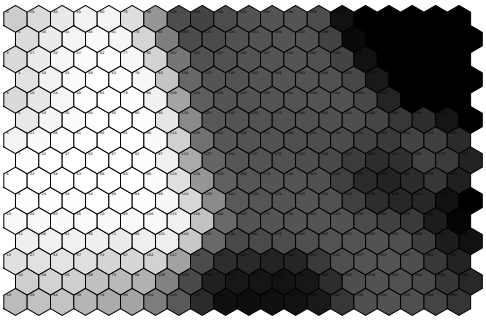


Figure 10: Component plane of 'number of debt cases' of the sub-map of region marked 'C' in Figure 2.

[2]Contact the author if you are interested in using the JavaSOM-Toolbox.

[3]http://www.ato.gov.au/etax

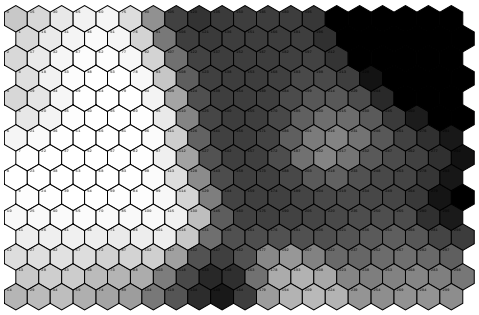


Figure 11: Component plane of 'number of debt cases paid' of the sub-map of region marked 'C' in Figure 2.

tax avoidance activities. Furthermore, this group has characteristics of longer debt age, higher stage of compliance enforcement taken by the ATO, and lower percentage of cases paid.

6 Conclusion and Future Work

We have highlighted the use of SOMs in exploring hot spots in a large real world dataset from the taxation domain. Based on our experiments, our approach is an effective tool for hot spots exploration since it offers visualizations that are easy to understand for non-technical users. Moreover, SOMs are able to handle missing values, are computationally feasible for large datasets, and are able to exploit multi-processor environments. Furthermore, in using our approach, users do not have to determine the number of clusters nor the between-cluster distance definition beforehand.

With our approach, users are able to select regions to drill down, whereas in agglomerative clustering algorithms, the between-cluster distance formula dictate how the population is split. Therefore, the user would be able to select regions/clusters based on their business drivers/needs. This is particularly useful as some attributes have higher importance compared to others.

This work is part of a larger research project where we are interested in observing the dynamics of hot spots over time such as to find entities who are moving in or out of hot spots. Such knowledge would be valuable as the analysts can derive strategies to encourage or to deter people to move in or out of the hot spots; or to evaluate effectiveness of their implemented strategies.

Acknowledgement

This research has been supported by the Australian Taxation Office and the authors express their gratitude to Grant Brodie, Georgina Breen, Nicole Wade, and Warwick Graco for providing key data and domain expertise.

References

Denny & Squire, D. M. (2005), Visualization of cluster changes by comparing Self-Organizing Maps, *in* T. B. Ho, D. Cheung & H. Liu, eds, 'PAKDD'05', Vol. 3518 of *Lecture Notes in Computer Science*, Springer, pp. 410–419.

Dittenbach, M., Merkl, D. & Rauber, A. (2000), Growing hierarchical Self-Organizing Map, *in* 'Proceedings of the International Joint Conference on Neural Networks', Vol. 6, Technische Universität Wien, IEEE, Piscataway, NJ, pp. 15–19.

Dolnicar, S. (1997), The use of neural networks in marketing: market segmentation with self organising feature maps, *in* 'Proceedings of WSOM'97, Workshop on Self-Organizing Maps, Espoo, Finland, June 4–6', Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland, pp. 38–43.

Han, J. & Kamber, M. (2006), *Data Mining: Concepts and Techniques (second edition)*, Morgan Kaufmann, San Francisco, CA.

Himberg, J. (1998), Enhancing the SOM-based data visualization by linking different data projections, *in* 'Proceedings of 1st International Symposium Intelligent Data Engineering and Learning (IDEAL'98)—Perspectives on Financial Engineering and Data Mining', Springer, Hong Kong, pp. 427–434.

Iivarinen, J., Kohonen, T., Kangas, J. & Kaski, S. (1994), Visualizing the clusters on the Self-Organizing Map, *in* C. Carlsson, T. Järvi & T. Reponen, eds, 'Proceedings of the Conference on Artificial Intelligence Research in Finland', Vol. 12, Finnish Artificial Intelligence Society, Helsinki, Finland, pp. 122–126.

Kaski, S. & Kohonen, T. (1998), Tips for processing and color-coding of Self-Organizing Maps, *in* G. Deboeck & T. Kohonen, eds, 'Visual Explorations in Finance with Self-Organizing Maps', Springer, London, pp. 195–202.

Kohonen, T. (1982), 'Self-organized formation of topologically correct feature maps', *Biological Cybernetics* **43**, 59–69.

Kohonen, T. (2001), *Self-Organizing Maps (Third Edition)*, Vol. 30 of *Springer Series in Information Sciences*, Springer, Berlin, Heidelberg.

Koikkalainen, P. & Oja, E. (1990), Self-organizing hierarchical feature maps, *in* 'Proceedings IJCNN-90, International Joint Conference on Neural Networks, Washington, DC', Vol. 2, IEEE Service Center, Piscataway, NJ, pp. 279–285.

Markey, M. K., Lo, J. Y., Tourassi, G. D. & Floyd Jr., C. E. (2003), 'Self-organizing map for cluster analysis of a breast cancer database.', *Artificial Intelligence in Medicine* **27**(2), 113–127.

Pampalk, E., Rauber, A. & Merkl, D. (2002), Using smoothed data histograms for cluster visualization in self-organizing maps, *in* 'Artificial Neural Networks - ICANN 2002: International Conference, Madrid, Spain, August 28-30, 2002. Proceedings', Vol. 2415/2002, Springer Berlin / Heidelberg, pp. 871–876.

Pareto, V. (1972), *Manual of Political Economy*, Macmillan, London. Translated by Ann S. Schwier. Edited by Ann S.Schwier and Alfred N.Page.

Trewin, D. (2003), Socio-economic indexes for areas: Australia 2001, Technical Report 2039, Australian Bureau of Statistics.

Tryba, V., Metzen, S. & Goser, K. (1989), Designing basic integrated circuits by self-organizing feature maps, *in* 'Neuro-Nîmes '89. International Workshop on Neural Networks and their Applications', ARC; SEE, EC2, Nanterre, France, pp. 225–235.

Vesanto, J. (1999), 'SOM-based data visualization methods', *Intelligent Data Analysis* **3**(2), 111–126.

Vesanto, J. & Alhoniemi, E. (2000), 'Clustering of the Self-Organizing Map', *IEEE Transactions on Neural Networks* **11**(3), 586–600.

Vesanto, J., Himberg, J., Alhoniemi, E. & Parhankangas, J. (2000), SOM toolbox for Matlab 5, Report A57, Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland.

Viveros, M. S., Nearhos, J. P. & Rothman, M. J. (1996), Applying data mining techniques to a health insurance information system, *in* T. M. Vijayaraman, A. P. Buchmann, C. Mohan & N. L. Sarda, eds, 'Proceedings of 22th International Conference on Very Large Data Bases (VLDB'96), September 3-6, 1996, Mumbai (Bombay), India', Morgan Kaufmann, pp. 286–294.

Wells, W. D. (1975), 'Psychographics: A critical review', *Journal of Marketing Research (JMR)* **12**(2), 196–213.

Williams, G. J. (1999), Evolutionary hot spots data mining - an architecture for exploring for interesting discoveries, *in* 'PAKDD '99: Proceedings of the Third Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining', Springer-Verlag, London, UK, pp. 184–193.

World Bank (2003), *World Development Indicators 2003*, The World Bank, Washington DC.

Xiong, H., Wu, J. & Chen, J. (2006), K-means clustering versus validation measures: a data distribution perspective, *in* 'KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining', ACM Press, New York, NY, USA, pp. 779–784.

The application of data mining techniques to characterize agricultural soil profiles

[1]Leisa J. Armstrong, [2]Dean Diepeveen, [1]Rowan Maddern

[1]School of Computer and Information Sciences
Edith Cowan University
2 Bradford Street, Mt Lawley 6050, Western Australia.

[2]Department of Agriculture and Food, Western Australia.
3 Baron-Hay Court South Perth, 6151 Western Australia

l.armstrong@ecu.edu.au, d.diepeveen@agric.wa.gov.au, maddern3@bigpond.com

Abstract

The advances in computing and information storage have provided vast amounts of data. The challenge has been to extract knowledge from this raw data; this has lead to new methods and techniques such as data mining that can bridge the knowledge gap. This research aimed to assess these new data mining techniques and apply them to a soil science database to establish if meaningful relationships can be found.

A large data set extracted from the WA Department of Agriculture and Food (AGRIC) soils database has been used to conduct this research. The database contains measurements of soil profile data from various locations throughout the south west agricultural region of Western Australia. The research establishes whether meaningful relationships can be found in the soil profile data at different locations. In addition, comparison was made between current data mining techniques such as cluster analysis and statistical methods to establish the most effective technique. The outcome of the research may have many benefits, to agriculture, soil management and environmental

Keywords: data mining, soil profiles, agriculture

1 Introduction

Data mining software applications, using various methodologies, have been developed by both commercial and research centres. These techniques have been used for industrial, commercial and scientific purposes. For example, data mining has been used to analyse large data sets and establish useful classification and patterns in the data sets. "Agricultural and biological research studies have used various techniques of data analysis including, natural trees, statistical machine learning and other analysis methods" (Cunningham and Holmes, 1999).

This research determined whether data mining techniques can also be used to improve pattern recognition and analysis of large soil profile experimental datasets. Furthermore, the research aimed to establish if data mining techniques can be used to assist in the classification methods by determining whether meaningful patterns exist across various soils profiles characterized at various research sites across Western Australia. Various data mining techniques were used to analyse a large data set of soil properties attributes. The data set has been assembled from soil surveys of Western Australian agricultural areas. The research has utilized existing data collected from ten commonly occurring soil types in order to establish patterns and correlations between a numbers of soil properties. The soils studies which have been conducted by the Western Australian Department of Agriculture (AGRIC) researchers over the past 20 years provide a vast amount of information on the classification of soil profiles and chemical characteristics.

The analysis of these agricultural data sets with various data mining techniques may yield outcomes useful to researchers in the AGRIC. It is envisaged that the information gained from this research will contribute to the improvement and maintenance of soils and the agricultural environment of Western Australia.

The research has a number of potential benefits to the AGRIC and the users of land within the south west land division of Western Australia. The collection and storage of large amounts of data in the AGRIC – Soil Profile Version 3.5.0 database has provided a valuable tool in the study of soils across Western Australia agricultural regions. However, the analysis and interpretation of such a large data set is problematic. This paper outlines research which may establish if new data mining techniques will improve the effectiveness and accuracy of the analysis of such large data sets. The analysis of such soil data sets is difficult given the complex relationships between large numbers of variables collected for each geographical location. The current process to assess soil data uses standard statistical procedures to interpret the soil profile data sets. The use of standard statistical analysis techniques is both time consuming and expensive. If alternative techniques can be found to improve this process, an improvement in the management of these soil environments may result.

The outcomes of the research could improve the management and systems of soil uses throughout a large number of fields that includes agriculture, horticulture, environmental and land use management. The application of data mining techniques has never been conducted for Western Australia soil data sets. A comparison of data mining techniques and statistical methods could produce a model for further understanding the data. In addition, the research could remove the constraining factors that have limited soil scientist's effective utilization of the large amounts of data collected in the last 20 years of research. The benefits of a greater understanding of soils could improve productivity in farming; maintain biodiversity, reduce reliance on fertilizers and create a better integrated soil management system for both the private and public sectors.

The research could be extended in the future with the possible inclusion of additional soil variables; these factors could include other location site information such as climatic data. This could result in the effective uses of soil profile data for the improvement of crop agronomy practices (Moore, 2004, p.3). A new method of interpretation of data could improve knowledge and the methods of data collection, with important factors within the data having been identified. The outcomes of the proposed research could be used for the creation of models of soils within the survey areas that could reduce the cost of data collection by reducing the amount of data collection required in the future.

The purpose of the study was to examine the most effective techniques to extract new knowledge and information from existing soil profile data contained within AGRIC soils databases. AGRIC has collected a large amount of information within its database system; however this data has limited meaning. The study applied data mining methods to a subset of data created by the AGRIC researchers to facilitate an improvement in the interpretation of the soil profile data set.

The Western Australian soil profile data set utilized in the investigation had been selected as it was a representative sample of the data sets population that has been collected over the past 20 years. Mr. Ted Griffin soil scientist for the AGRIC outlined the limitations of time, resources and data complexity to conducting in-depth analysis to date. The data set has allowed each soil type to be compared in a number of geographical locations, for example a loamy gravel soil from Wagin to be compared with loamy gravel from Albany.

It was envisaged that the application of new techniques to the selected data set may overcome the limitations of current soil science research methods. In addition, it may provide a framework of methods that can be applied from the sample population to larger soil databases. The research has overcome a number of problems contained within the data set; one problem was contained with in the data source. The data has been collected from a natural source that contains missing values that could affect results of any experiments conducted and require clearing prior to commencement. The problem that the study aimed to overcome was the selection of the correct methods to apply within the data mining application. In addition, the selection of appropriate data mining techniques is critical in the understanding of the soil profile data. In order for this process to be of some benefit to the understanding of soil characteristics, the findings must be discussed in close consultation with AGRIC statisticians and other soil experts.

The overall aim of the research was to determine whether the application of data mining techniques to an agricultural soil profile data could improve the verification of valid patterns and profile clusters when compared to standard statistical analysis techniques? In order to make this assessment, the research firstly established what current analysis techniques are being used to determine valid patterns and soil profile clusters, secondly, what standard data mining techniques, when used on the data set can establish valid patterns and soil profile clusters and thirdly which data mining techniques are the most efficient in determining patterns and clusters when compared to standard statistical analysis techniques

2 Background

The Western Australian Department of Agriculture (AGRIC) conducted a large scale soil mapping project in the south west of the state in the mid-1980s. This soil mapping project was conducted with the support of the National Soil Conservation Program (NSCP), National Landcare Program (LCP) and Natural Heritage Trust. Current classification techniques to analyse the soil survey data have been outlined (Schoknecht, Tille, and Purdie, 2004). Work by the soil scientist; Purdie in the early 1990s led to the standardization of the methods and outputs of the soil-landscape mapping program (Schoknecht, et al., 2004). This included the development of a nested hierarchy of soil-landscape mapping units. This new method was advantageous as it allowed varying levels of information to be displayed from varying scales of mapping. In addition, the standard classification allowed for possible correlations to be established between different surveys and also enabled computer processing of data on a state-wide and national level. It also provided a means by which the pre-existing survey could be incorporated into a seamless map across the agricultural districts, (Schoknecht et al., 2004).

The Purdie soil classification is the basis of Australian soil classification standards which have subsequently been adopted as the official system (Isbell, 1996). The use of soil classification maps have been shown to play a substantial role in agricultural production, salt control, large scale land management and land improvement. This has allowed a greater understanding of biophysical and environmental management (Schoknecht, 2002). Figure 1 shows the classification of soil types for Western Australia The soil profile data set contains a high level of variability in some survey sites with limited set of experimental data. For example, a reduced number of soil attributes is available for older developed areas, due to testing being carried out at the time the land was cleared.

According to Schoknecht, Tille, Purdie (2004, p. 14) "the soil groups of Western Australia are classified into 60 main groups; this provides a standard way of giving

common names to the main soils of the state". Thirteen soil super groups are defined using three primary criteria: texture or permeability profile; coarse fragments and water regime. Sixty soil groups are defined by further divisions of the soil super groups based on one or more of the following secondary and tertiary criteria: calcareous layer (presence of carbonates); colour; depth or horizons/ profile; pH (acidity/alkalinity); structure.

Further, (Schoknecht, 2002, p.5) outlines the collection of data in the field:

"Soil description is best conducted on an exposed profile such as a pit or road cutting, but alternatively using a soil auger or coring device. In the field, the soil profile is divided into layers (horizons) based on one or more above properties listed above. The properties, depth and arrangement of the layers are used to assign the soil to a soil super group or soil group"

Soil profile data is collected through the exposure of the site, as shown in Figure 2. The soil profile data set used in this research was collected from soil surveys of Western Australia over the last 20 years. A high level of variability was found in this data set. Some surveys sets have limited sets of experimental data, for example a reduced number of soil attributes is available for older developed areas due to testing being carried out at the time of land clearing.

Following the collection of the soil profile data, all data was stored in a central database. Measurements were made based on a visual assessment of the profile, notes on soil location including longitude and latitude and chemical analysis of soil samples taken across the profile site. The data was compiled into a number of different forms within the database with the forms linked by unique keys. The Agriculture WA – soil profile version 3.5.0 database is an MS Access database that allows the collection and extraction of data via a graphical user interface (GUI).

3 Review of Literature

A number of studies have applied data mining techniques to extract meaning from data collected from natural systems research. For example, the collection of data from natural systems is challenging, with most of the data sets incomplete due to the difficulty and methods of data collection. Missing data sets can be problematic and may limit the analysis and extraction of new knowledge. The problem of missing values was analysed by Ragel and Cremilleux (1999, p.1): "To complete missing values a solution is to use relevant associations between the attributes of the data. The problem is that it is not an easy task to discover relations in the data containing missing values."

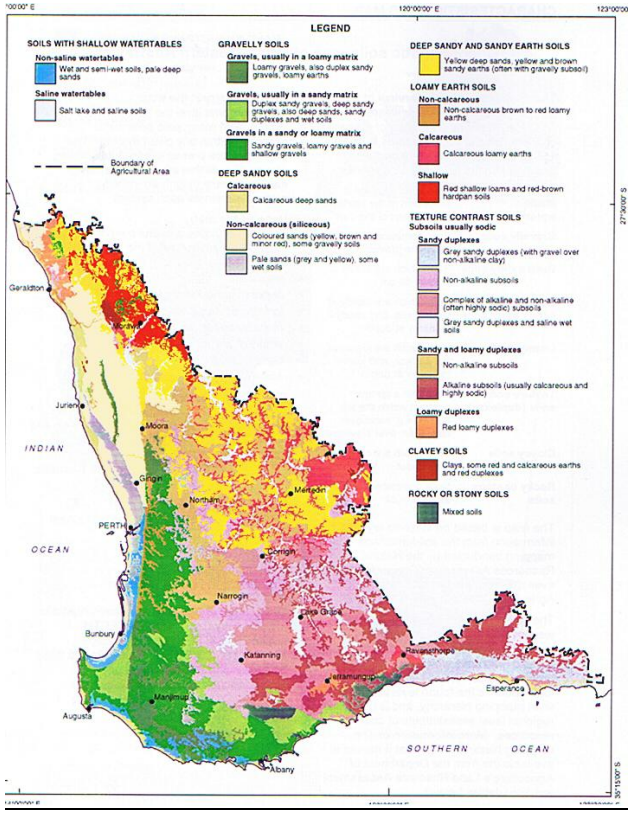


Figure 1: Characteristic of soils of south-western Australia. (Schoknecht, 2002, p.91)

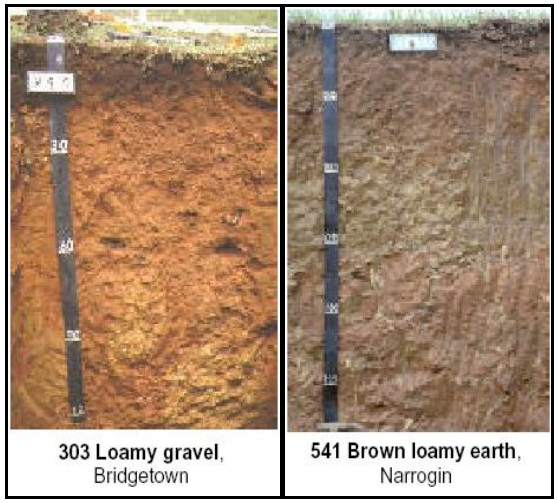


Figure 2: Images of soil profiles of experimental field sites located in the southwest agricultural region. (Schoknecht, 2002, p. 177 and p.121).

3.1 Similar Studies

A number of studies have been carried out on the application of data mining techniques for agricultural data sets. For example, a study by Ibrahim (1999) on a sample data set applied six classification algorithms to 59 data sets and then six clustering algorithms were subsequently applied to the data generated. The results were studied and the patterns and properties of the clusters were formed to provide a basis for the research. The research

provided a comparison of performance for the 6 classification algorithms set to their default parameter settings. It was found that Kernel Density, C4.5 and Naïve Bayes followed by rule learner, IBK and OneR were the most accurate. The study utilized the WEKA data mining benchmark program.

The main objectives of the research conducted by Ibrahim (1999) was to apply unsupervised clustering to the file built in step 1 to analyze the generated clusters and determine whether there are any significant patterns. Ibrahim (1999, p. 2) outlined a number of findings:

It was discovered that number of instances was not useful in clustering the data sets, as it was the only significant variables in clustering the data sets before it was excluded from the generated data set. This prevented analysis based on other variables including the variables that contain values for the accuracy of each classification algorithm.

The research conducted by Ibrahim (1999) has provided a platform from which further work in this field might be undertaken. The scope of the research was limited and the investigation revealed a number of interesting clusters in machine learning performance data. It can be concluded that a larger investigation is required which uses more data sets and data set characteristics.

In another study WEKA was used to develop a classification system for the sorting and grading of mushrooms (Cunningham and Holmes, 1999). The system developed a classification system that could sort mushrooms into grades and attained a level of accuracy equal to or greater than the human inspectors. The process involved the pre-processing of the data, not just cleaning the data, but also creating a test dataset in conjunction with agricultural researchers.

The attributes used to create the set included both objective and subjective measurement. The total dataset used a total of 282 mushroom types, criteria and attributes. The objective attributes were weight, firmness and percentage of cap opening. The subjective attributes were used to estimate the degree of dirt, stalk damage brushing, shrivel and bacterial blotch. The above data was collected and then compared with the grading of the three human inspectors and allocated a grade 1st, 2nd or 3rd.

The data, a total of 68 attributes including photo images, was used by the j4.8 algorithm classifier within WEKA to create a model for the human inspectors and the automated system. The model created using the human rules showed that each inspector used different combinations of attributes when assigning grades to mushrooms (Cunningham and Holmes, 1999). The application of data mining techniques provided within the WEKA software application created a model that analyzed all attributes and created a model that was faster and more accurate than the human system.

The decision tree analysis method has been used in the prediction of natural datasets in agriculture and was found to be useful in prediction of soil depth for a dataset. In Mckenzie and Ryan (1999) the uses of slope angle, elevation, temperature and other factors were analysed and models created for prediction of soil depth across a sample area.

The model was tested through the use of random data sets. “at each level, trees with increasing numbers of terminal nodes were fitted 20 times with 5% of the data randomly selected and withheld to provide a test of the predictive strength of the model” (Mckenzie and Ryan, 1999). This process is outlined in Fig 3.

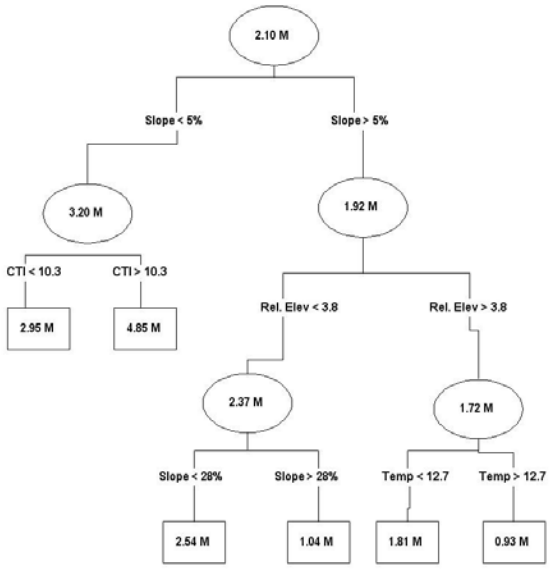


Figure 3: Regression tree. (Mckenzie and Ryan, 1999, p.83)

4 Research Methodology

4.1 Soil Data Collection

The dataset was collected as part of a survey by Schoknecht, Tille and Purdie (2004), and included a large amount of information from different sites within the target area of Western Australia. This information was collected from various locations where a pit was dug and samples taken. The samples were then sent for chemical and physical analysis at the agricultural laboratories in South Perth. The data was then stored in a database with the following information point and site data: “Site Description, soil profile description, soil classification, soil profile chemical properties, soil profile physical properties” (Schoknecht, Tille and Purdie, 2004, p.10). Table 1 describes data collected for each soil sample.

The total number of sites analysed was over 7000, with varying amounts of information obtained for each site. The amount of detail in the database for each given location varied in relation to the period in which it was taken. More in depth information was collected as sampling methods improved. The database was linked to other databases, a map unit database, a soil photos database and map unit polygons.

Field	Description
SOIL CLASSIFICATION	WA Soil Group code
MAP UNIT	Soil-landscape map unit (first three are zone)
AGENCY_CODE	Site's Agency code
PROJ_CODE	Sites Project code
S_ID	Site ID
O_ID	Observation ID (usually = 1 and largely redundant)
SAMP_ID	Sample ID unique identifier for sample taken within a site if null no sample taken
H_NO	Horizon (or layer) ID, from field morphology observations, sequence numbers may be missing
SAMP_H_MATCH	code indicating the degree of matching between the layer depth and sample depth A exact, B sample a subset of layer, C sample crosses layer but predominantly of layer, D other
SAMP_UPPER_DEPTH	sample upper depth
SAMP_LOWER_DEPTH	sample lower depth avDepth sample average depth
CACO3	CaCO3 %
CACO3_imp	HCl fizz test where from field observations N nil, S slight, M moderate, H high, V very high
OC	Organic Carbon %
PH	pH in CaCO3
Clay	clay %
EC	EC, ms/m
ExCA	Exchangeable Ca cmol(+)/kg
ExMG	Exchangeable Mg cmol(+)/kg
ExK	Exchangeable K cmol(+)/kg
ExNA	ExCEC CEC cmol(+)/kg
ExSUM	Sum Exchangeable Ca, Mg, K, Na cmol(+)/kg
ExESP	Exchangeable Na % of Sum
ExH	Exchangeable H cmol(+)/kg
ExMN	Exchangeable Mn cmol(+)/kg
ExAL	Exchangeable Al cmol(+)/kg
ExSAT_PC	Saturation % (100*Sum/CEC)
ExBASE	Base Status (100*Sum/clay)
ExCaP	Exchangeable Ca % of Sum
ExMgP	Exchangeable Mg % of Sum
ExKP	Exchangeable K % of Sum

Table 1: Data field descriptions. (Griffin, 2005)

The mapping of soil data set was conducted over a number of years, with the first survey undertaken in the 1930s of the area around Salmon Gums by Burvill and Teakle for the CSIRO. Since then a number of surveys have been conducted by the CSIRO and AGRIC for a number of locations within Western Australia. The problems that have arisen from these surveys are the scale and the amount of chemical analysis conducted for each. Until a standardized method was introduced analysis methods were not uniform due to the large volume of samples taken not all chemical testing was conducted for all locations. The methods of collection have generated gaps in the data set and not all samples for all locations contain all possible values.

4.2 Soil Classification

The classification of the soils was considered critical to the study because the soil types must be the same in all locations across the study area for the results to be accurate. The soils were classified according to work by Schoknecht, (2002, p. 5); that outlined the technique for the grouping of soil types. They are:

1. Soil super groups: Thirteen soil super groups are defined using three primary criteria: Texture or permeability profile, coarse fragments *(presence and nature) and* Water regime.

2. Soil groups: Sixty soil groups are defined by further divisions of the soil super groups based on one or more of the following secondary and tertiary criteria:

> Calcareous layer (presence of carbonates): colour, depth or horizons/ profile Ph (acidity/alkalinity) and structure.

4.3 Data mining Process

The data mining process was conducted in accordance with the results of the statistical analysis. The following steps are a general outline of the procedure that allowed a cluster analysis to be conducted on the dataset:

4.3.1 Data collection cleaning and checking

Relevant data was selected from a subset of the DAFWA soil science database.

4.3.2 Data formatting

The data was formatted into an Excel format from the Access database, based on the ten soil types and relevant related fields. The data was then copied into a single Excel spread sheet. The Excel spread sheet (ESS) was then formatted to replace any null or missing values in the soil data set to allow coding for the file in the next phase.

4.3.3 Data coding

The soil data set was then converted into a comma delimited (CSV) format file for the ESS. This file was then saved and opened using a text editor. The text editor was used to format and code the data into the type that will allow the data mining techniques and programs to be applied to it. The coding was formatted so that the input will recognize names of the attributes, the type of value of each attribute and the range of all attributes. Coding was then conducted to allow the machine learning algorithms to be applied to the soil data set to provide relevant outcomes that were required in the research. The data

coding attributes were named in line with the data table shown in Table 1.

4.3.4 Case studies

The soil data set was then broken down to five profiles;

1. One soil – one trait.

2. One soil – two traits.

3. Two soils – one trait.

4. Two soils – two traits.

5. All soils – all traits.

Grey deep sandy duplex and loamy gravel were used as these soils and their traits contained the maximum number of values within the data set. The traits that contained the highest number of values were clay and EC and these were used in the first four stages with clay used in single trait instances. The sub data set were then applied to the expectation-maximization (EM) algorithm and FarthestFirst algorithm. The clustering data was then collected including means and standard deviation to determine algorithm accuracy against actual values.

4.3.5 Analysis and review of outcomes

Comparison and review of the experiments were conducted according to the methods referred to in figures 4 and 5.

The research has a number of limitations that could impact on the results achieved. These were outlined by Palace (1996) who stated that the applications of data mining techniques improve as the data set size increases. The soil data set has over 2800 sets of data with an average 12 measurements per geographical location. The size and type of the dataset was a major limitation. The accuracy of any analysis technique increases with the amount of data contained in the dataset. This may be in part due to the patterns that are contained with the data. More patterns eluded may help to define a stronger relationship and more reliable results. The size of the data set analysed was limited because of the short time available for the research and the ability of the human interpretation of the outcomes because of the complexity of the dataset used.

4.4 Analysis of Data

The research adopted action research methodology, where improvement and changes may have to be undertaken to provide the DAFWA with outcomes that meets their required specifications for the project.

The research used Excel software to conduct qualitative analyses and to create a benchmark for the analysis of the dataset. The benchmark allowed current statistical methods for the dataset to be established and any limitations to be identified. The dataset was then analysed using a clustering process within the data mining software. The results were then compared against the benchmark for a number of factors that included ease of application, speed, time and accuracy of results to determine if data mining was superior to current methods. The results of statistical and data mining experiments may still require some expertise to be understood and utilized.

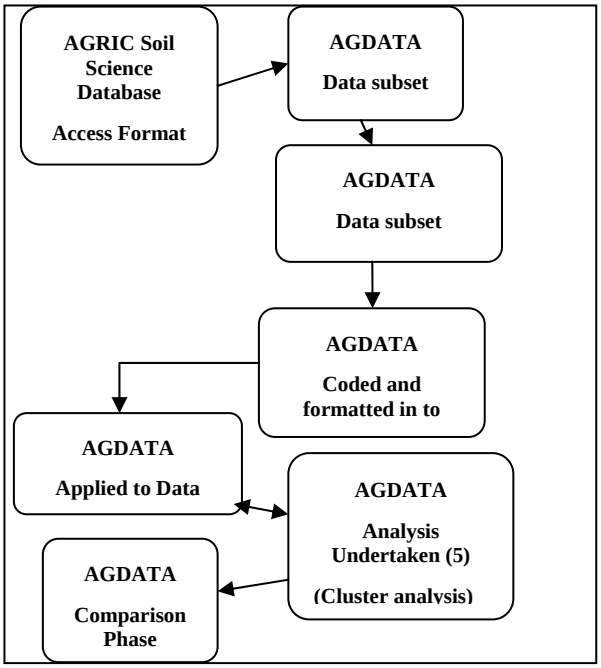


Figure 4: Experimental technique: Data mining

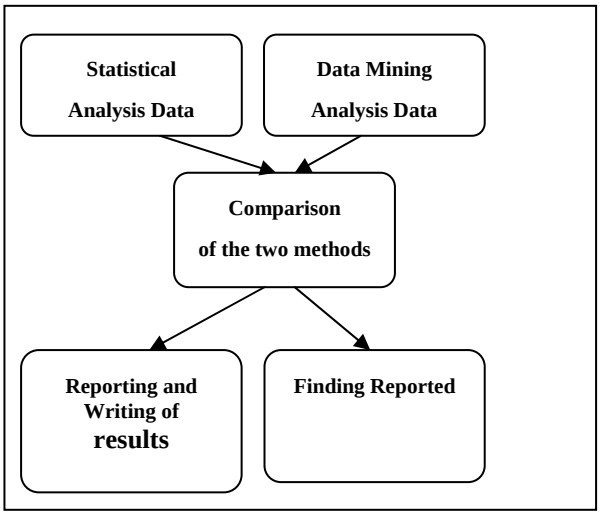


Figure 5: Data mining vs. traditional statistical methods.

5 Results

The analysis and interpretation of patterns is a time consuming process that requires a deep understanding of statistics. The process requires a large amount of time to complete and expert analysis to examine any patterns and relationships within the data.

5.1 Statistical results

The research activities involved a process to establish if patterns could be found in the data. These processes involved the statistical manipulation of the data set in Excel. The aim of the research was to determine if a relationship or correlation can be established with soil

trait data. The process involved the creation of analysis tools and charting the data so that longitude and latitude data and trait data is displayed and experts can interpret the findings.

The initial statistical data analyses involved four processes:

- Raw data traits plotted against longitude and latitude for each sample location using a 3D surface map.
- Standardized traits plotted against longitude and latitude with the data levelled using the minimum trait value to level the data for plotting in the 3D surface map.
- Correlation table analysis.
- Regression correlation analysis.

The process of plotting data required expert analysis for a relationship to be established. Such analysis was conducted in conjunction with Mr. E.A Griffin, Soil Scientist for the Department of Agriculture and Food, Western Australia.

The dataset was constructed from the DAFWA soil science data was designed to collect repetitive samples of the data contained in the south western agricultural region. The total data set contained 493 sites with an average of 5 samples taken for each location with a total of 2841 sample sets taken. The samples were analysed for a possible 41 traits but very few sets were complete for all data. The total number of data points possible was 116,481 but due to missing values the total number of data points considered was 34881.

Stage 1: Initial raw data analysis

Data was processed from raw data into a single Excel spread sheet with the inclusion of an elevation and longitude and latitude data for each sample.

The creation of a 3D surface map requiring the latitude and longitude to be rounded to a single decimal place prior to the data insertion into a pivot table. The data was rounded by using the following formula:

(=Round (Round number, number of decimal places))

The rounded longitude and latitude was then inserted into a new spread sheet with the individual traits e.g. (Lo-Lat-CaC03) for CaC03 and the samples with missing longitude and latitude data were removed.

The longitude – latitude – trait data was then selected and a pivot point table created in a new sheet e.g. (Lo-La-Cac03 (PP-G)) where the '*PP*' stands for pivot point table and '*G*' stands for graph. (see Table 2 and 3) .

Longitude	114.2	114.3	114.4	114.5
Latitude		Trait	Trait	Trait
-35	1.333334	Trait	Trait	Trait
-34.9	Trait	Trait	Trait	Trait
-34.8	Trait	Trait	Trait	Trait

Table 2 Pivot Table of Samples Latitude vs Longitude vs Traits

The pivot point table was then formatted, by removing the column and row totals and then the count was changed into an average so that the fields were representative of the data. The pivot point table was then copied and all data was formatted to two decimal places to allow ease of analysis.

Longitude	114.2	114.3	114.4	114.5
Latitude		Trait	Trait	Trait
-35	1.33	Trait	Trait	Trait
-34.9	Trait	Trait	Trait	Trait
-34.8	Trait	Trait	Trait	Trait

Table 3 Pivot Table of Samples Latitude vs Longitude vs Traits

The formatted data was then graphed using a 3D surface map with longitude (X), latitude (Y) and trait values (Z), an example of this is shown below in figure 5. The process was repeated for all soil traits (CaC03 – ExKP).

Figure 5 displays an example of one soil trait; the levels of Electrical Conductivity for the soils in the south west; The results indicate that there are high levels in the south east Agricultural region of Western Australia.

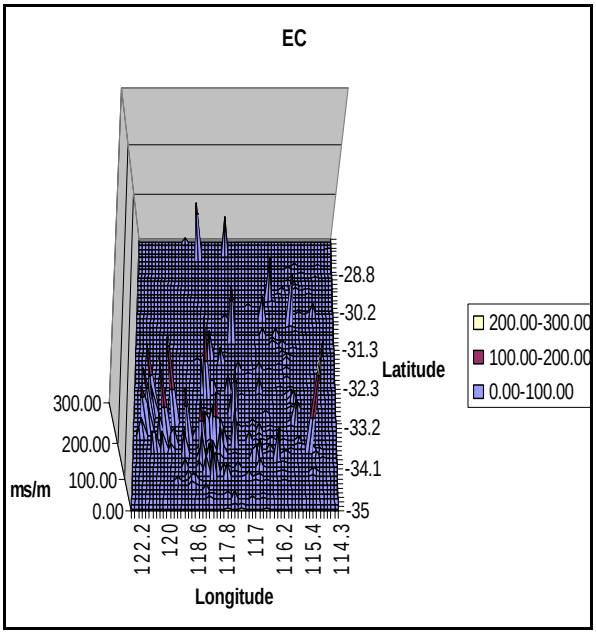


Figure 5: Normalized data of Electrical Conductivity trait for all soil types

Stage 2: Standardized data

This stage involved the further analysis of the soil data.

The longitude and latitude data was rounded to 1 decimal place, using (=Round (Row number, number of decimal places)).

All soil trait data was then standardize using the following formula

=IF(Data.V2="";" ";STANDARDIZE(Data.V2;AVERAGE(Data.V$2:V$2842);STDEV(Data.V$2:V$2842)))

The data was then isolated as per stage 1 and place into a worksheet for each trait, sheet name given by trait name (e.g. Calcium Carbonate content- CaC03). Creation of a second set of pivot point tables for each trait, longitude and latitude and placed into a new worksheet (e.g. CaC03 (PP-G)). The pivot point table then had the row and column totals removed and the fields changed from count to average. The pivot point table was then copied to obtain the raw data and the table formatted to two decimal places.

The minimum data value was obtained and used to establish a baseline, with the lowest value of the trait added to trait dataset. Longitude and latitude data was then set out in a new table as per the other tables, with the above formula filling the trait section of the table. The data was then graphed using the 3D surface charts, latitude (X), longitude (Y) and trait value (Z). The process of adding the individual standardized trait values to the minimum value displayed the data with a plan of minimum values with peaks on both sides of the plan. An example of one trait; electrical conductivity is shown below in Figure 6. The data shows that there is a high concentration on the western coastline, located in the Margaret River region of the state. The data shows that there is a comparison between the normal (figure 6) and standardized data graphs and that there may be a significant correlation.

Stage 3: Correlation Analysis

The next stage in the research investigated the possible correlations that may exist within the data set. A correlation table was created from the basic soil dataset. The correlation table outlines the relative relationships between all traits in the dataset. The correlation table indicated that there were a number of strong relationships between the traits; these results are show in table 4.

The current method was then applied to a subset of the soil data set; the subset contained the three main soil types that had the greatest number of geographical locations in the survey area. The three main soil types were: grey deep sand duplex, loamy gravel and pale deep sand. The analysis process was repeated because of a request from researchers at the DAFWA. The process was repeated with the exception of the creation of the correlation table, as that was not required with the limited amount of data. The overall process was very time-consuming and repetitive, with the research requiring seven days to complete the whole process with a large amount of help and feedback from the statistician. The research, in addition to being time consuming and complex, required a large amount of human input and interaction to complete the process. The process was designed with the aid of an agricultural statistician from the DAFWA to ensure that the analysis is true to the process presently used at the DAFWA.

5.2 Data mining results

The benchmark having been established, the data analysis was then replicated using WEKA data mining software to determine if any advantage could be gained in both time saving and interpretation of the soil data set. The application of the data to WEKA required that some pre-processing be undertaken. The dataset produced in Excel for the statistical processes were copied and then converted to .CSV file format to allow them to be applied to WEKA. The .CSV file extension allowed initial analysis to be conducted, with later conversion to be taken in to an ARFF WEKA data file for the experimental outcome to be saved.

The data mining platform allowed number of data interpretations including classify, cluster, and associate routines to be conducted after the pre-processing stage. The soil data set did not require any filtering because of the limited amount of missing values and the outcomes required by the researchers. The initial screen provided a set of information that is required by the researchers and took a large amount of time to complete with the current statistical methods.

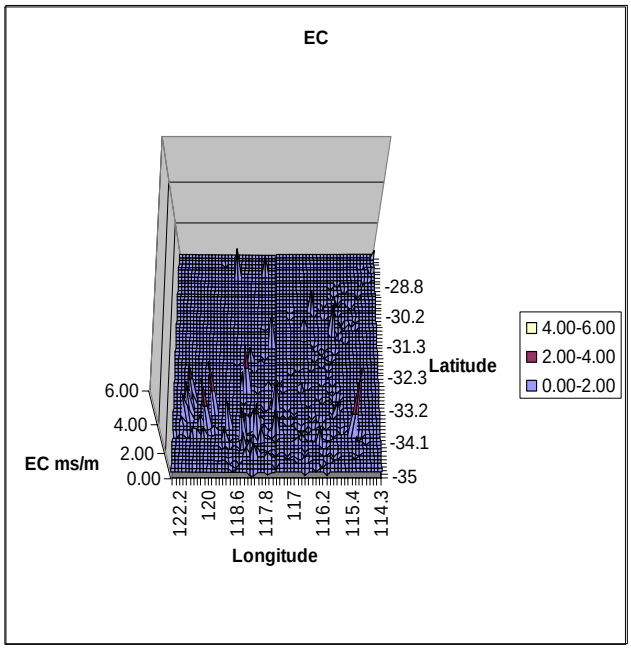


Figure 6: Standardized data of EC for all soil types, EC all soils

The full soil data set was applied to the EM-1 100-N-1-5 100-M 1.0E6 and FarthestFirst clustering algorithm to see if any patterns could be established with the model being constructed using a training model to build the associations in the data. The clustering algorithms outlined above perform their operations in two different ways, and the differences between the two will be used to

determine the accuracy when compared with each other. The expectation-maximization (EM) algorithm was outlined by the WEKA data mining software and provides a basic outline of the algorithms operation. EM assigns a probability distribution to each instance which indicates the probability of it belonging to each of the clusters. EM can decide how many clusters to create by cross validation, or you may specify how many clusters to generate.

The cross validation performed to determine the number of clusters is done in the following steps:

1. The number of clusters is set to 1

2. The training set is split randomly into 10 folds.

3. EM is performed 10 times using the 10 folds the usual CV way.

4. The log likelihood is averaged over all 10 results.

5. If log likelihood has increased the number of clusters is increased by 1 and the program continues at step 2.

The number of folds is fixed to 10, as long as the number of instances in the training set is not smaller 10. If this is the case the number of folds is set equal to the number of instances. The EM algorithm required that some of the parameters be changed to ensure it produced the same amount of clusters as the FarthestFirst to allow analysis to be undertaken. This was simply done by outlining the number of outputs in the options before applying EM to the individual dataset.

The WEKA data mining software outlined the operation of the FarthestFirst algorithm and clusters using the farthest first traversal algorithm. It works as a fast simple approximate cluster and is modelled after SimpleKMeans. The stages of analysis were conducted using both algorithms and the data collected and formatted below. The comparison of the accuracy of each clustering method was determined by analysis of the grouping numbers, means and standard deviation.

The results of the experiments are shown below with each case study having two tables. The first table displays the algorithm name in the first column and outlines the number of clusters created in the second column, with the number of occurrences. The second table displays the clusters with their means and StdDev created by both of the clustering methods. Each case study also includes data about the actual number of data points in the data set; this was included to determine the accuracy of each algorithm. The first four case studies were designed to limit the amount of data input so that the results were simple to understand. The last case study included all data from the original soil data set for completeness.

The data from each of the two algorithms varied, with EM producing more data than FarthestFirst, and so the last case study contains four large tables of means and StdDev because of the seven clusters created. The analysis of the output involved looking at the cluster percentage and the number allocated to each cluster in comparison with the actual dataset. Case study five, because of the large amount of information produced by the experiment, will not be analysed and the results and discussion will focus of the first four case studies for comparison to the current statistical methods.

The results of the data mining experiments have shown that FarthestFirst algorithm was equal to or more accurate at clustering compared to the EM algorithm in all four tests. The results showed that the FarthestFirst results for case study 2 and 3 grouped the data correctly with a minimum of error. The accuracy of the EM algorithm much less when compared with the actual data points and only grouped the data correctly in case study 3, with the other three case studies clustering much less accurately.

Case study 1: (One soil type – 1 trait - grey deep sand duplex - Clay)

Case study one compared the clay trait for all of grey deep sand duplex soil and it was found that the EM algorithm grouped the two clusters in half. The FarthestFirst algorithm weighted the groups more to cluster 0 and that reflected the actual data with more accuracy with a single group of 536 instances.

		Trait 1
EM	0	268 (50%)
	1	268 (50%)
FarthestFirst	0	417 (78%)
	1	119 (22%)

	Trait 1
EM	0 Mean = 3.9336 StdDev = 1.745
	1 Mean = 34.6761 StdDev = 15.4472
FarthestFirst	0 centroid = 1.0
	1 centroid = 75.0

Clay = 536 instances

Case study 2: (One soil type – 2 traits - Grey deep sand duplex – Clay and EC)

Case study two compared grey deep sand duplex with traits clay and EC and it was found that EM grouped the two clusters more accurately than case study one. EM grouped the instances more in line with the actual data but it was found that FarthestFirst still was more accurate. FarthestFirst grouped cluster 1 correctly with EC having 480 instances, but there was still some error with clay instances grouped in cluster 0 with 541 instances and actual data having only 536.

		Trait 1
EM	0	610 (60%)
	1	411 (40%)
FarthestFirst	0	541 (53%)
	1	480 (47%)

	Trait 1
EM	0 Mean = 4.052 StdDev = 2.417
	1 Mean = 37.223 StdDev = 30.9041
FarthestFirst	0 centroid = 35.7 Clay
	1 centroid = 350.0 EC

Clay = 536 instances

EC = 480 instances

Case study 3: (Two soil types – one trait - Grey deep sand duplex, Loamy gravel (CLAY)

Case study three compared two soil types (grey deep sand duplex and loamy gravel) with one trait (clay). Both the algorithms group the instances correctly with grey deep sand duplex and having 536 instances of clay, and loamy gravel having 612 instances of clay.

		Trait 1
EM	0	536 (47%)
	1	612 (53%)
FarthestFirst	0	612 (53%)
	1	536 (47%)

	Trait 1
EM	0 Mean = 19.7913 StdDev = 18.9894
	1 Mean = 21.7818 StdDev = 15.3254
FarthestFirst	0 centroid = 47.35 Loamy gravel clay
	1 centroid = 0.0 Grey deep sandy duplex clay

Grey deep sandy duplex (Clay = 536 instances)

Loamy gravel (Clay = 612 instances)

Case study 4: (Two soils types; Grey deep sand duplex, Loamy gravel – two traits – (Clay, EC))

Case study four compared two soil types (grey deep sand duplex and loamy gravel) and two traits (Clay and EC). The results show that EM grouped the instances weighted towards cluster 2 at 991 instances and weighted cluster 0 less, having only 34 instances. FarthestFirst was more accurate and only wrongly classified one instance out in clusters 1, 2 and 3.

		Trait 1
EM	0	34 (2%)
	1	582 (27%)
	2	991 (46%)
	3	562 (26%)
FarthestFirst	0	536 (25%)
	1	480 (22%)
	2	541 (25%)
	3	612 (28%)

	Trait 1
EM	0 Mean = 101.6457 StdDev = 58.0094
	1 Mean = 34.9632 StdDev = 14.236
	2 Mean = 2.916 StdDev = 1.744
	3 Mean = 10.3491 StdDev = 4.6035
FarthestFirst	0 centroid = 52.0 Grey deep sandy duplex clay
	1 centroid = 350.0 Grey deep sandy duplex EC
	2 centroid = 146.0 Loamy gravel EC
	3 centroid = 0.3 Loamy gravel clay

Grey deep sandy duplex (Clay = 536 instances)

Loamy gravel (Clay = 611 instances)

Grey deep sandy duplex (EC = 479 instances)

Loamy gravel (EC = 540 instances)

Case study 5: (ALL – DATA)

The data was too complex to conduct analysis in the limited time available and was done for completeness of the research on advice from the statistician. Note that table two outlines all the seven clusters created by the two algorithms and the instance allocation for each.

		Trait 1
EM	0	342 (12%)
	1	340 (12%)
	2	95 (3%)
	3	420 (15%)
	4	464 (17%)
	5	323 (12%)
	6	320 (11%)
	7	491 (18%)
FarthestFirst	0	1154 (41%)
	1	126 (5%)
	2	779 (28%)
	3	337 (12%)
	4	261 (9%)
	5	69 (2%)
	6	69 (2%)

Figure 7 and 8 provide examples of the cluster analyses performed for 2 different soil types based on different chemical traits.

5.3 Summary of Results

The two analysis processes both provided a method by which soils data analysis can take place. The results of the research suggest that clustering may be an effective tool for the comparison of soil types, traits and locations within the study area of the south west agricultural region of Western Australia. The application of statistical methods required a large amount of time to complete and to produce a usable outcome for soil researchers. It was evident that the data mining required less time and knowledge to complete an analysis process, compared with current statistical processes, but the interpretation of the data still requires expert human analysis from the DAFWA.

6 Discussion

The collection of information and data has increased with the advent of new computing technology, but establishing patterns within this data has become more difficult and requires new approaches and tools if it is to be undertaken. The advent of this problem has provided an opportunity from which data analysis has started to take over from current methods. Furthermore, this technology has reduced the time taken to undertake data analysis and has increased automation of the process.

The research undertaken showed that data mining has advantages and can be easily applied to the soil data set to establish patterns in the data. The application of the WEKA data mining platform provided an easy and quick method for the cluster analysis. The platform provides a number of clustering algorithms that can be used for different tasks. The experiments described in this paper used two of these clustering algorithms, EM and FarthestFirst to determine the most accurate when compared with actual results.

The integrity of the data is critical to ensure that results are not affected by outliers and null values in the data set, or other adverse factors. The establishment of clusters in the data required a large amount effort by the researchers when using current methods. Furthermore, the current methods still required some post Excel analysis because the platform was limited in the interpretation of the graphs generated. The application of the same clustering techniques using the data mining software reduced the time taken to process the data sets; with the process time reduced to one day, and also allowed a greater amount of knowledge to be gained from the data.

6.1 Evaluation of statistical methods

Current statistical methods provided a platform from which analysis of agricultural soil profiles could be undertaken. The application of these methods has demonstrated accurate analysis of clusters and patterns when used on soil science databases. However, the current technique involves an in-depth understanding of statistics and requires a large amount of human input and time to complete. The current statistical methods that are being used to determine valid patterns and soil profiles clusters are 3D surface mapping and basic statistical methods including correlation tables and distribution analysis.

Increases in the amount of data collected from field experiments have meant that the time and complexity has increased to the point that it has become difficult to obtain new knowledge. The experiments undertaken during this research required expert input and provided a large amount of graphical data and statistical analysis. The processes used in the experiments provided a benchmark for the comparison of the two data mining algorithms and are still useful in conducting basic soil analysis.

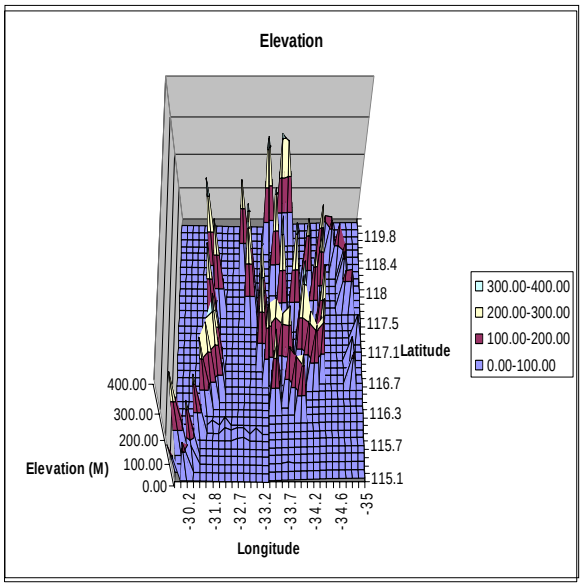
The three statistical methods that provided 3D surface maps came with a formatting problem. The graphs where also produced in reverse to the actual physical location due the reverse of longitude and latitude data. This resulted in anomalies in the production of 3D graphs which resulted in the need to use analysis techniques which required visual comparison.

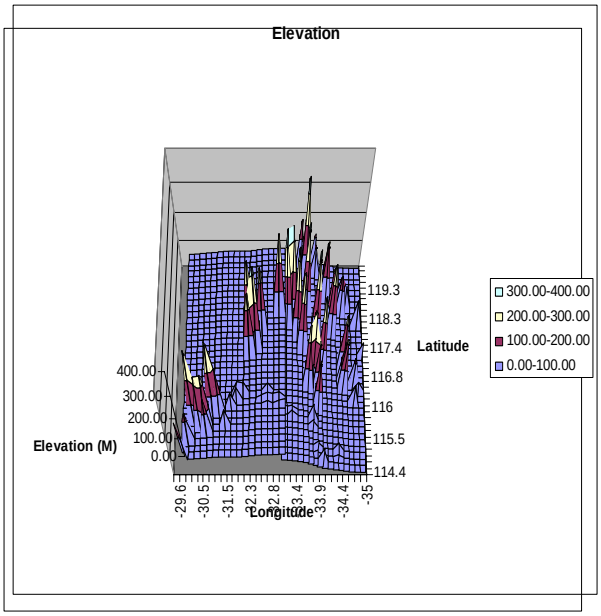
This problem was a major draw back when submitted to DAFWA researchers and they required extra time and effort to compare traits. The creation of a correlation table for full normalized data sets outlined the significant relations that exist between the traits. This method of analysis provided a quick over view of the data and allowed DAFWA researchers to conduct further research into the relationships.

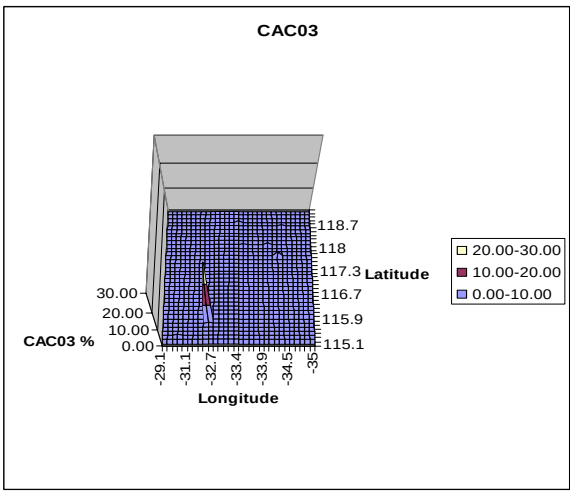
6.2 Evaluation of data mining

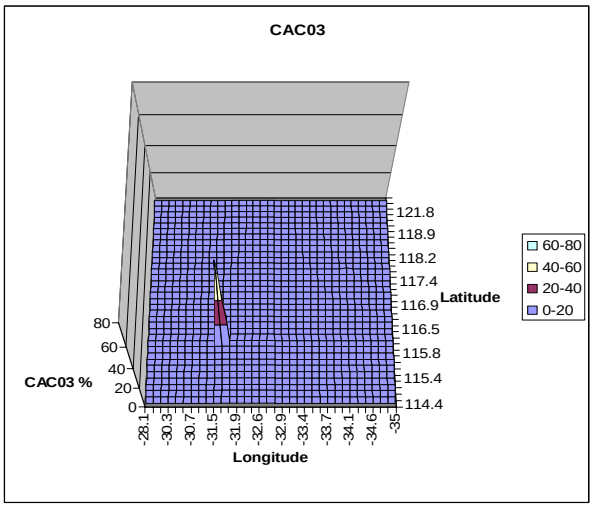
The application of data mining techniques has proven to be almost as accurate as standard statistical analysis techniques and with the increase in the number of instances this is projected to increase in accuracy. The WEKA data mining software provided a simple platform from which to undertake the research and comparison of the data set. The input of the data into data mining applications proved to be simple with the conversion of an Excel spread sheet into a CSV file and then an ARRF file.

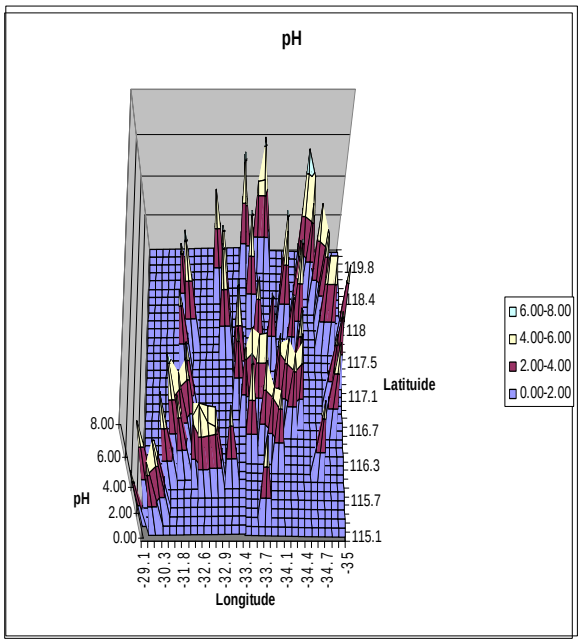
The two clustering algorithms used were also compared for ease of use and time taken to complete the analysis process for each of the five case studies. The EM algorithm required a larger amount of input to setup for each clustering operation and required that the number of clusters set. Processing time taken to complete each case study was also significant when compared with FarthestFirst. The analysis of the results showed that EM only correctly verified the dataset on a single instance. The two soils - two traits was the case study that EM was as accurate as FarthestFirst, both grouped the two clusters the same with cluster 0 at 536 instances and cluster 1 at 612 instances. The reason for this is unknown and both were one hundred per cent accurate on the actual data. The reason behind this requires further investigation.











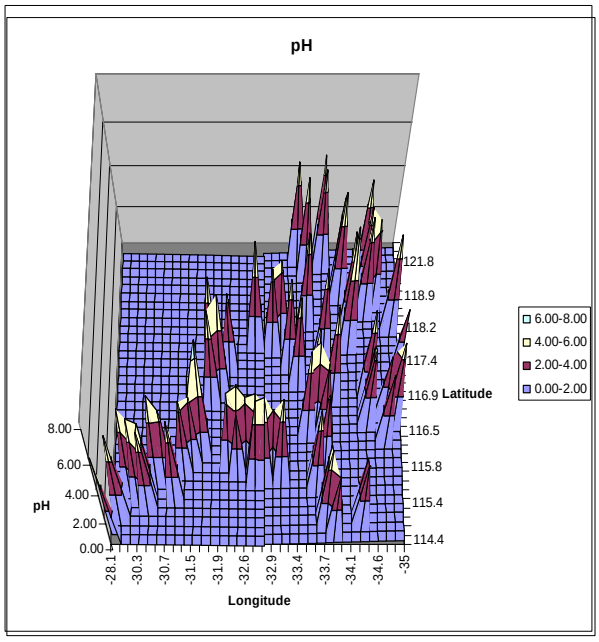


Figure 7 Elevation, CaCO3 and pH of Grey deep sandy duplex soil.

Figure 8 Elevation, CaCO3 and pH of pale deep sand.

Case study one was the major outlier with a high rate of incorrect classification for both instances, with EM splitting the instances into even groups of 268 for cluster 0 and 1. FarthestFirst grouped the instances more accurately than EM but still mis-classified 199 instances or 22 per cent of the dataset.

The FarthestFirst algorithm provided a much more accurate tool for the verification of valid patterns and profile clusters when tested against the benchmark, with most cluster groups within four instances of the actual data. The FarthestFirst algorithm proved to be much simpler to use and required less processing time to complete each case study. The FarthestFirst algorithm, when applied to data sets, can establish valid patterns and soil profile clusters. The FarthestFirst algorithm was the most efficient technique in determining patterns and clusters when compared to standard statistical analysis techniques.

The data mining application also provided a number of functions, such as visualize, where all traits were charted against each other and allowed for a quick analysis. This function was validated with the charting of the soil locations with the longitude and latitude data reflecting the initial data analysis graph created: See figures 9, 10, 11 for sample screen shots of WEKA analysis.

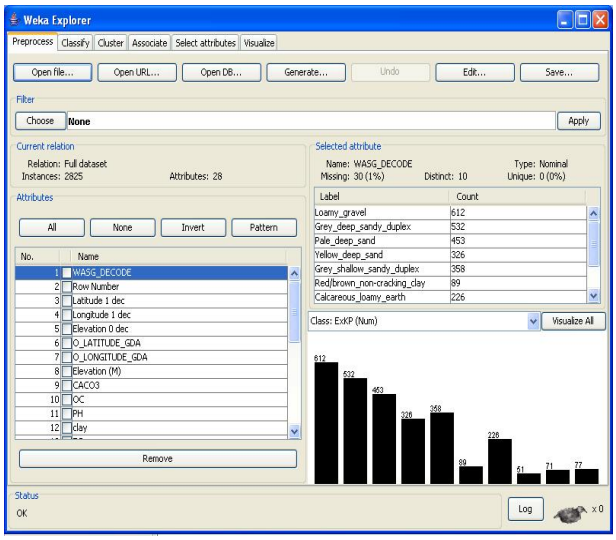


Figure 9 Screenshot of WEKA data mining tool.

6.3 Comparison between methods

The two methods of soil analysis had advantages and disadvantages, with both providing accurate clustering of the experimental dataset. The accuracy of the data mining clusters method on the agricultural soils was dependant on the selection of the correct algorithm, and was shown to have a wide grouping within the two algorithms researched. The two methods researched showed that data mining can equal the verification of valid patterns when compared to standard analysis techniques.

Analysis and classification of soil traits under the current system is very subjective with groups open to human interpretation. This human input means that clustering of similar soil types and traits can become less accurate and this can have an affect of the accuracy of analysis and knowledge gathering. The advantage found in the application of data mining techniques is that human interpretation is reduced and the data is clustered based on the actual information without bias.

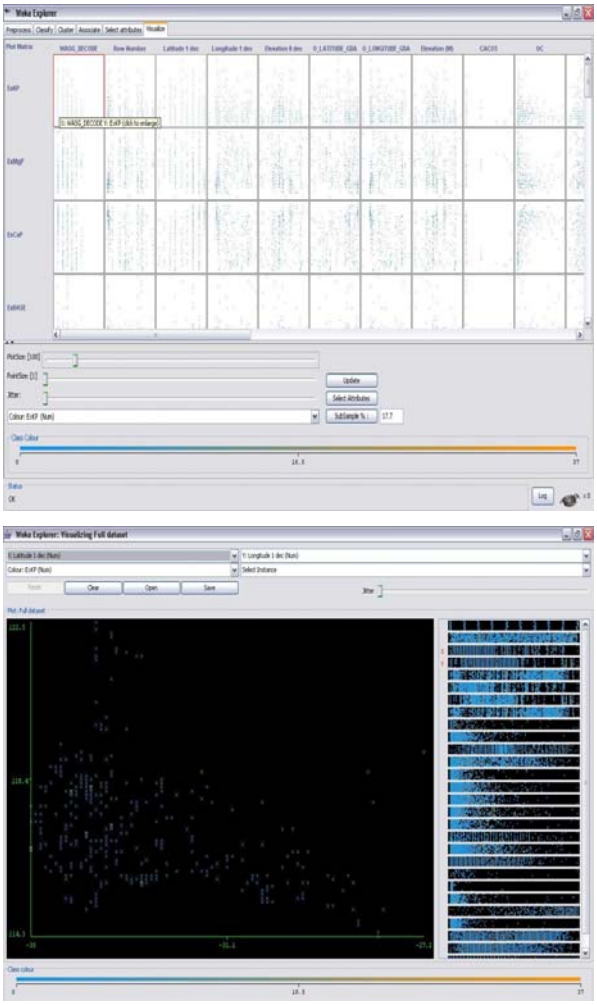


Figure 10 Screenshots showing initial analyses of all soils by WEKA.

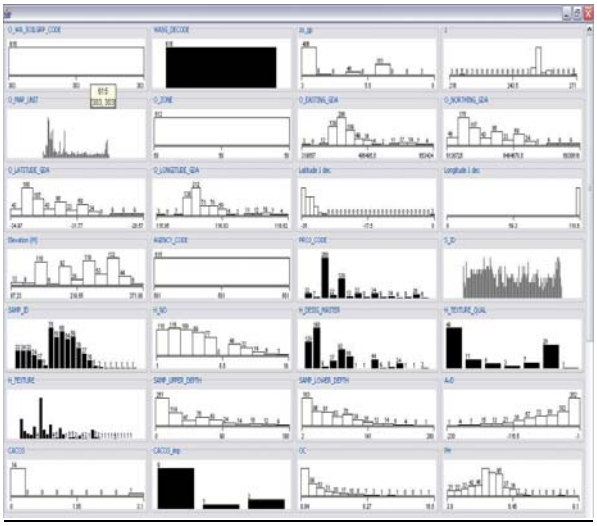


Figure 11 Initial analysis of Loamy gravel soil type by WEKA

Although data mining has a number of advantages over the current statistical methods, the WEKA software and process still has a number of problems. The research encountered a number of disadvantages that included selection of the current algorithm and the graph output being only in 2D with no provision for placing a third set of data on the visual display. The application of both methods still requires knowledge of the results required to allow selection of the correct techniques to provide new knowledge.

Comparison of the two methods has shown that data mining is still not one hundred per cent accurate on all applications, when compared with standard statistical methods, but has shown to have greater benefits. The benefits of data mining include speed and increased levels of automation, but still do not provide all the analytical tools required for analysis of an agricultural soil database.

6.4 Issues related to research

During the course of conducting this research project there where a number of problems that had to be overcome. These problems included the application of data mining techniques, quantity of data, tools including WEKA and Excel, skills required to undertake the research, limited time and interpretation of results.

The application of data mining techniques required a deep understanding of the process involved and required that a large amount of background research be undertaken before commencement of the research. The quantity of data is a problem when it is applied to data mining and statistical research with the data set size having a direct correlation with accuracy of outcomes.

The problem of a small dataset was overcome by creating small subsets of known values to remove data size as a variable in the research process and to focus on the methods applied to that data. The tools used in the research included WEKA and Excel and were very effective for conducting this research. The tools were very complex to use at a higher level and this problem was overcome with the help of experts in the field and by background research.

The interpretation of the research results was a complex process and did not focus on the relationships between the soils traits, but on the establishment of this relationship and their accuracy. The accuracy of each method and each algorithm was the corner stone of this research, with the methods used to determine these being critical to the research outcomes. The problem of establishing the accuracy of method was overcome by analysis of the grouping and numbers of instances in clusters. The comparison was then made to the actual number of instances in the clusters and the level of incorrect classification. The analysis of the relationships within the data traits, as it applies to soil science, is outside the scope of this research will be undertaken by DAFWA researchers.

Due to the limited amount of time available to conduct the research only limited experiments were conducted in the effectiveness of clustering algorithms. The limited amount of data did not allow use of the full range of tools available, within the WEKA software, to be tested. Future research could be conducted to build on this research and analyse more of the functions and algorithms available in the WEKA data mining software. All the problems encountered during the research were overcome with the aid of experts in given fields and with perseverance.

7 Conclusions

The experiments conducted analysed a small number of traits contained within the dataset to determine their effectiveness when compared with standard statistical techniques. The agriculture soil profiles that were used in this research were selected for completeness and for ease of application to data mining. The soil original dataset was almost compete but still contained some missing values that had to be removed in a text editor because of the affect on the clustering process.

Standard statistical analysis was used to establish a benchmark with pivot tables created using normal and standardized data. Pivot tables were then used to create a 3D surface map, charting traits against longitude and latitude. Statistical techniques take soil types that have been classified that are the same to produce means and estimates to determine the properties of that soil type. Data mining clustering methods do not use that same assumption, but, rather than using these predefined classifications, assigns instances based on their values to provide an objective method of classification.

The five case studies were designed to test the concept and methodology of data mining and to establish the accuracy of EM and FarthestFirst clustering algorithms. The two algorithms were selected because of these different methods of grouping the data into clusters. This process was done to allow another level of comparison in the research. The research outcome found that FarthestFirst algorithm grouped instances more accurately that the EM algorithm, when compared with the statistical benchmark. The results showed that FarthestFirst had a lower mis-classification rate, and classified case study two correctly, with limited error in case studies three and four.

The accuracy of data mining depends on the amount of data used to create clusters, with the literature indicating that an increase in dataset size improves accuracy. Further research would look at increased dataset size to determine if this would increase the instance classification. This would create more focus on data mining and less on current statistical methods. There were a number of areas not explored by the research due to time limitations, such as the differences between the soil profile horizons within the same excavation site being of particular interest to DAFWA researchers.

The recommendations arising from this research are: That data mining techniques may be applied in the field of soil research in the future as they will provide research tools for the comparison of large amounts of data. Data mining techniques, when applied to an agricultural soil profile, may improve the verification of valid patterns and profile clusters when compared to standard statistical analysis

techniques. The results of this research were passed on to the DAFWA researchers so they can determine if the application of data mining techniques may aid in their current and future soils research.

8 References

Cunningham, S. J., and Holmes, G. (1999). Developing innovative applications in agriculture using data mining. In the *Proceedings of the Southeast Asia Regional Computer Confederation Conference*,1999.

Griffin, M. T. (2005). *Data field descriptions.* Perth: Department of Agricultural and Food Western Australia.

Ibrahim, R. S. (1999). *Data Mining of Machine Learning Performance Data.* Unpublished Master of Applied Science (Information Technology), Publisher; RMIT University Press.

Isbell, R. F. (1996). *The Australian Soil Classification. Australian soil and land survey handbook.* (Vol. 4). Collingwood, Victoria, Australia: CSIRO Publishing.

Mckenzie, N., and Ryan, P. (1999). Spatial prediction of soil properties using environmental correlation. *Geoderma*, 89(1-2), 67-94.

Palace, B. (1996). Data Mining: What is Data Mining? Retrieved Aug 30, 2005, from http://www.anderson.ucla.edu/faculty/jason.frand/teacher/technologies/palace/datamining.htm

Ragel, A., and Cre´milleux, B. (1999). MVC—a preprocessing method to deal with missing values. *Knowledge-Based Systems*, 12, 285–291.

Schoknecht, N. (2002). *Soil Groups of Western Australia (Technical Report).* Perth: Department of Agriculture.

Schoknecht, N., Tille, P., and Purdie, B. (2004). *Soil-Landscape mapping in south-western Australia (Technical Report).* Perth: Department of Agricultural.

	CACO3	*OC*	*PH*	*clay*	*EC*	*ExCA*	*ExMG*	*ExK*	*ExNA*	*ExCEC*
CACO3	**1.00**									
OC	-0.05	**1.00**								
PH	0.44	-0.15	**1.00**							
clay	0.23	-0.19	0.42	**1.00**						
EC	0.32	-0.09	0.43	0.33	**1.00**					
ExCA	0.12	0.45	0.39	0.13	0.15	**1.00**				
ExMG	0.26	0.00	**0.60**	**0.61**	0.43	0.41	**1.00**			
ExK	0.30	0.03	**0.64**	0.33	0.40	0.45	0.46	**1.00**		
ExNA	0.27	-0.14	**0.58**	0.45	**0.57**	0.13	**0.75**	0.45	**1.00**	
ExCEC	0.19	0.09	**0.69**	**0.57**	0.28	**0.62**	**0.84**	**0.60**	**0.65**	**1.00**
ExSUM	0.30	0.19	**0.67**	**0.48**	0.46	**0.74**	**0.88**	**0.63**	**0.71**	**0.93**
ExESP	0.22	-0.27	0.28	0.32	0.45	-0.18	0.39	0.22	**0.67**	0.32
ExH	**-0.77**	**0.82**	-0.12	-0.20	-0.08	0.42	0.14	0.11	-0.11	0.00
ExMN	-0.04	0.11	0.11	0.02	0.33	0.12	0.05	0.12	0.02	-0.28
ExAL	-0.14	0.14	**-0.49**	0.08	0.09	-0.01	0.28	0.04	0.19	-0.14
ExSAT_PC	0.25	-0.15	**0.49**	0.42	0.30	0.30	**0.47**	0.33	0.35	0.37
ExBASE	-0.04	0.35	-0.02	-0.24	0.02	0.30	0.06	0.05	0.01	0.03
ExCaP	-0.22	0.34	-0.24	**-0.55**	-0.32	0.28	**-0.48**	-0.16	**-0.47**	-0.20
ExMgP	0.04	-0.24	0.08	**0.52**	0.08	-0.23	0.38	-0.06	0.15	-0.03
ExKP	0.20	-0.12	0.13	-0.05	0.08	-0.05	-0.12	**0.46**	0.00	0.04

	ExSUM	*ExESP*	*ExH*	*ExMN*	*ExAL*	*ExSAT_PC*	*ExBASE*	*ExCaP*	*ExMgP*	*ExKP*
CACO3										
OC										
PH										
clay										
EC										
ExCA										
ExMG										
ExK										
ExNA										
ExCEC										
ExSUM	**1.00**									
ExESP	0.29	**1.00**								
ExH	0.30	-0.23	**1.00**							
ExMN	0.10	0.00	0.00	**1.00**						
ExAL	0.19	0.12	0.00	-0.01	**1.00**					
ExSAT_PC	**0.49**	0.25	0.00	0.45	**-0.47**	**1.00**				
ExBASE	0.18	-0.10	**0.76**	0.03	-0.03	0.13	**1.00**			
ExCaP	-0.21	**-0.69**	0.27	-0.01	-0.11	-0.17	0.20	**1.00**		
ExMgP	0.08	0.19	-0.17	0.01	0.06	0.01	-0.16	**-0.82**	**1.00**	
ExKP	-0.03	0.07	-0.20	0.02	0.02	-0.02	-0.11	-0.05	-0.23	**1.00**

Table 4 Correlation Table of Soil Data Traits

Useful Clustering Outcomes from Meaningful Time Series Clustering

Jason R. Chen

Department of Information Engineering
Research School of Information Science and Engineering
College of Engineering and Computer Science
The Australian National University
Canberra, ACT, 0200, Australia
Email: Jason.Chen@anu.edu.au

Abstract

Clustering time series data using the popular subsequence (STS) technique has been widely used in the data mining and wider communities. Recently the conclusion was made that it is meaningless, based on the findings that it produces (a) clustering outcomes for distinct time series that are not distinguishable from one another, and (b) cluster centroids that are smoothed. More recent work has since showed that (a) could be solved by introducing a lag in the subsequence vector construction process, however we show in this paper that such an approach does not solve (b). Motivating the terminology that a clustering method which overcomes (a) is meaningful, while one which overcomes (a) and (b) is useful, we propose an approach that produces *useful* time series clustering. The approach is based on restricting the clustering space to extend only over the region visited by the time series in the subsequence vector space. We test the approach on a set of 12 diverse real-world and synthetic data sets and find that (a) one can distinguish between the clusterings of these time series, and (b) that the centroids produced in each case retain the character of the underlying series from which they came.

Keywords: Time Series, Clustering, Subsequence-Time-Series Clustering

1 Introduction

Clustering analysis is a tool used widely in the Data Mining community and beyond (B.S.Everitt et al. 2001). In essence, the method allows us to summarise what can be a very large data set X with a much smaller set $C = \{c_i | i = 1, \ldots, k\}$ of representative points (called centroids), and a membership map $\gamma : X \rightarrow C$ relating each point in X to its representative in C. It then becomes much easier to access and manipulate the essential "information" in the data set, and this is one of the reasons why clustering is so widely used, and often as a pre-processing step for further analysis.

Time series are a special type of data set where elements have a temporal ordering. Imagine we have a system or process evolving over time, and that we take measurements x_t (scalar or vector) on a fixed periodic basis. Then our time series X is

$$X = \{x_t | t = 1, \ldots, n\} \quad (1)$$

where n is the number of measurements taken, and where the subscript t reflects the temporal ordering in the set.

Given a single time series, or a number of time series from the same system or process, we often wish to summarise the series as a set of key features or shapes found in the series. Clustering analysis seems an obvious candidate for such a purpose and, historically, the approach has been to construct a set Z of delay vectors (called "subsequences" in (E.Keogh et al. 2003), "regressors" in (G.Simon et al. 2006)) by moving a sliding window of length w across the data, where the pth delay vector is given by,

$$z_p = \{x_{p-(w-1)}, x_{p-(w-2)}, \ldots, x_{p-2}, x_{p-1}, x_p\} \quad (2)$$

and where $Z = \{z_p | p = w \ldots n\}$. Clustering Z in $\mathbb{R}^w$ using one of a number of possible metrics (e.g. including those induced by the L^p norms, but usually the L^2 (Euclidean) norm, Mahalanobis distance, DTW distance, etc.) was typically conducted using one of the many possible clustering algorithms (k-means, hierarchical, EM-algorithm, Self Organising Maps, etc.) to produce a set C of representatives that was used as the set of key features in the time series. In this paper we refer to this approach as Subsequence-Time-Series (STS) clustering.

STS Clustering has been widely used, as highlighted in (E.Keogh et al. 2003). However, surprisingly, work in (E.Keogh et al. 2003) found that the outcome of clustering a time series in this way is meaningless. The two principal findings made to support this conclusion were,

(A) that one cannot distinguish between the clustering outcomes of distinct time series, and

(B) that cluster representatives are smoothed and generally do not look at all like any part of the original time series

Clearly, both (A) and (B) are problematic properties for any prospective "summary" of a time series to have. While (E.Keogh et al. 2003) proposed that these two problems were one and the same, i.e. that one cannot distinguish between cluster centres of different time series because they are all smoothed and hence alike, we show in this paper that this is in fact false. Problems (A) and (B) are really two separate problems that can be solved separately, and it will turn out that our contribution in this paper will be to propose a method which solves problem (B).

A number of reasons and solutions to the dilemma raised in (E.Keogh et al. 2003) have been proposed in the literature. Struzik (Z.R.Struzik 2003) proposed that the "meaningless" outcome results only in pathological cases, i.e. when the time series structure is fractal, or when the redundancy of subsequence sampling causes trivial matches to hide the underlying rules in the series. They suggested autocorrelation operations to suppress the latter, but did not confirm this with experiments. Denton (A.Denton 2005) proposed density based clustering, as opposed to k-means or hierarchical clustering, as a solution. They proposed that time series can contain significant noise, and that density based clustering identifies and removes this noise by only considering clusters rising above a preset threshold in the density landscape. Chen (J.R.Chen 2007) proposed an alternative clustering metric based on temporal and formal distances that did lead to meaningful time series clustering,

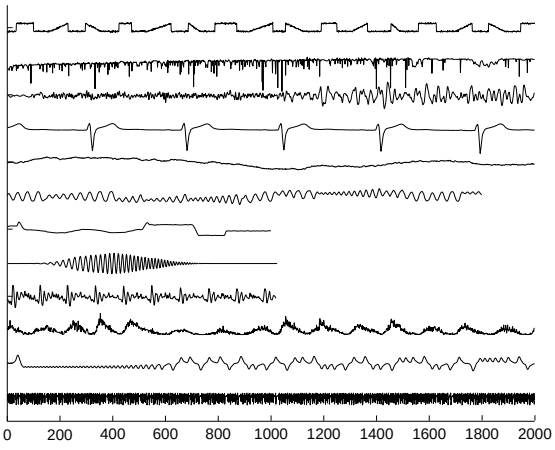


Figure 1: The time series used in experiments: CBFV time series, the Earthquake, Koski-ECG, Random Walk, Chaotic, Shuttle, Speech time series from the UCR database (E.Keogh 2002), and the Sunspot, Lorenz and Henon time series from (G.Simon et al. 2006)

however the work was limited to time series that are cyclic. Peker (K.Peker 2005) identified that clustering with a very large number of clusters leads to cluster centroids which are more representative of the signal in the original time series. Goldin et. al. (D.Goldin et al. 2006) proposed an alternative way of measuring distance between different time series clusterings, which gave quite high recognition rates with respect to the time series from which they came. Although each of these works has shown promising results, none has clearly demonstrated a method which overcomes the problems (A) and (B) identified in (E.Keogh et al. 2003).

Some work in the literature proposed that a lag should be introduced into the window formation aspect of STS-clustering (J.R.Chen 2007, G.Simon et al. 2006), i.e.

$$z_p = \{x_{p-(w-1)q}, x_{p-(w-2)q}, \ldots, x_{p-2q}, x_{p-q}, x_p\} \quad (3)$$

and $Z = \{z_p \,|\, p = (w-1)q+1, \ldots, n\}$. That is, delay vectors should not be formed (as in Equation 2) from w contiguous data points in the time series, but rather from w data points each separated themselves by q data points. In essence, introducing the lag q unfolds the delay vector distribution (spreads delay vectors away from the diagonal of delay space) so that the "information" existing in the time series is retained. Work in (G.Simon et al. 2006) went on to show that the clustering outcomes of different time series could indeed be distinguished from one another if this unfolding is carried out first. We will refer to this approach as Unfolded-Time-Series (UTS) Clustering.

While we feel that the work in (G.Simon et al. 2006) provides an important step towards the solution of the time series clustering dilemma, i.e. it solves problem (A), we propose that it is not the final one. More specifically, work in (J.R.Chen 2007) found that, even when a lag was introduced in the the delay vector construction process, smoothed cluster centroids were still produced. i.e. problem (B) still existed. Specifically, quite a number of lags were tried in the experiments involving the Cylinder-Bell-Funnel (CBF) time series in that work, however none gave the desired outcome. Hence, it seems there is some discrepancy about whether introducing a lag results in meaningful clustering, or whether something else is required. The remainder of this paper is about making sense of what is happening here.

Time Series	Lag (q)	Time Series	Lag (q)
CBFV	9	Shuttle	50
Balloon	7	Flutter	5
Earthquake	7	Speech	7
Koski ECG	18	Sunspot	20
Random Walk	100	Lorenz	14
Chaotic	7	Henon	1

Table 1: Lags selected for $w = 5$

2 Meaningful versus Useful

Let us first provide a grounding to our work here by confirming the results in (G.Simon et al. 2006). We select 12 time series (as shown in Figure 1): (1) the CBF time series used in (J.R.Chen 2007) with variability added in feature onset and duration [1] , (2-9) from (E.Keogh 2002) identified in (E.Keogh et al. 2003) as having some of the most distinct characteristics of any time series in that database, and (10-12) used in (G.Simon et al. 2006). We adopt the same basic methodology as in (G.Simon et al. 2006) (and (E.Keogh et al. 2003)) of creating a set of centroid sets for each time series and measuring the difference between centroid locations for the same, and then different, time series. The steps taken were (details, where not present, can be found in (G.Simon et al. 2006)):

1. Some time series from (E.Keogh 2002) are very long. For these time series we take only the first 2000 points. This is sufficient to make our point in this paper
2. Normalise each time series to have zero mean and standard deviation one. This is mandatory if a fair comparison of distances between centroid sets of different time series pairs is to be made.
3. Choose a lag according to a maximum or plateau in the Sum-of-Distance-To-Diagonal ($SDTD$) measure, as proposed in (G.Simon et al. 2006). We in fact use the Mean-Distance-To-Diagonal ($MDTD$), calculated by dividing $SDTD$ by the number of points in Z (a constant), since this measure will be useful later. That is,

[1] In contrast to work in (J.R.Chen 2007), where a fi xed feature onset and duration CBF time series was used, in this paper, we construct the CBF time series with features with variable onset and duration. Specifi cally, we take"windows"of data points of length 128, where each window contains one of either the Cylinder, Bell, or Funnel features, and where the onset and termination of the feature is selected with a uniform probability over the data point ranges of 16 to 48 and 80 to 112 respectively. This results in, by visual inspection, at least an equal, if not greater, variability in feature onset and duration than the original CBF time series used in (E.Keogh et al. 2003). We show the resulting time series in Figure 2. For clarity, denote the "fi xed" CBF time series used in (J.R.Chen 2007) as the CBFF time series, and the "variable" CBF time series we use here as the CBFV time series.

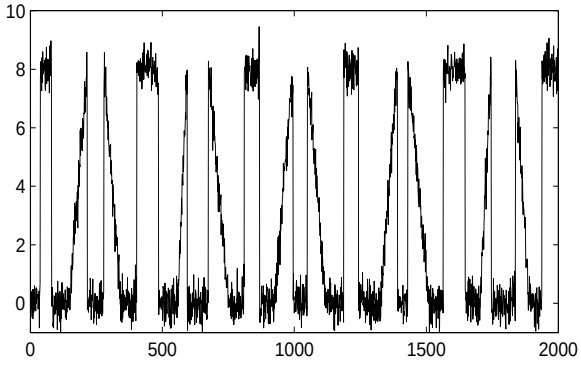


Figure 2: CBFV Time Series

$$MDTD = \frac{1}{n'} \sum_{p=1}^{n'} \sqrt{\frac{\|v_1 - z_p\|^2 \|u\|^2 - ((v_1 - z_p)^T u)^2}{\|u\|^2}} \quad (4)$$

where $n' = n - (w-1)q$ reflects the number of delay vectors in Z, $\|.\|$ is the L_2 norm, and $u = v_2 - v_1$ where v_1 and v_2 are the w-dimensional origin $(0, 0, 0, 0, \ldots)$ and unit vectors $(1, 1, 1, 1, \ldots)$ respectively. The idea here is to find a lag that provides a sufficiently unfolded delay vector distribution, and maxima in the $MDTD$ value reflect when this occurs. The lags selected for each time series, for w = 5 are shown in Table 1.

4. Cluster the normalised time series using UTS-clustering with the lag selected from Step 3. Use k-means clustering and the Euclidean metric. Repeat the clustering two lots of 100 times to create two sets of centroid sets for each time series: a base set $\mathcal{B}$ and a comparison set $\mathcal{C}$.

5. For each base centroid set $\mathcal{B}_r$ (i.e. $r = 1, \ldots, 12$ for our set of 12 time series here) calculate the distance between it and every comparison set $\mathcal{C}_s$, $s = 1, \ldots, 12$. When $r = s$, the distance is called the *within* distance (i.e reflecting that both sets of clusterings came from the same time series), while for $r \neq s$, the distance is called the *between* distance.

6. For each time series as the base, plot the within distance value, and the between distances values, for a range of number-of-cluster (k) values (from 3 to 70).

The rationale behind these steps is that, if the clustering of one time series is distinguishable from that of another, then the within distance should always be less than the between distance. In other words, the position of centroids in clusterings produced from the same time series should coincide more than those of different time series. As in (G.Simon et al. 2006), *sets* of centroid sets were used to allow for the effect of different clustering outcomes from the one time series due to differing initial seeds.

The result of the experiments is shown in Figure 3 and we see that the clustering outcome produced by different time series are indeed distinguishable if a lag is introduced, as reported by (G.Simon et al. 2006). For each time series as a base, the distance between centroids produced by clustering the same time series is always smaller than that produced by clusterings from different time series. The results shown here are for $w = 5$, however as in (G.Simon et al. 2006), this experiment was also repeated with a number of higher and lower w values, and were found to support the same conclusions in each case. Hence introducing a lag into the sliding windows part of the STS-clustering process does indeed seem to solve problem (A).

We now then return to our initial observation that introducing a lag in the CBFF experiment in (J.R.Chen 2007) did not prevent smoothed cluster centroids from being produced, i.e. that it did not solve problem (B). A number of lag values were selected in those experiments; however it may have been that an appropriate one (i.e. one obtained using the $MDTD$ criteria) was not selected. Let us repeat the experiment conducted in (J.R.Chen 2007), but this time use the $MDTD$ criteria to select the lag. UTS-cluster the CBFV time series with $w = 10$, $k = 3$ and a lag $q = 13$ selected using the $MDTD$ criteria. Figure 4 shows the result is still the same set of smoothed cluster centroids first reported in (E.Keogh et al. 2003). In the lower plot in Figure 4, we indicate the membership of the different delay vectors constructed from the time series. Each delay vector in Z has a distinct element x_t in the time series as its first component. For each delay vector, we plot in Figure 4 a coloured dot below the x_t which forms its first component,

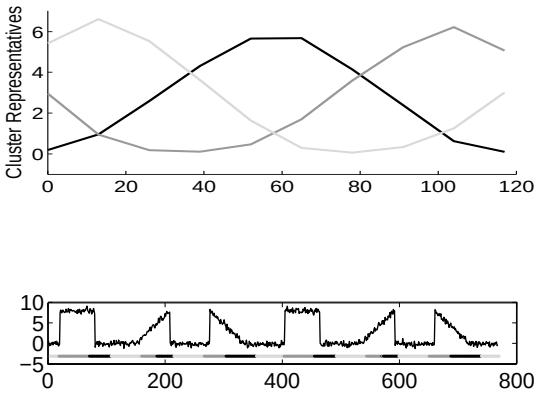


Figure 4: UTS Clustering: CBFV results

where the colour of the dot indicates the delay vector's cluster membership. In essence we have formed a membership "bar" for delay vectors that can be superimposed back on the original time series. Note that the membership outcome in Figure 4 is not what we are looking for, since it has not captured the three features into separate clusters.

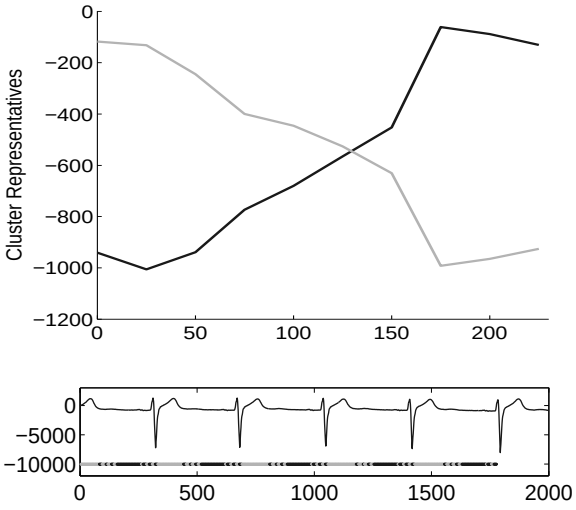


Figure 5: UTS Clustering: Koski-ECG results

Let us try UTS-clustering other time series in our data set to see if the CBFV time series represents an isolated problematic result. First, try to identify the two main features in the Koski-ECG time series: the in-beat, and the between-beat phases of the heart in this series. Figure 5 shows the centroids produced using $w = 10$, $k = 2$ and a lag $q = 25$ selected using the $MDTD$ criteria. The resulting centroids look nothing like the features in the original time series. Further, although the membership bar broadly demarcates between the in-beat and between-beat phases, a curious oscillation between memberships occurs at transitions.

Next, let us try UTS-clustering the Chaotic time series. We choose to look for 3 clusters, with $w = 10$ and a lag $q = 10$ selected using the $MDTD$ criteria. Figure 6 shows the strange result. The membership bar seems to indicate that the UTS clustering method has identified oscillations in this time series occurring at three different vertical axis levels (i.e. roughly at the values -0.25, 0 and 0.25), however the oscillations in the time series at these three levels have mysteriously been smoothed. Note that there is no part of

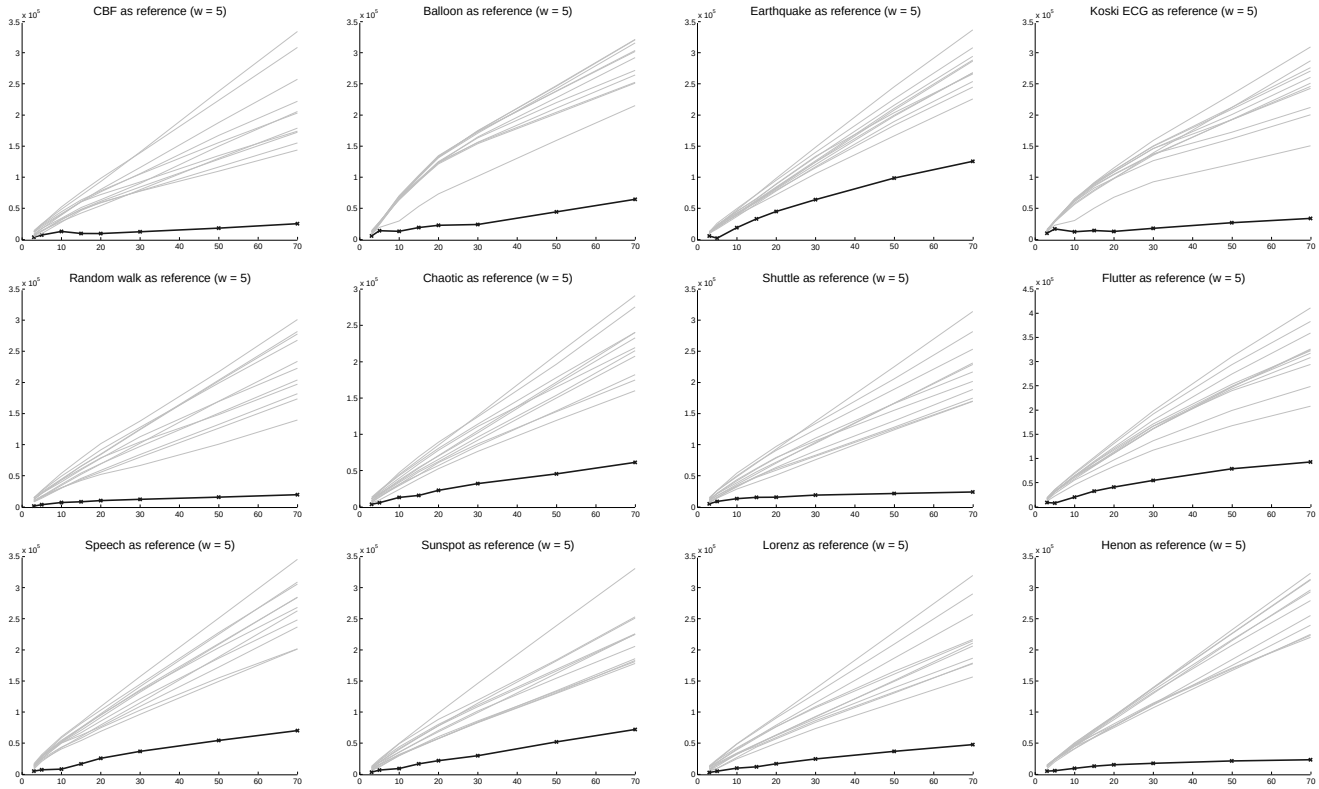


Figure 3: Results of the within-between distance experiments using UTS Clustering

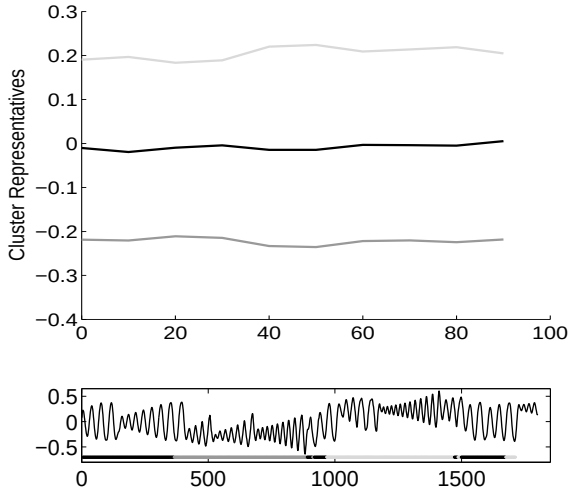


Figure 6: UTS Clustering: Chaotic results

the original time series at this time scale where the flat segments shown occurred.

It would seem, then, that introducing a lag into the sliding windows part of the STS-clustering process does not solve problem (B). However, let us conduct another experiment involving all the time series in our data set that further confirms what we have just observed. Work in (J.R.Chen 2007) proposed that the smoothing effect associated with problem (B) occurs because the STS and UTS clustering methods produce centroids that lie away from the cluster members they are meant to represent. The work there made the observations based on experiments with a small number of cyclic time series. Let us confirm this observation using the full range of general time series data sets introduced above. We propose a simple measure we term the CP-PP ratio to measure the existence of this effect. Given a set $C = \{c_i | i = 1, \ldots, k\}$ of centroids from a time series clustering, first, measure the average minimum centroid to data point distance as

$$MCD = \frac{1}{k} \sum_{i=1}^{k} \min_j d(c_i, z_{ji}), 1 \leq j \leq n_i \qquad (5)$$

where z_{ji} indicates the jth data point in the ith cluster, and where $d(.,.)$ indicates Euclidean Distance. That is, MCD gives the average distance between the centroids in C and the nearest members in their respective clusters. Next calculate, for every point in Z, the average distance between this data point and the closest data point in the same cluster as

$$MDD = \frac{1}{k} \sum_{i=1}^{k} \frac{1}{n_i} \sum_{p=1}^{n_i} \min_j d(z_{pi}, z_{ji}), 1 \leq j \leq n_i \qquad (6)$$

Finally, take the CP-PP ratio simply as

$$CP\text{–}PP = \frac{MCD}{MDD} \qquad (7)$$

One would envisage CP-PP's generally at, or less than, unity, reflecting a deflationary influence on MCD of clusters centres lying in among the data points in the cluster, and the inflationary influence on MDD of outliers and points on the boundaries of the cluster. If we UTS-cluster each of the 12 time series introduced above and calculate CP-PP for a range of k values, we obtain the results shown in Figure 7 (we split these results across two plots for clarity). Note the large values of CP-PP for many time series, especially the Henon, Shuttle, Koski-ECG, Random Walk, Lorenz and Chaotic time series. These larger values of CP-PP reduce at higher k values, and this concurs with the findings of Peker (K.Peker 2005) that clustering with large k numbers produces centroids with characteristics more like those of

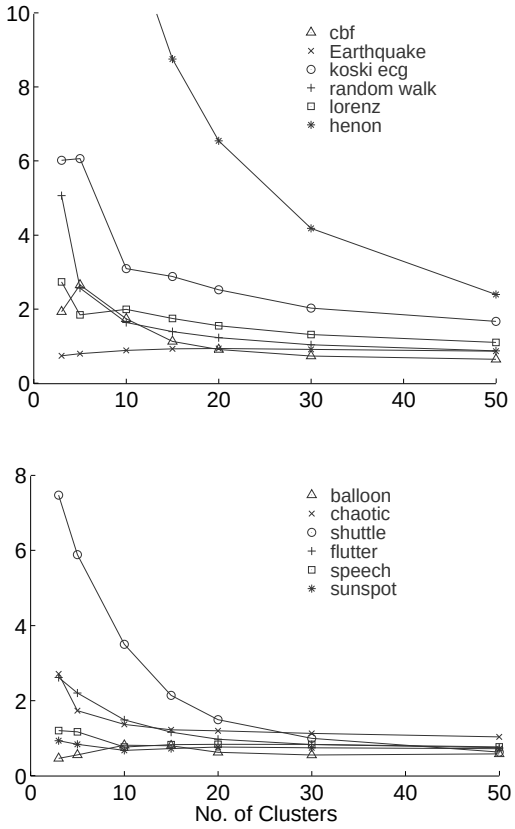


Figure 7: Ratio of centroid-to-nearest-point distance to point-to-nearest-point distance (CP-PP Ratio) for a range of k values

the original time series. Note that the results in Figure 7 are average values calculated over 100 clustering runs for each time series using random initial seeds, and so are not the result of any particular clustering initialisation used.

Let us now summarise the resulting situation. Work in (E.Keogh et al. 2003) concluded that STS clustering was not meaningful for reasons (A) and (B) above. Work in (G.Simon et al. 2006) proposed that clustering is meaningful if a lag is introduced. Our results here would suggest that clustering time series does not necessarily become meaningful (in the sense that it was used in (E.Keogh et al. 2003), i.e. (A) and (B) above) by only introducing a lag, since, although (A) is overcome, (B) is not. Hence we now introduce a more precise set of terminology to reflect this fact.

Definition 1 *The outcome of a time series clustering algorithm is meaningful if the cluster centres produced for one time series are distinguishable from those produced for other distinct time series.*

One can determine if cluster centres are distinguishable based on the experimental method outlined in Steps 1 to 6 above. We propose that *meaningful* is the correct term to use here. If the cluster centroids produced from a particular time series are distinguishable (i.e. distinct) from those of many other time series, then the clustering is meaningful since it has preserved the "information" in the original time series that made the time series distinct from all others.

Definition 2 *The outcome of a time series clustering algorithm is useful if it is (i) meaningful and (ii) if centroids well represent members in their respective clusters (in the sense that they retain the properties - e.g. sharpness, magnitudes, etc - of the signal in the original time series).*

We adopt the term *useful* here since our stated motivation at the outset of this paper was for clustering to obtain a "summarising" feature set for time series data. A clustering is

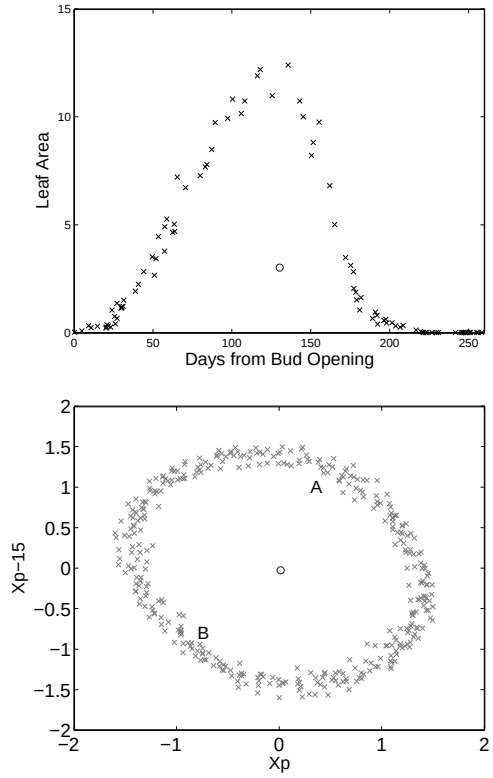


Figure 8: Hypothetical tree leaf area (top) and Undamped Pendulum delay space plots (bottom)

only useful in this context if centroids well represent their cluster members in the sense that they retain the properties of the signal in the original time series. Our aim for the remainder of this paper is to propose a method that produces useful time series clustering.

3 Asking the Right Question

So, according to the definitions in the previous section, introducing an appropriate lag leads to meaningful time series clustering, but not to useful clustering. How can the clustering process produce cluster representatives that look nothing like the cluster members they represent? We propose two possibilities; either clustering as a method is flawed, or that the clustering technique is sound, but that we are applying it in the wrong way, (i.e. we are using it in a way that is asking the wrong question). The clustering technique has a long history of successful use outside the time series domain, so it is improbable the fault lies there. We then explore if we could be applying the technique in the wrong way.

The general technique of clustering involves extracting some common pertinent features (e.g. height and weight) from a set of objects (e.g. children) and creating a clustering space formed as a coordinate system where each axis represents one of the aforementioned features. In constructing such a space, an important question to ask is what subset of this space corresponds to realisable instances or outcomes in these objects in the real world? It seems clear that we should limit any clustering process to this subset, or nonsensical clustering outcomes can result.

For example, take for illustration purposes a hypothetical data set of the total leaf area for a particular species and maturity of healthy deciduous tree during the growing season. Figure 8 (top) shows the data in a clustering space formed by capturing the two pertinent parameters from the physical system: the total leaf area in metres squared and the number of days from when the leaf emerges from the

bud. Data points are shown as crosses, and if we cluster for a single cluster, we obtain a non-sensical outcome: a healthy tree in mid-summer without (or with few) leaves. However, it is easy to see why this point was chosen as the centroid; it really is the point in the clustering space which has minimum sum of distance to all the data points. Clustering for a single cluster is an unusual, but valid, step which best highlights the point we wish to make. Clustering with more than one cluster here leads to the same effect, albeit in a less pronounced way.

The previous example was illustrative, but involved a non-time series derived data set. Lets look more closely at this issue with regard to clustering a set of time-series-derived delay vectors. In Figure 8 (bottom) we show a plot of a set of delay vectors of a simple undamped pendulum time series with measurement noise [2] (ignore the labels A and B for the moment). We know that this pendulum system only lives in (i.e. it can only produce delay vectors that lie in) the annular region shown, and so we should limit any clustering process here to this subset of delay space. If we take the usual approach of clustering without restriction (e.g. k-means and the Euclidean Metric) for one centroid, we get the problematic result shown. This result is problematic since the outcome represetented by this point (zero swing amplitude with zero velocity) does not occur in this system. Again, it is easy to see why this point was chosen as the centroid; it really is the point in the clustering space which has minimum sum of distance to all the data points. As with the Leaf-Area example, clustering with more than one cluster here leads to the same effect, albeit in a less pronounced way. What sense does it make to cluster in a space which includes points that cannot be produced by the underlying system? i.e. that can never appear in the time series. Note that the idea that clustering should be limited to a subset of the clustering space pertinent to the underlying physical system is not new. Work proposing methods for clustering on subspaces ((R.M.Haralick & R.Harpaz 2005) and references therein), and on manifolds (M.Breitenbach & G.Z.Grundic 2005) exists and is motivated by exactly this line of thinking, although we should not expect arbitrary time series data to generate delay vectors which exibit a subspace or manifold structure.

So, we have proposed that one should cluster in a restricted region corresponding to plausible outcomes realisable in nature. The problem is that we have available a set of time-series-derived delay vectors that generally represent only a partial view of where in delay space the underlying system lives. We then need to make an assumption as to where our system lives in delay space. The two obvious possible approaches are: (i) cluster in delay space without restriction (the approach to date), or (ii) cluster in that part of delay space where we have evidence (from the time series) that the system lives. We propose that each is a valid assumption to make, but that each asks a different question. That is, if we want to assume that the system can produce the full range of signals corresponding to an unrestricted set of delay vectors in delay space, then the answer we get necessarily can include delay vectors as centroids that were not in (or not like any in) the original time series. As we saw in the Leaf-Area and Undamped-Pendulum examples above, these centroids really do represent the best average member of members in their cluster if we accept that the physical system can produce any delay vector in delay space. The clustering technique is providing an appropriate answer to the question we are asking here (i.e. we cannot complain about centroids being smoothed).

The other obvious approach we can take is to make assumption (ii), which corresponds to asking a different question. When we look to cluster a time series for k clusters, the question we generally want to ask is: given the features (delay vectors) that were *seen* in the time series, which k of these features "summarise" (in the sense that we described how the clustering technique summarises a data set in the introduction to this paper) the time series best?, i.e. we are looking for (given Definition 2 above) a useful clustering. Assumption (ii) then makes sense since we are forming a clustering space containing only these features, and the clustering technique will return what we desire; a centroid set as the set of features *existing* in the time series that best summarise the time series. With this as motivation, we now present the details of the approach.

[2] This time series consists of essentially a noisy sinusoid. The plot shown is then formed by applying Equation 3 with $w = 2$ and a lag q which unfolds the data into the annular region shown

4 An Algorithm

Algorithm 1 TF Clustering Algorithm

1: Find the ED distance between all point pairs in Z, and store in A^*, i.e. $A^*_{ij} = d(z_i, z_j)$, where $d(.,.)$ denotes the Euclidean Distance metric
2: Set main diagonal of A to zero
3: Set first upper and lower diagonal of A equal to those in A^*
4: For each remaining entry in A, set A_{ij} to A^*_{ij} if $A^*_{ij} < \epsilon$ or zero otherwise
5: Use Floyd Warshall Algorithm or alternative to calculate a shortest path distance matrix D from A
6: Cluster using the K-mediods algorithm on the space $(Z, d_{TF}(.,.))$ where $d_{TF}(z_i, z_j) = D_{ij}$

There are three main phases to our approach. The first phase (Steps 1 to 4 in Algorithm 1) is to define a clustering "space" that is restricted to the region of delay space that was visited by the time series. Our approach is to capture the geometric structure of the delay vectors in delay space in a graph $\mathcal{G}$, where there is a node in $\mathcal{G}$ for each delay vector in Z, and where arcs represent the Euclidean distance between delay vectors. The key to creating a space that is restricted to the region visited by the time series using this approach is to apply an upper bound ϵ on the distance that can exist between vectors, above which the nodes in $\mathcal{G}$ representing them are not connected by a direct arc. For example, delay vectors in localities A and B in the Pendulum delay space plot in Figure 8 will not be directly connected by an arc in $\mathcal{G}$ if a reasonable ϵ is set. Rather, the distance between these vectors will be built up as the sum of distances along a path (in the annular region) of locally adjacent (i.e. within ϵ spaced) vectors between A and B.

It is interesting to note that the construction of our restricted region is quite similar to the first 2 steps of the seminal method by Taunenbaum et. al. (J.B.Tenenbaum et al. 2000) for identifying embedded manifolds from data. However, our situation here is slightly different to what they address since (i) for general time series we do not necessarily expect to find a manifold structure, (ii) we are not interested in applying the dimensionality reduction steps (i.e. steps subsequent to Steps 1 and 2) in that method prior to clustering since they impede the selection of cluster centroids, and (iii) there is the notion of temporal adjacency between delay vectors in Z which does not exist in general non-time-series derived data sets, and this is critical in affecting how arcs are input into the graph. Note that, by temporally adjacent vectors, we mean two vectors $z_i, z_j \in Z$ where $i = j + 1$ or $i = j - 1$.

The cost of arcs in $\mathcal{G}$ are stored in a matrix A. The last point (iii) above hints at why we split the construction of A into two parts, i.e. in Step 3 we construct the first upper and lower diagonals (entries representing temporally adjacent vectors), and then in step 4 we do the remaining entries. A time series is usually a discrete sampling of a continuously evolving system, resulting in a discrete trajectory of delay vectors in delay space. Then, we know that temporally adjacent vectors really should be connected (since they are separated by a distance that simply reflects our sampling rate) and so we always do this (Step 3). For the remainder

of delay vector pairs (Step 4) we connect them with an arc only if the distance is below a threshold ϵ.

The second phase of our approach concerns building the metric space $(Z, d_{TF}(.,.))$ in which to cluster. We construct $d_{TF}(.,.)$ using A. This involves building a distance matrix D, where $D_{ij} = d_{TF}(z_i, z_j)$ holds the shortest path distance in $\mathcal{G}$ between nodes i and j. There are a number of ways to construct D from A. One is the well known Floyd Warshall Algorithm. A reduced computation time approach uses Dijkstra's algorithm with Fibonacci heap data structures (J.B.Tenenbaum et al. 2000). We used the latter for implementations in this paper. We call d_{TF} the TF-metric [3]. Phase 3 of the approach simply involves clustering in the space (Z, d_{TF}) using the k-mediods algorithm. Denote the above algorithm as the TF-clustering algorithm or the TF-algorithm.

As in (J.B.Tenenbaum et al. 2000) the setting of the threshold ϵ is an integral part of our approach. When we cluster a new time series, we need a way to set ϵ. An appropriate value will ensure we don't capture in d_{TF} the straight line distance between points on diverse and unrelated parts of the delay vector distribution in delay space. Roughly speaking, we don't want to fill in "holes" or voids in this distribution when constructing our restricted region. Clearly then the value of ϵ should not be anywhere near the extent of the distribution (this would just give UTS-clustering) but rather some fraction of it. Note that the $MDTD$ value provides a measure of the extent of the distribution of delay vectors in delay space. For the purposes of our experiments in this paper, we used ϵ set at 10 percent of the $MDTD$ value. Intuitively, this represented a reasonable value given our motivation above of not filling in "holes", and it also translated into the good membership and centroid results we present later. Investigating the result of setting ϵ differently is outside the scope of the work at this time.

Prior to presenting experimental results for the TF-algorithm, we introduce a variation of it which can be useful for very large datasets. The idea is to approximate the restricted region by the union of smaller convex regions, with a centroid determined for each. This can be done using standard UTS-clustering with a large [4] number of clusters ("mini-clusters"). We then take the centroids from this clustering as input into what is basically the TF-algorithm above. Denote this as the TF-Minicluster algorithm, where the details are shown in Algorithm 2. Note that this algorithm can be computationally advantageous compared to the standard TF-algorithm, since Step 5 of Algorithm 1 is now required on a much smaller set of points. However, its disadvantage is that we must choose the number of clusters p, with the risk of not approximating the restricted region well.

Algorithm 2 TF-Minicluster Algorithm

1: Perform Steps 1 to 4 in Algorithm 1 on the raw point set Z to give A as before
2: Cluster Z using UTS-clustering with a large number (p) of clusters to give a centroid point set $\bar{Z}$ and the raw point set partitioned into mini-clusters
3: Follows steps 1 to 2 in Algorithm 1 on the centroid point set $\bar{Z}$ to give an intial adjacency matrix $\bar{A}^*$
4: Initialise $\bar{A}$ to zero. Construct $\bar{A}$ by checking, for each non zero element A_{ij}, whether $z_i, z_j \in Z$ are in different mini-clusters. If so, set $\bar{A}_{ij}$ to $\bar{A}^*_{ij}$
5: Perform Steps 5 and 6 in Algorithm 1 on $\bar{A}$

[3] Note that the TF-metric really is a metric, since (a) $d_\epsilon(zi, zj) \geq 0$, (b) $d_\epsilon(zi, zj) = 0$ only when $z_i = z_j$, and (c) the triangle inequality holds since $d_\epsilon(zi, zj)$ is the shortest path, so that $d_\epsilon(zi, zk) + d_\epsilon(zk, zj)$ must necessarily represent a longer (or at most equal length) path. Further, existence of such a path is guaranteed, since z_i,z_j are guaranteed to be connected by an intermediate path of temporally adjacent points.

[4] If too few clusters are used then the union of small regions will not approximate the restricted region well

5 Results

We saw with the CP-PP ratio in Figure 7 that UTS clustering results in centroids lying away from data points. We would hope that this is rectified by our approach, and clearly it will be the case. Since we cluster using k-mediods on (Z, d_{TF}), we know that each centroid must coincide with a data point and so the CP-PP ratio must be zero. This is a desirable outcome, but let us clarify. In the TF-algorithm, we have (i) proposed to restrict the region to that visited by the time series in order to produce useful clustering and (ii) proposed an approach that achieves (i) by using k-mediods on (Z, d_{TF}). Let us confirm that the desirable outcome has been achieved due to (i) and not (ii). We can do this by presenting the CP-PP ratio results for the TF-miniclustering approach, where (ii) is then not applicable.

The CP-PP ratio results for the TF-miniclustering approach are shown in Figure 9. Note how the ratios for all the time series are now down around or below unity. This is in contrast to the results shown in Figure 7 for the UTS-clustering algorithm, where very large CP-PP ratio values were observed. Only the Henon time series still has a higher value of around 3 for small k values, however this is much smaller than its respective value in Figure 7, and it could be brought down by selecting a greater number p of mini-clusters [5]. Hence, unlike for the UTS (and STS) clustering algorithms, the TF-clustering algorithm produces centroids that sit in among their clusters' members. This is a requirement if the useful clustering outcome we desire is to be achieved.

The placement of cluster centroids by our method would then seem, at the "macro" level, to be correct. Lets us look now at the TF-clustering outcomes for specific time series in our data set. We first show the result of TF clustering the CBFV time series in Figure 10 (top). What is immediately obvious is that the three distinct shapes (the Cylinder, Bell and Funnel) now get returned as centroids, as desired, rather than the three sine-type waves observed in Figure 4 for UTS-clustering. This is confirmed in by the membership bars shown in Figure 10 (bottom), i.e. each shape now populates a distinct cluster [6] . We show in Figure 10 (middle) plots of the full window of points over which the delay vectors chosen as centroids were constructed. One of the advantages of the TF algorithm compared to the UTS algorithm is that, since it selects members of Z as cluster centroids, we have available for our centroid representation the full window of time series data points over which the delay vector spanned. This circumvents any centroid discretisation issues caused by introducing a lag.

Next, we show in Figure 11 the result of TF-clustering the Koski-ECG time series. Recall from Figure 5 how UTS clustering returns very strange centroids that look nothing like any part of the time series. Figure 11 shows that the TF-algorithm correctly returns the in-beat and between-beat

[5] For this experiment we chose p as the number of points in the time series divided by 15 (i.e. so that on average there were 15 points in each mini-cluster) where 15 reflected more or less the greatest value we could select before insufficient points remained for clustering at the higher k numbers for the smaller data sets.

[6] Note that the range of time series represented by each cluster is the bar shown plus the length of the delay vector , i.e. q times w, which is why the bars don't seem to be centred properly. This factor must also be taken into account in the membership plots for experiments which follow

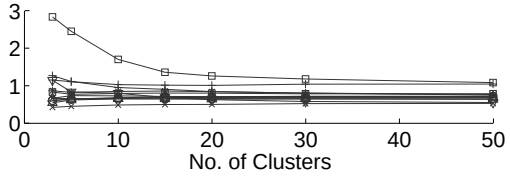


Figure 9: $CP - PP$ Ratio: TF-Miniclustering

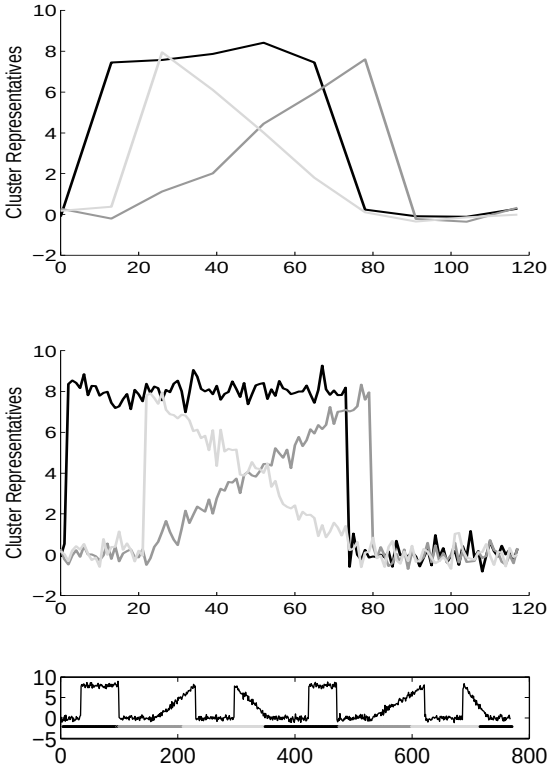


Figure 10: TF Clustering: CBFV results

phases in this time series. This is confirmed by the membership bars shown in the bottom plot [6], where homogeneous bars without the high frequency oscillation between membership evident in Figure 5 have been returned.

Finally, Figure 12 shows the result of TF-clustering the Chaotic Time Series. Recall from Figure 6 how the UTS Clustering algorithm seemed to identify three distinct levels in this time series, but how the oscillations in the time series at these levels disappeared. Figure 12 shows that the TF-algorithm includes these oscillations in its centroids. In essence, it has picked out the three best features to summarise this time series as an oscillatory signal at each of these three levels, and this seems an intuitively appropriate "summary" of this time series.

It would seem that our method provides a means for the proper selection of cluster centroids, and indeed produces clustering membership outcomes more in line with what one would intuitively expect. However, recall that a useful clustering must be meaningful. The final step in our experiments is to show that the TF-algorithm produces meaningful clusterings. We repeat Steps 1 to 6 of Section 2 above, where in Step 4, we cluster with the TF-algorithm in place of UTS-clustering. Figure 13 shows the resulting within versus between distance of clusters for all twelve time series over a range of k values. We can see that, for each time series, the between distances are always greater than the within distance, suggesting that the TF algorithm indeed produces meaningful clustering outcomes.

6 Conclusion

The finding was made in (E.Keogh et al. 2003) that clustering time series using the traditional STS approach was meaningless since (A) the clustering outcomes of different time series were not distinguishable from one another, and (B) that cluster centroids were smoothed. While (E.Keogh et al. 2003) proposed that these two problems were one and the same, i.e. that one could not distinguish between cluster centres of different time series because they were all

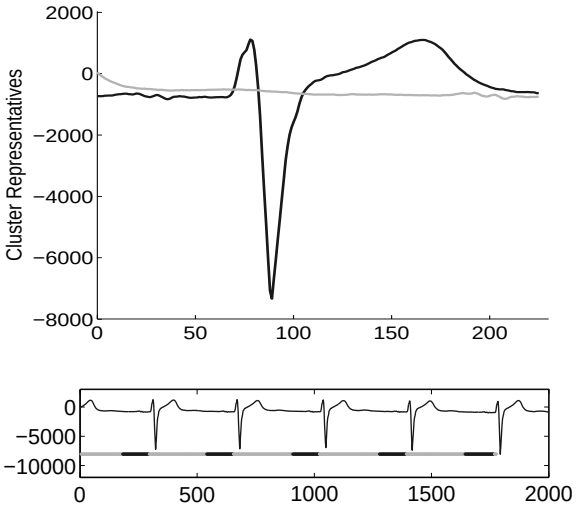


Figure 11: TF Clustering: Koski-ECG results

smoothed and hence alike, we showed in this paper that this is in fact false. Problems (A) and (B) are really two separate problems that can be solved separately. Work in (G.Simon et al. 2006) showed that (A) could be solved by introducing a lag in the delay vector construction process. In this paper we showed, however, that such an approach does not solve (B). Introducing the alternative terminology that a clustering method overcoming problem (A) is meaningful, while one overcoming (A) and (B) is useful, we proposed an approach that produces useful time series clustering. The two key elements of the approach were, (I) unfolding the delay vector distribution by using a lag, and then, (II) clustering only in the "subset" of delay space visited by the time series. We proposed that one should cluster in the subset of delay space on which the underlying physical system that produced the data lives, although, we noted that one does not typically know the extent of this region a-priori. Two choices then seem possible: to cluster in all of delay space as has been the case to date, or to cluster in the subset of delay space visited by the time series. Both lead to meaningful clustering, however we proposed that the question we are typically asking when looking to cluster a time series corresponds to the latter. In essence, we want a summary of the time series as the k features existing in the time series which best summarise it. Finding the low dimensional subset of higher

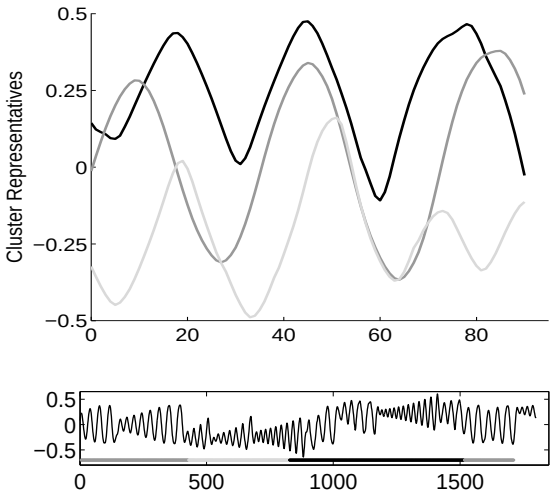


Figure 12: TF Clustering: Chaotic results

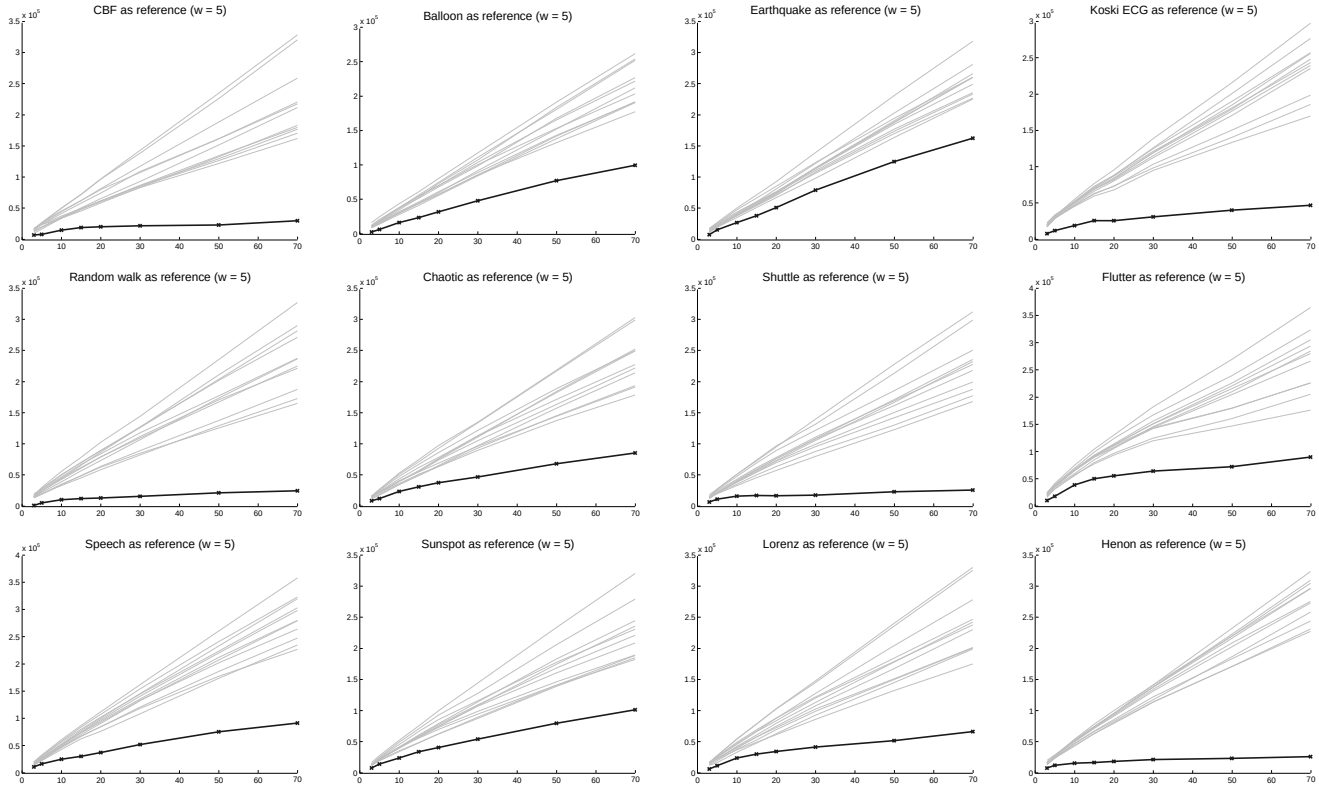


Figure 13: Results of the within-between distance experiments using TF Clustering

dimensional spaces on which a data set lives is not new in the literature, although the fact that time series data sets have temporal ordering, and that we are looking to cluster the data set once the relevant subset is found, adds novelty here. We noted the close parallels of seminal work in the dimensionality-reduction-of-data-sets literature and the approach proposed in this paper. Experiments to validate our approach were conducted on 12 real life and synthetic data sets. These experiments indicated that our approach could overcome both problems (A) and (B); it produced meaningful clusterings, and it produced cluster representatives that well represented (were located in among) the data points in their respective clusters. Quite a number of solutions have been suggested to the STS-clustering dilemma since it was first identified in (E.Keogh et al. 2003). However, to our knowledge, this is the first method that directly addresses both the problems (A) and (B) identified there. As such, we propose the TF-algorithm as a means for obtaining useful clustering outcomes when the clustering of time series is required.

References

A.Denton (2005), Kernel-density-based clustering of time series subsequences using a continuous random-walk noise model, *in* 'Proceedings of IEEE International Conference on Data Mining', Houston, USA.

B.S.Everitt, S.Landau & M.Leese (2001), *Clustering Analysis*, Wiley.

D.Goldin, R.Mardales & G.Nagy (2006), In search of meaning for time series subsequence clustering: matching algorithms based on a new distance measure, *in* 'Proceedings of Conference of Information and Knowledge Management', Arlington, USA.

E.Keogh (2002), 'The ucr time series data mining archive', `http://www.cs.ucr.edu/~eamonn/TSDMA/index.html`.

E.Keogh, J.Lin & W.Truppel (2003), Clustering of time series subsequences is meaningless: Implications for previous and future research, *in* 'Proceedings of the International Conference of Data Mining'.

G.Simon, J.A.Lee & M.Verleysen (2006), 'Unfolding preprocessing for meaningful time series clustering', *Neural Networks* **19**, 877–888.

J.B.Tenenbaum, Silva, V. & J.C.Langford (2000), 'A global geometric framework for nonlinear dimensionality reduction', *Science* **290**, 2319–2322.

J.R.Chen (2007), 'Making clustering in delay vector space meaningful', *Knowledge and Information Systems, an International Journal* **11**(3), 369–385.

K.Peker (2005), Subsequence time series (sts) clustering techniques for meaningful pattern discovery, *in* 'International Conference Integration of Knowledge Intensive Multi-Agent Systems (KIMAS)', Waltham, USA.

M.Breitenbach & G.Z.Grundic (2005), Clustering through ranking on manifolds, *in* 'Proceedings of the 22nd International Conference on Machine Learning', Bonn, Germany.

R.M.Haralick & R.Harpaz (2005), Linear manifold clustering, *in* 'Proceedings of the International Conference on Machine Learning and Data Mining', Leipzig.

Z.R.Struzik (2003), *Foundations of Intelligent Systems*, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, chapter Time Series Rule Discovery: Tough, Not Meaningless, pp. 32–39.

A Two-Step Classification Approach to Unsupervised Record Linkage

Peter Christen

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

Abstract

Linking or matching databases is becoming increasingly important in many data mining projects, as linked data can contain information that is not available otherwise, or that would be too expensive to collect manually. A main challenge when linking large databases is the classification of the compared record pairs into matches and non-matches. In traditional record linkage, classification thresholds have to be set either manually or using an EM-based approach. More recently developed classification methods are mainly based on supervised machine learning techniques and thus require training data, which is often not available in real world situations or has to be prepared manually. In this paper, a novel two-step approach to record pair classification is presented. In a first step, example training data of high quality is generated automatically, and then used in a second step to train a supervised classifier. Initial experimental results on both real and synthetic data show that this approach can outperform traditional unsupervised clustering, and even achieve linkage quality almost as good as fully supervised techniques.

Keywords: Data linkage, data matching, deduplication, entity resolution, clustering, support vector machines, quality measures.

1 Introduction

With many businesses, government organisations and research projects collecting large amounts of data, techniques that allow efficient processing, analysing and mining of massive databases have in recent years attracted interest from both academia and industry. Increasingly, data from various sources has to be linked, matched and aggregated in order to improve data quality, or to enrich existing data with additional information. Similarly, detecting and removing duplicate records that relate to the same entity within one database is often required in the data pre-processing step of many data mining projects. The aim of such linkages and deduplications is to match and aggregate all records that relate to the same entity, such as a patient, a customer, a business, a product description, a publication, or a genome sequence.

Record or data linkage and deduplication can be used to improve data quality and integrity (Winkler 2004), to allow re-use of existing data sources for new studies, and to reduce costs and efforts in data acquisition. In the health sector, for example, linked data might contain information that is needed to improve health policies (Kelman et al. 2002), and that traditionally has been collected with time consuming and expensive survey methods. Statistical agencies routinely link census data for further analysis (Gill 2001), while businesses often deduplicate their databases to compile mailing lists or link them for collaborative e-Commerce projects. Within taxation offices and departments of social security, record linkage is used to identify people who register for assistance multiple times or who work and collect unemployment benefits. Another application of current interest is the use of record linkage in crime and terror detection. Security agencies and crime investigators increasingly rely on the ability to quickly access files for a particular individual (Wang et al. 2006), which may help to prevent crimes and terror by early intervention.

The problem of finding similar entities does not only apply to records that refer to persons. In bioinformatics, record linkage can help finding genome sequences in large data collections that are similar to a new, unknown sequence at hand. Increasingly important is the removal of duplicates in the results returned by Web search engines and automatic text indexing systems, where copies of documents (for example bibliographic citations) have to be identified and filtered out before being presented to the user (Bhattacharya and Getoor 2007). Finding and comparing consumer products from several online stores is another application of growing interest (Bilenko et al. 2005). As product descriptions are often slightly different, matching them becomes difficult.

If unique entity identifiers (or keys) are available in all databases to be linked, then the problem of linking at the entity level becomes trivial: a simple database *join* is all that is required. However, in most cases no unique keys are shared by all databases, and more sophisticated linkage techniques need to be applied. These techniques can be broadly classified into deterministic, probabilistic, and modern approaches (Christen and Goiser 2007, Winkler 2006).

A general schematic outline of the record linkage process is given in Figure 1. As most real-world data collections contain noisy, incomplete and incorrectly formatted information, data cleaning and standardisation are important pre-processing steps for successful record linkage, and also before data can be loaded into data warehouses or used for further mining (Rahm and Do 2000). A lack of good quality data can be one of the biggest obstacles to successful record linkage and deduplication (Clarke 2004). The main task of data cleaning and standardisation is the conversion of the raw input data into well defined, consistent forms, as well as the resolution of inconsistencies in the way information is represented and encoded (Churches et al. 2002).

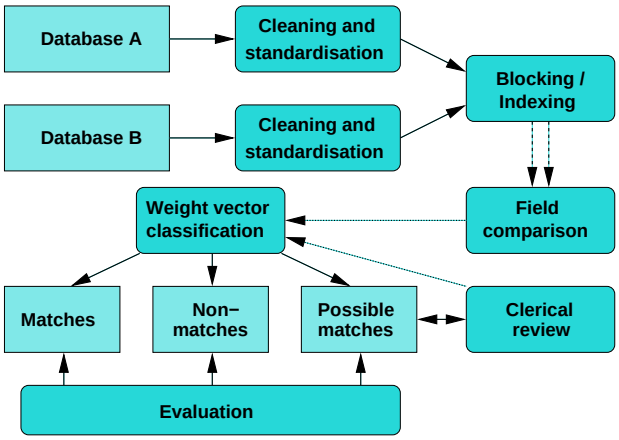


Figure 1: General record linkage process. The output of the blocking step are candidate record pairs, while the comparison step produces weight vectors with numerical similarity weights.

R1:	Christine	Smith	42	Main	Street
R2:	Christina	Smith	42	Main	St
R3:	Bob	O'Brian	11	Smith	Rd
R4:	Robert	Bryce	12	Smythe	Road

WV(R1,R2):	0.9	1.0	1.0	1.0	0.9
WV(R1,R3):	0.0	0.0	0.0	0.0	0.0
WV(R1,R4):	0.0	0.0	0.5	0.0	0.0
WV(R2,R3):	0.0	0.0	0.0	0.0	0.0
WV(R2,R4):	0.0	0.0	0.5	0.0	0.0
WV(R3,R4):	0.7	0.4	0.5	0.7	0.9

Figure 2: Four example records (made of given name and surname; and street number, name and type attributes) and the corresponding weight vectors resulting from the comparisons of these records.

If two databases, **A** and **B**, are to be linked, potentially each record from **A** has to be compared with all records from **B**. The total number of potential record pair comparisons thus equals the product of the size of the two databases, $|\mathbf{A}| \times |\mathbf{B}|$, with $|\cdot|$ denoting the number of records in a database. Similarly, when deduplicating a database, **A**, the total number of potential record pair comparisons is $|\mathbf{A}| \times (|\mathbf{A}|-1)/2$, as each record potentially has to be compared to all others. The performance bottleneck in a record linkage or deduplication system is usually the expensive detailed comparison of fields (or attributes) between pairs of records (Baxter et al. 2003, Christen and Goiser 2007), making it unfeasible to compare all pairs when the databases are large. Assuming there are no duplicate records in the databases (i.e. one record in database **A** can only match to one record in database **B**, and vice versa), then the maximum number of true matches corresponds to the number of records in the smaller database. Therefore, while the computational efforts increase quadratically, the number of potential true matches only increases linearly when linking larger databases. This also holds for deduplication, where the number of duplicate records is always less than the number of records in a database.

To reduce the large amount of potential record pair comparisons, record linkage methods employ some form of indexing or filtering techniques, collectively known as *blocking* (Baxter et al. 2003): a single record attribute or a combination of attributes, often called the *blocking key*, is used to split the databases into blocks. All records that have the same value in the blocking key will be inserted into one block, and candidate record pairs are then generated only from records within the same block. These candidate pairs are compared using a variety of comparison functions applied to one or more (or a combination of) record attributes. These functions can be as simple as an exact string or a numerical comparison, can take variations and typographical errors into account (Cohen et al. 2003, Christen 2006), or can be as complex as a distance comparison based on look-up tables of geographic locations (longitudes and latitudes).

Each comparison returns a numerical similarity value (called *matching weight*), often in normalised form. Two attribute values that are equal, therefore, would have a similarity of 1, while the similarity of two completely different values would be 0. Attribute values that are somewhat similar would have a similarity value somewhere between 0 and 1. As illustrated in Figure 2, a vector (called *weight vector*) is formed for each compared record pair containing all the matching weights calculated by the different comparison functions. These weight vectors are then used to classify record pairs into *matches*, *non-matches*, and *possible matches*, depending upon the decision model used (Christen and Goiser 2007, Fellegi and Sunter 1969, Gu and Baxter 2006). Record pairs that were removed by the blocking process are classified as non-matches without being compared explicitly.

Two records that have the same values in all their attributes will with high likelihood refer to the same entity, as it is very unlikely that two entities have the same values in all their attributes. The weight vector calculated when comparing such a pair of records will have matching weights of 1 in all vector elements. On the other hand, weight vectors that have 0 or very low similarity values in all their elements are with high likelihood the result of a comparison of two records that refer to different entities, as it is highly unlikely that two records that refer to the same entity have different values in all their record attributes. For example, even if a woman changes her surname and her address when she gets married, her date of birth and her maiden name will stay the same.

From this follows that it is often easy to classify with high accuracy record pairs that are very similar as matches, and pairs that are very dissimilar as non-matches. On the other hand, it is much more difficult to classify pairs that have some similar and some dissimilar attribute values. This is illustrated in Figure 2, where records *R1* and *R2* are very similar, with only two minor difference in the given name and street type attributes (which usually are taken care of in the data cleaning and standardisation step (Churches et al. 2002)), and thus very likely refer to the same person. On the other hand, records *R3* and *R4* are more different to each other, and it is not obvious if they refer to the same person.

Based on the above observations, it is possible to automatically extract training examples (weight vectors) from the set of all weight vectors that with high likelihood correspond to true matches or true non-matches, and to then use these weight vectors to train a supervised classifier. From the six weight vectors shown in Figure 2, *WV(R1,R2)* can be used as a training example for matches, while *WV(R1,R3)* and *WV(R2,R3)*, and possibly even *WV(R1,R4)* and *WV(R2,R4)*, can be used for non-matches.

This two-step approach to automated record pair classification, which has been inspired by similar approaches that were developed for text classification (Basu et al. 2002, Liu et al. 2003, Nigam et al. 2000, Yu et al. 2002), is presented in more detail in Section 3, and evaluated experimentally in Section 4. First, in the following section, an overview of related research is presented. Conclusions and an outlook to future work is then given in Section 5.

2 Related Work

The classical probabilistic record linkage approach, as developed by Fellegi and Sunter (1969), has been improved in recent years mainly through application of the expectation-maximisation (EM) algorithm for better parameter estimation in record pair classification (Winkler 2000), and through the use of approximate string comparisons to calculate partial agreement weights when attribute values have typographical variations (Christen 2006, Winkler 2006).

In the late 1990s researchers started to explore the use of techniques originating in machine learning, data mining, artificial intelligence, information retrieval and database research to improve the linkage process. Many of these approaches are based on supervised learning techniques and assume that training data is available (i.e. record pairs with known true match and true non-match status). However, such training examples are often not available in real world situations, or have to be prepared manually (an expensive and time consuming process).

One supervised approach is to learn distance measures for approximate string comparisons, such as the costs for character inserts, deletes and substitutions for edit-distance (Bilenko and Mooney 2003, Cohen et al. 2003), with the aim to adapt similarity computations to a particular data domain. Decision tree induction (Elfeky et al. 2002, Neiling 2005, Tejada et al. 2002) and support vector machines (SVM) (Nahm et al. 2002) are two popular supervised machine learning techniques that have been employed successfully for record pair classification. These techniques usually achieve better linkage quality compared to unsupervised methods.

In (Elfeky et al. 2002), three approaches to record pair classification are described; the first based on supervised decision trees, the second using unsupervised k-means clustering (with three clusters, one each for matches, possible matches and non-matches), and the third being a hybrid approach that combines the first two to overcome the problem of lack of training data. In this hybrid approach, a sub-set of weight vectors is clustered in a first step (again into matches, possible matches and non-matches), and the match and non-match clusters are then used as training data for a supervised classifier in a second step. Both the fully supervised and hybrid approach outperformed the clustering approach in experimental studies.

Active learning is another approach, aimed at reducing the amount of training data required. In (Sarawagi and Bhamidipaty 2002), a system is described that presents a difficult to classify record pair to a user for manual classification. After such a pair is classified manually, it is added to the training set and the classifiers are re-trained. This process is repeated until all record pairs are successfully classified. The authors reported that manually classifying less than 100 training pairs using their approach provided better results than a fully supervised approach that used 7,000 randomly selected examples. A similar approach has been presented in (Tejada et al. 2002), where a committee of decision trees is used to learn a set of rules that describe linkages.

Unsupervised clustering techniques have been investigated both for improved blocking (Cohen and Richman 2002, McCallum et al. 2000) and for automatic record pair classification (Elfeky et al. 2002). The k-means clustering algorithm has been used in (Gu and Baxter 2006) to group weight vectors into matches and non-matches (i.e. $k = 2$). In this approach, a user can identify a 'fuzzy' region in the middle between the two cluster centroids where the difficult to classify record pairs are located. These pairs will then be given to the user for manual clerical review. Using synthetic data, it was shown that this approach can significantly reduce the number of record pairs that have to be reviewed manually, while keeping high linkage quality. In (Goiser and Christen 2006), the clustering techniques k-means and farthest-first were compared with supervised decision tree induction on both synthetic and real data sets. Surprisingly, the simple farthest-first technique achieved results comparable to decision trees.

Another area where unsupervised techniques have been explored in recent years is entity resolution of relational data based on relational clustering (Bhattacharya and Getoor 2007). While the techniques described so far assume that only similarities between attribute values of record pairs are available for classification, in relational data the entities have additional relational information that can be used to improve the quality of entity resolution. Relational information includes, for example, census databases that contain a family relationship attribute (with values such as 'married to', 'dependent of', or 'parent of'); or bibliographic data where, besides the name of a paper, a publication record also contains a list of authors. Two author names in different publications that have several co-authors in common in other publications will more likely refer to the same real person compared to an author with the same name that has different co-authors. Experimental results (Bhattacharya and Getoor 2007) on various data sets have shown that collective relational entity resolution outperforms non-relational entity resolution that is based on record pair similarities only. However, there are still many situations in the real world where no relational data is available, and this paper concentrates on improving the unsupervised classification of such non-relational data.

The two-step approach presented here has been inspired by similar approaches to text classification, where often only a small number of labeled positive examples and a very large number of unlabeled examples are available. The aim is then to learn a binary classifier from these positive and unlabeled examples. In (Yu et al. 2002), the PEBL approach is presented, which is based on iteratively training a SVM using the positive and a selected set of strong negative examples. More unlabeled examples are included into the negative training set as the trained classifier becomes more accurate, until all unlabeled examples are classified. A comparison of different approaches to learning from positive and unlabeled examples is given in (Liu et al. 2003). The techniques compared were PEBL, Naïve Bayes classification, Rocchio text classification in combination with SVM, and an EM based approach (called S-EM) that uses 'spy' documents, positive examples that are inserted into the set of unlabeled documents to better model their distributions (Liu et al. 2002). A new approach, that uses a biased SVM formulation, is then proposed that achieved better classification results than all previous methods (Liu et al. 2003).

In a related text classification scenario, only small numbers of both positive and negative labeled training examples, as well as a large number of unlabeled examples, are available. In (Nigam et al. 2000), a combination of the EM and Naïve Bayes classifiers is presented. Training is started using only the labeled data, and then iteratively refined using the unlabeled examples. The experimental results presented showed that this approach was able to reduce classification errors by up to 30%.

Also related to the work presented here is semi-supervised clustering (Basu et al. 2002), which is based on the idea of using a small amount of labeled data to initialise the cluster centroids, for example for k-means, rather than using random centroid ini-

tialisation. Experimental results discussed in (Basu et al. 2002) show that this can significantly improve cluster quality. In the area of record linkage, such an approach can be taken for classifying weight vectors, by initialising two cluster centroids, one to the exact similarity values (matches) and the other to total dissimilarity values (non-matches). Such a clustering approach will be compared to other classification techniques in Section 4 below.

3 Two-step Record Pair Classification

The idea behind the approach presented in this paper is based on the following two assumptions. First, the weight vectors generated in the comparison step that have exact or high similarity values in all their vector elements were with high likelihood produced when two records were compared that refer to the same entity, as it is very unlikely that two different entities have high similarities in all their attributes. Second, weight vectors with mostly low similarity values were with high likelihood produced when two records were compared that refer to different entities, as it is highly unlikely that two records that refer to the same entity have different values in all their attributes.

Thus, the hypothesis investigated in this paper is that it is possibly to select in a first step weight vectors as training examples that with high likelihood correspond to either true matches or true non-matches, and to then use these examples in a second step to train a supervised classifier. This paper concentrates on the first step, and presents and evaluates several approaches to automatically select training examples. Combined, these two steps will allow fully automated, unsupervised record pair classification, without the need to know the true match and non-match status of the weight vectors produced in the comparison step.

3.1 Step 1: Selection of Training Examples

There are two main approaches to selecting training examples, either using thresholds or nearest-based. As illustrated in Figure 1, pairs of records that were generated in the blocking step are compared using d comparison functions (with $d \geq 1$), resulting in a set $\mathbf{W}$ of weight vectors $\mathbf{w}_i$ ($1 \leq i \leq |\mathbf{W}|$) of length d containing matching weights (similarity values), with $|\cdot|$ denoting the number of elements in a set. It is assumed that all comparison functions return normalised similarity values between 1.0 (exact similarity) and 0.0 (total dissimilarity), i.e. $0.0 \leq \mathbf{w}_i[j] \leq 1.0, 1 \leq j \leq d, \forall \mathbf{w}_i \in \mathbf{W}$. The weight vector that contains exact similarities in all its vector elements is denoted by $\mathbf{m}$ (i.e. $\mathbf{m}[j] = 1.0, 1 \leq j \leq d$), and the weight vector that contains total dissimilarities only by $\mathbf{n}$ (i.e. $\mathbf{n}[j] = 0.0, 1 \leq j \leq d$).

The aim of the training example selection process is to choose weight vectors from $\mathbf{W}$ that with very high likelihood correspond to true matches and true non-matches, respectively, and to insert them into two sets, the match example training set, $\mathbf{W}_M$, and the non-match example training set, $\mathbf{W}_N$. Generally, only a fraction of all weight vectors will be selected for training, and thus it is expected that $(|\mathbf{W}_M| + |\mathbf{W}_N|) \ll |\mathbf{W}|$. In the following, the two approaches to training example selection are presented in more detail.

3.1.1 Threshold-based Selection

In this approach, one threshold for matches, t_m (with $0.0 < t_m < 1.0$), and one for non-matches, t_n (with $0.0 < t_n < 1.0$), are used to select weight vectors that have all their similarity values either within t_m of the exact match value (1.0) or within t_n of the total dissimilarity value (0.0). More formally, the match and non-match example sets $\mathbf{W}_M$ and $\mathbf{W}_N$ are formed according to:

$$\begin{aligned} \mathbf{W}_M &= \{\mathbf{w}_i \in \mathbf{W} : (\mathbf{m}[j] - \mathbf{w}_i[j]) \leq t_m, 1 \leq j \leq d\}, \\ \mathbf{W}_N &= \{\mathbf{w}_i \in \mathbf{W} : (\mathbf{n}[j] + \mathbf{w}_i[j]) \leq t_n, 1 \leq j \leq d\}. \end{aligned}$$

Depending upon the values of t_m and t_n, there is the possibility that a weight vector could be included into both training example sets $\mathbf{W}_M$ and $\mathbf{W}_N$. In such a situation, this weight vector will be removed from both $\mathbf{W}_M$ and $\mathbf{W}_N$, as it cannot be a good quality training example for both matches and non-matches. For example, this would happen when $t_m = t_n = 0.6$ for a weight vector which has all similarity values set to 0.5, i.e. $\mathbf{w}_i[j] = 0.5, 1 \leq j \leq d$.

3.1.2 Nearest-based Selection

Rather than using thresholds, in this approach the weight vectors closest to $\mathbf{m}$ are selected into $\mathbf{W}_M$, and the weight vectors closest to $\mathbf{n}$ into $\mathbf{W}_N$. More formally, if x_m and x_n (with $x_m > 0$ and $x_n > 0$) are the number of weight vectors to be selected into $\mathbf{W}_M$ and $\mathbf{W}_N$, respectively, and the distance between two weight vectors is calculated using the Manhattan distance as $dist(\mathbf{w}_i, \mathbf{w}_k) = \sum_{j=1}^{d} |\mathbf{w}_i[j] - \mathbf{w}_k[j]|$, then the training example sets are formed according to:

$$\begin{aligned} \mathbf{W}_M &= \{\mathbf{w}_i \in \mathbf{W}, \mathbf{w}_k \notin \mathbf{W}_M : dist(\mathbf{m}, \mathbf{w}_i) < dist(\mathbf{m}, \mathbf{w}_k)\}, \\ \mathbf{W}_N &= \{\mathbf{w}_i \in \mathbf{W}, \mathbf{w}_k \notin \mathbf{W}_N : dist(\mathbf{w}_i, \mathbf{n}) < dist(\mathbf{w}_k, \mathbf{n})\}, \end{aligned}$$

with $x_m = |\mathbf{W}_M|$ and $x_n = |\mathbf{W}_N|$.

There are two variations of how the x_m and x_n nearest vectors can be chosen. First, they can be selected regardless if some of them contain the same values in all of their vector elements. For example, there might be a number of weight vectors that contain only exact match values (i.e. that are equal to $\mathbf{m}$) if there are pairs of records that are exact matches, i.e. that have the same values in all compared attributes. Similarly, as illustrated in Figure 2, there will be a large number of weight vectors that only contain total dissimilarity values (i.e. weight vectors that are equal to $\mathbf{n}$). In the worst case, the weight vectors selected into $\mathbf{W}_M$ will all be equal to $\mathbf{m}$ and the weight vectors selected into $\mathbf{W}_N$ will all be equal to $\mathbf{n}$. This situation would not be very useful for training the classifier in step two. Thus, in order to make sure weight vectors with different values are selected, the x_m and x_n nearest *unique* vectors can be inserted into the sets $\mathbf{W}_M$ and $\mathbf{W}_N$ of training examples. These two variations will be referred to as *non-unique* and *unique* nearest in the experimental results presented in Section 4 below.

A second variation in the nearest-based approach is how to choose the values of x_m and x_n. Both can be set to the same value, resulting in a balanced classification problem that has the same number of match and non-match training examples. However, as discussed in Section 1 earlier, the number of true non-matches in the set of weight vectors generated by the blocking and comparison steps will likely be much larger than the number of true matches, because the number of true matches is usually limited by the size of the smaller data set. Classifying the weight vector set $\mathbf{W}$ is therefore an imbalanced classification problem, and this should be reflected in the number of training examples provided to the classifier in step

Data set	Number of records	Task	Pairs completeness	Reduction ratio	Number of weight vectors (i.e. $\|\mathbf{W}\|$)	Ratio of true matches to true non-matches
Census	449 + 392	Linkage	1.000	0.988	2,093	1 / 5.40
Restaurant	864	Deduplication	1.000	0.713	106,875	1 / 953.24
DS-Gen	1,000	Deduplication	0.957	0.995	2,475	1.13 / 1
DS-Gen	2,500	Deduplication	0.940	0.997	9,878	1 / 2.06
DS-Gen	5,000	Deduplication	0.953	0.997	35,491	1 / 4.48
DS-Gen	10,000	Deduplication	0.948	0.997	132,532	1 / 9.32

Table 1: Data sets used in experiments. See Section 4.1 for more details.

two. An estimation of the ratio of matches to non-matches, r, can be calculated based on the number of records in both data sets, $|\mathbf{A}|$ and $|\mathbf{B}|$, and the number of weight vectors $|\mathbf{W}|$:

$$r = \frac{min(|\mathbf{A}|, |\mathbf{B}|)}{|\mathbf{W}| - min(|\mathbf{A}|, |\mathbf{B}|)}. \quad (1)$$

The number of weight vectors selected into the match examples training set $\mathbf{W}_M$ will therefore usually be smaller than the number of vectors selected into the non-match examples training set $\mathbf{W}_N$. In the experiments presented in Section 4 below, the results for this variation will be shown in two separate tables.

3.2 Step 2: Classification of Record Pairs

Once example training data for matches, $\mathbf{W}_M$, and non-matches, $\mathbf{W}_N$, has been selected, any binary classifier can be trained on them, followed by the classification of the weight vectors that have not been selected as training examples, i.e. $\mathbf{W}_T = \mathbf{W} \setminus (\mathbf{W}_M \cup \mathbf{W}_N)$. In this paper, a support vector machine (SVM) classifier will be used, as this technique can handle high-dimensional data and has shown to be robust to noisy data. The use of other classifiers, such as decision trees, is possible and will be investigated as part of future work.

One important issue that is also left for future work is that the example training data generated automatically in the first step will be linearly separable, as the two training sets only contain examples that are either close to the exact match value or close to the total dissimilarity value. Thus, there will be a 'gap' between the match and non-match training examples. Similar to the inclusion of 'spy' documents in the S-EM approach (Liu et al. 2002), adding randomly sampled weight vectors from this 'gap' into the training example sets should improve the overall classification accuracy. This idea is currently being implemented and results will be reported elsewhere.

4 Experimental Evaluation

In this section, the different approaches to automatically select training examples for matches and non-matches will be compared with three other classification methods. The first is a linear kernel SVM that uses all weight vectors and their match status for supervised classification (10-fold cross validation results are reported). The second is the standard k-means clustering approach using Euclidean distance and with two clusters (one for the matches and one for the non-matches), with the cluster centroids initialised to the exact match vector $\mathbf{m}$ and total dissimilarity vector $\mathbf{n}$, respectively. The third is an 'optimal threshold' classifier that has access to the match status of all weight vectors, and that emulates an optimal probabilistic approach (Fellegi and Sunter 1969). It sums each weight vector into a single matching weight (i.e. it generates 1-dimensional weight vectors), and then finds the optimal classification threshold using these matching weights that minimises the number of false matches and false non-matches.

All techniques described here were implemented in the *Febrl* (Christen et al. 2004) open source record linkage system,[1] which is written in the Python programming language. For SVM classification the *PyML*[2] Python module was used, which is based on the *libsvm* library (Chang and Lin 2001). The default linear kernel SVM method from *PyML* was chosen in all experiments presented here. Further experiments using SVMs with non-linear kernels and other classification approaches are planned for future work. All reported experiments were carried out on a Dell Optiplex GX280 with an Intel Pentium 3 GHz CPU and 2 Gigabytes of main memory, running Linux 2.6.20 (Ubuntu 7.04 Feisty Fawn) and using Python 2.5.1.

4.1 Data Sets and Linkage Setup

The proposed approaches were evaluated using both real and synthetic data sets, which are summarised in Table 1. Two small real data sets were taken from the *SecondString* toolkit,[3] while artificial data sets of various sizes were created using the *Febrl* data set generator (Christen 2005). This generator works by first creating a number of *original* records based on frequency tables containing real world names (given- and surname) and addresses (street number, name and type; postcode; suburb and state name), followed by the random generation of *duplicates* of these records based on modifications (like inserting, deleting or substituting characters, and swapping, removing, inserting, splitting or merging words), also based on real error characteristics. All data sets generated for this paper contained 60% original and 40% duplicate records, with up to nine duplicates for one original record (the number of duplicates created per original record are 'Zipf' distributed), and with a maximum of three modifications per attribute and maximum ten modifications per record.

A standard blocking approach (Baxter et al. 2003) was used for all experiments reported here, with the blocking keys being combinations of name, address and postcode values. For the record pair comparison step, the Winkler (Christen 2006, Winkler 2004) approximate string comparator (commonly employed in record linkage for name comparisons) was used on the name and address attributes. Additionally, for the Census and synthetic data sets, character difference comparisons were used on the zipcode, postcode, street number and state abbreviation attributes.

The pairs completeness measure shown in Table 1 is the number of true matched record pairs generated by a blocking technique divided by the total number of true matched pairs (Christen and Goiser 2007). It measures how effective a blocking technique is in generating true matched record pairs. Pairs completeness

[1] http://febrl.sourceforge.net
[2] http://pyml.sourceforge.net
[3] http://secondstring.sourceforge.net

Data set	Train set	Threshold				
		0.1	0.3	0.5	0.7	0.9
Census	$\mathbf{W}_M$	0	100	96.2	73.4	67.9
	$\mathbf{W}_N$	0	0	100	100	100
Restau-	$\mathbf{W}_M$	100	98.5	4.5	0.19	0.2
rant	$\mathbf{W}_N$	0	0	100	100	100
DS-Gen	$\mathbf{W}_M$	0	100	100	100	100
1,000	$\mathbf{W}_N$	100	100	100	99.0	86.1
DS-Gen	$\mathbf{W}_M$	100	100	100	99.8	99.7
2,500	$\mathbf{W}_N$	100	100	100	99.4	92.0
DS-Gen	$\mathbf{W}_M$	100	100	100	98.0	96.5
5,000	$\mathbf{W}_N$	100	100	100	99.7	96.3
DS-Gen	$\mathbf{W}_M$	100	100	100	95.5	93.6
10,000	$\mathbf{W}_N$	99.2	99.7	100	99.9	98.3

Table 2: Quality of threshold-based training example selection. $\mathbf{W}_M$ denotes the match example training set, and $\mathbf{W}_N$ the non-match example training set. All result values are given as percentages.

Data set	Train set	Non-unique near.			Unique nearest		
		1%	5%	10%	1%	5%	10%
Census	$\mathbf{W}_M$	100	100	81.8	100	100	79.9
	$\mathbf{W}_N$	100	100	100	100	100	100
Restau-	$\mathbf{W}_M$	9.8	2.0	1.0	5.6	1.1	0.59
rant	$\mathbf{W}_N$	100	100	100	100	100	100
DS-Gen	$\mathbf{W}_M$	100	100	100	100	100	100
1,000	$\mathbf{W}_N$	100	96.7	95.5	100	95.9	95.5
DS-Gen	$\mathbf{W}_M$	100	100	100	100	100	100
2,500	$\mathbf{W}_N$	99.0	98.4	98.3	99.0	98.4	98.2
DS-Gen	$\mathbf{W}_M$	100	100	99.0	100	100	99.0
5,000	$\mathbf{W}_N$	100	99.8	99.5	99.7	99.7	99.6
DS-Gen	$\mathbf{W}_M$	100	99.0	75.4	100	98.6	74.1
10,000	$\mathbf{W}_N$	100	99.8	99.7	99.8	99.8	99.7

Table 3: Quality of balanced nearest-based training example selection.

Data set	Train set	Non-unique near.			Unique nearest		
		1%	5%	10%	1%	5%	10%
Census	$\mathbf{W}_M$	100	100	100	100	100	100
	$\mathbf{W}_N$	100	100	100	100	100	100
Restau-	$\mathbf{W}_M$	100	100	90.8	100	76.7	58.6
rant	$\mathbf{W}_N$	100	100	100	100	100	100
DS-Gen	$\mathbf{W}_M$	100	100	100	100	100	100
1,000	$\mathbf{W}_N$	100	96.7	95.5	100	95.9	95.5
DS-Gen	$\mathbf{W}_M$	100	100	100	100	100	100
2,500	$\mathbf{W}_N$	99.0	98.4	98.3	99.0	98.4	98.2
DS-Gen	$\mathbf{W}_M$	100	100	100	100	100	100
5,000	$\mathbf{W}_N$	100	99.8	99.5	99.7	99.7	99.6
DS-Gen	$\mathbf{W}_M$	100	100	100	100	100	100
10,000	$\mathbf{W}_N$	99.9	99.8	99.7	99.8	99.8	99.7

Table 4: Quality of imbalanced nearest-based training example selection.

corresponds to the *recall* measure as used in information retrieval. The reduction ratio measure, rr, is the number of record pairs generated by the blocking process divided by the number of all possible record pairs. For a linkage between two data sets, $\mathbf{A}$ and $\mathbf{B}$, $rr = 1.0 - |\mathbf{W}|/(|\mathbf{A}| \times |\mathbf{B}|)$ (with $\mathbf{W}$ the set of weight vectors generated in the comparison step), while for a deduplication $rr = 1.0 - 2|\mathbf{W}|/(|\mathbf{A}| \times (|\mathbf{A}| - 1))$. The more record pairs are removed by a blocking technique the higher the reduction ratio value becomes. However, reduction ratio does not take the quality of the generated candidate record pairs into account (how many are true matches or not). The ratio of matches to non-matches shown in Table 1 refers to the corresponding number of weights vectors in $\mathbf{W}$.

4.2 Quality Measures

The quality of the training examples selected in step one (shown in Tables 2, 3 and 4) are given as the percentage of correctly selected weight vectors, i.e. the percentage of true matches in the match examples training set $\mathbf{W}_M$, and the percentage of true non-matches in the non-match examples training set $\mathbf{W}_N$.

Due to the usually imbalanced distribution of matches and non-matches in the weight vector set $\mathbf{W}$, the commonly used accuracy measure is not suitable for assessing the quality of record linkage (Christen and Goiser 2007). The large number of non-matches would dominate the accuracy measure and yield results that are too optimistic. Instead, the F-measure, the harmonic mean of precision P and recall R, is used for measuring classifier quality: $F = 2PR/(P + R)$, with $P = TP/(TP + FP)$ and $R = TP/(TP + FN)$. TP is the number of true positives (true matched record pairs classified as matches), TN the number of true negatives (true non-matched record pairs classified as non-matches), FN the number of false negatives (true matched record pairs classified as non-matches), and FP the number of false positives (true non-matched record pairs classified as matches).

4.3 Results and Discussion

Tables 2, 3 and 4 show the quality of the training examples selected using the different approaches discussed in Section 3.1. As can be seen clearly, in many cases the selected weight vectors are of very high quality, i.e. they are all (or almost all) true matches and true non-matches. The threshold-based approach is problematic, as threshold values that are set too low will possibly result in no weight vectors being selected into a training set. The nearest-based approach overcomes this problem, especially the imbalanced nearest-based selection, which produced very good quality training data in almost all experiments. In the balanced nearest-based approach, it seems that too many weight vectors are selected into the match example training set $\mathbf{W}_M$, resulting in a loss of its quality, as with increasing training set size more false matches (false positives) will be selected due to the imbalanced numbers of true matches and non-matches in the weight vector set $\mathbf{W}$.

For the Census and Restaurant data sets, the 0% values for the threshold-based approach in Table 2 indicate that each of the record pairs compared had similar or equal values in at least one of the compared record attributes, while for the larger synthetic data sets there were record pairs that had no similar attribute values at all. This means that the blocking step for the synthetic data sets could be improved, as record pairs that have no similar attribute values at all clearly should not be compared using the computationally expensive comparison functions.

As can be seen from Table 5 and Figures 3, 4 and 5, using the automatically selected training example sets for classification produced results of a wide variety, ranging from almost as good as the supervised optimal threshold and SVM classifiers, to F-measure values much lower than those of k-means clustering. With most data sets, the linear SVM classifier achieved the best F-measure results, outperforming the optimal threshold classifier. Of the two-step approaches, the threshold based approach seems to be very sensitive to the chosen threshold value, while with the nearest-based approach, imbalanced training set size outperforms balanced training set size in most cases, often achieving significantly better results than k-means clustering. For the balanced nearest-based approach, there seems to be a general trend that smaller training set size results in better classi-

Classification approach	Data sets					
	Census	Restaurant	DS-Gen 1,000	DS-Gen 2,500	DS-Gen 5,000	DS-Gen 10,000
Optimal threshold	0.784	0.839	0.900	0.846	0.813	0.787
SVM	0.785	0.466	0.944	0.917	0.884	0.829
K-means clustering	0.434	0.002	0.802	0.814	0.763	0.213
Threshold-0.1	0.000	0.000	0.000	0.844	0.808	0.750
Threshold-0.3	0.000	0.000	0.857	0.809	0.735	0.655
Threshold-0.5	0.187	0.001	0.711	0.609	0.527	0.751
Threshold-0.7	0.171	0.704	0.826	0.816	0.744	0.492
Threshold-0.9	0.149	0.379	0.779	0.681	0.688	0.458
Nearest-1%, NU, B	0.566	0.001	0.854	0.821	0.728	0.500
Nearest-5%, NU, B	0.643	0.001	0.865	0.815	0.573	0.199
Nearest-10%, NU, B	0.317	0.001	0.840	0.757	0.344	0.080
Nearest-1%, U, B	0.566	0.002	0.863	0.834	0.741	0.515
Nearest-5%, U, B	0.500	0.002	0.861	0.813	0.582	0.203
Nearest-10%, U, B	0.271	0.002	0.841	0.757	0.348	0.087
Nearest-1%, NU, IB	0.567	0.044	0.865	0.815	0.780	0.738
Nearest-5%, NU, IB	0.644	0.012	0.851	0.823	0.805	0.751
Nearest-10%, NU, IB	0.410	0.002	0.830	0.817	0.797	0.739
Nearest-1%, U, IB	0.568	0.008	0.858	0.806	0.781	0.757
Nearest-5%, U, IB	0.511	0.005	0.849	0.822	0.807	0.756
Nearest-10%, U, IB	0.388	0.003	0.832	0.815	0.799	0.747

Table 5: F-measure classification results. 'U' and 'NU' denote the unique and non-unique weight vector selection, respectively, and 'B' and 'IB' the balanced and imbalanced training set size selection. Note that 'Optimal threshold' and 'SVM' are supervised classification techniques, while all other approaches are unsupervised.

fication quality compared to larger training sets. No such trend is visible for the imbalanced nearest-based approach. There is also no clear advantage or disadvantage for using unique or non-unique nearest-based selection of training examples.

These initial results indicate that the proposed two-step approach to automatic record pair classification is feasible and can achieve linkage quality almost as good a fully supervised classification. Specifically, the nearest-based selection of match and non-match training example sets can automatically generate training data of high quality. However, more investigation is needed for the second step of the proposed approach; on how to best use the generated training example sets for classification. More experiments using different data sets and additional classifiers have to be conducted in order to validate the general applicability of the proposed approach to a wide range of data with different characteristics.

5 Conclusions and Future Work

In this paper, a novel two-step approach to record pair classification has been presented that aims to automate the record linkage process. This approach combines an automatic selection of training examples, that with high likelihood are either true matches or true non-matches, with a traditional supervised classifier. Initial experiments on a range of data sets showed promising results, in certain cases achieving F-measure values almost as good as a fully supervised linear SVM classifier, but generally better than an unsupervised k-means clustering approach.

There are two main extensions to the basic approach presented here that will be investigated in the near future. First, rather than only using a classifier once in the second step to classify all weight vectors in $\mathbf{W}_T$, an iterative approach, similar to PEBL (Yu et al. 2002), will be explored. The basic idea is that in each iteration the strongest classified matches and non-matches in $\mathbf{W}_T$, i.e. the weight vectors furthest away from the decision boundary, will be added to the training sets $\mathbf{W}_M$ and $\mathbf{W}_N$. This process is repeated until a certain stopping is fulfilled.

Second, similar to the inclusion of 'spy' documents in the S-EM approach to partially supervised text classification (Liu et al. 2002), adding randomly sampled weight vectors from the 'gap' between the selected matches and non-matches into the training example sets should improve the overall classification accuracy. Random sampling should be conducted such that weight vectors closer to the exact match vector $\mathbf{m}$ are more likely included into the set of match training examples, $\mathbf{W}_M$, while weight vectors that contain mainly dissimilarity values should more likely be included into the set of non-match training examples, $\mathbf{W}_N$. This idea is currently being implemented and results will be reported elsewhere.

Additionally, the effects of using different approximate string comparison techniques (Christen 2006) on the proposed approach will also be investigated.

Other future work will include a scalability and complexity analysis, as well as timing measurements, of the proposed approach. Given that only a fraction of all weight vectors are selected into the two training sets in step one, the time required to train a classifier in the second step of the proposed approach should be small compared to using all weight vectors for supervised classification or clustering.

6 Acknowledgements

This work is supported by an Australian Research Council (ARC) Linkage Grant LP0453463 and partially funded by the New South Wales Department of Health. The author would like to thank Paul Thomas for proof-reading this paper.

References

Basu, S., Banerjee, A. & Mooney, R.J. (2002), Semi-supervised clustering by seeding, *in* 'International Conference on Machine Learning' (ICML'02), Sydney, Australia, pp. 19–26.

Baxter, R., Christen, P. & Churches, T. (2003), A comparison of fast blocking methods for record linkage, *in* 'ACM SIGKDD Workshop on Data Cleaning, Record Linkage and Object Consolidation', Washington DC, pp. 25–27.

Bhattacharya, I. & Getoor, L. (2007), 'Collective entity resolution in relational data', *ACM Transac-*

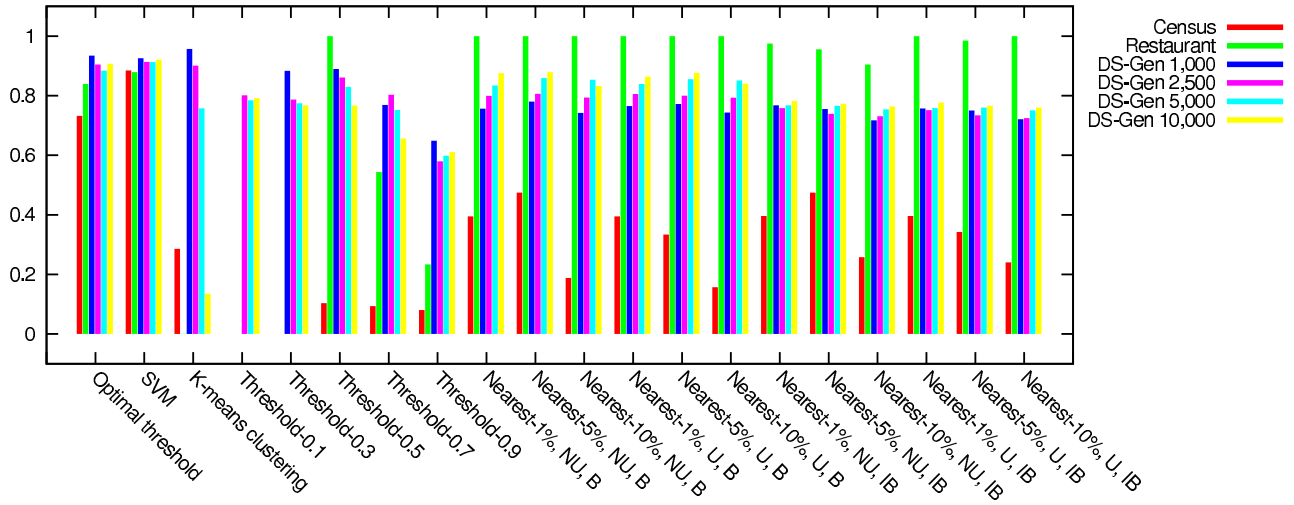


Figure 3: Precision results for all data sets and classification methods.

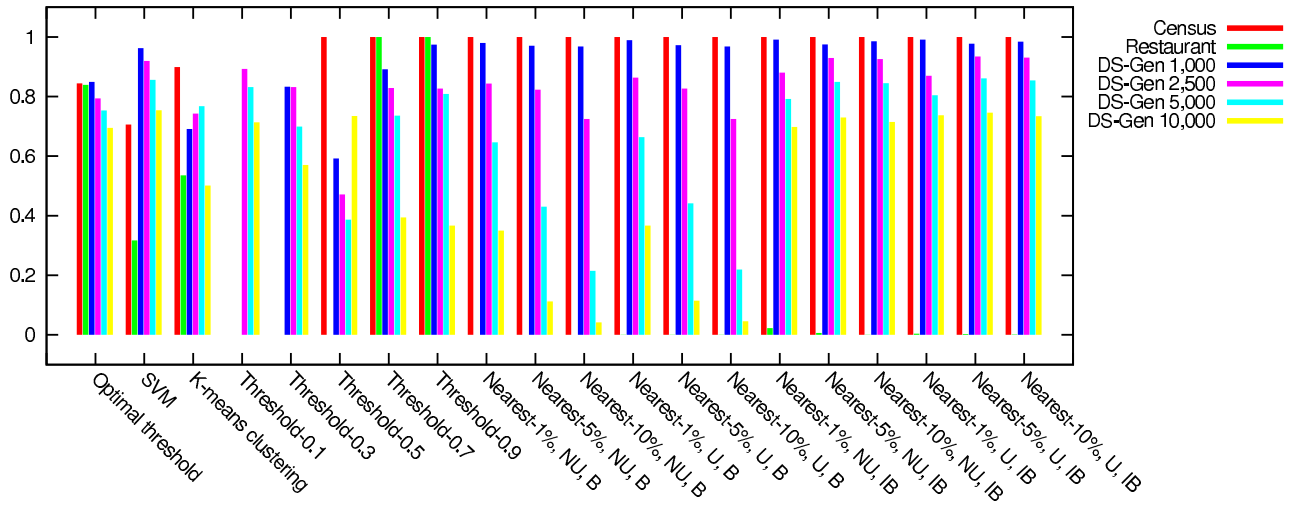


Figure 4: Recall results for all data sets and classification methods.

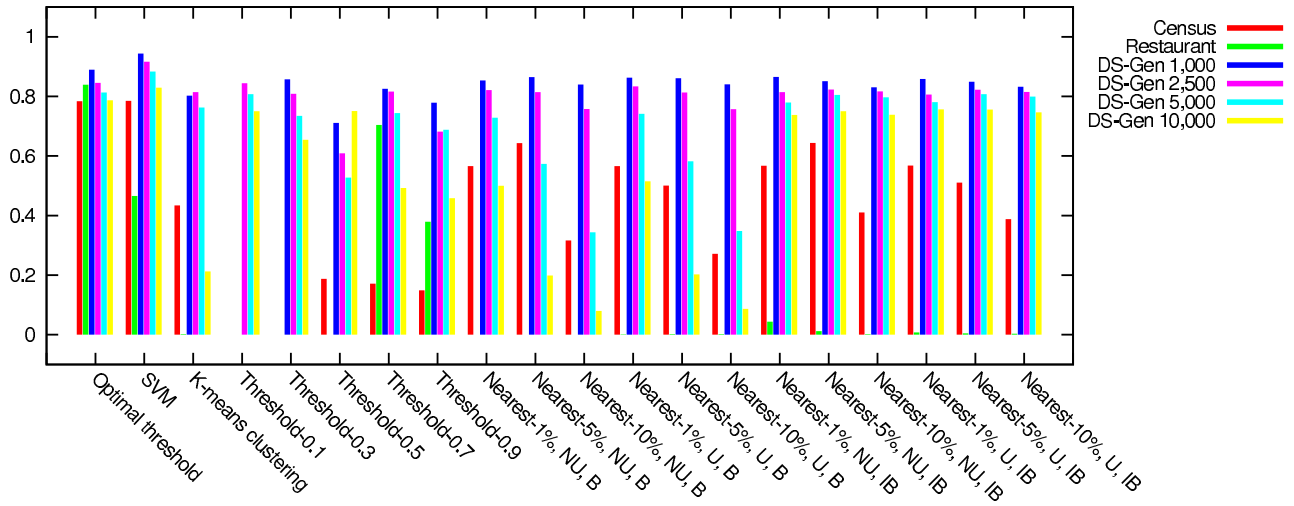


Figure 5: F-measure results for all data sets and classification methods.

tions on Knowledge Discovery from Data (TKDD), vol. 1, no. 1.

Bilenko, M. & Mooney, R.J. (2003), Adaptive duplicate detection using learnable string similarity measures, *in* 'ACM International Conference on Knowledge Discovery and Data Mining' (SIGKDD'03), Washington DC, pp. 39–48.

Bilenko, M., Basu, S. & Sahami, M. (2005), Adaptive product normalization: Using online learning for record linkage in comparison shopping, *in*

'IEEE International Conference on Data Mining' (ICDM'05), Houston, Texas, pp. 58–65.

Chang, C.-C. & Lin, C.-J. (2001), LIBSVM: a library for support vector machines, manual. Department of Computer Science, National Taiwan University. Software available at: `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Christen, P., Churches, T. & Hegland, M. (2004), Febrl – A parallel open source data linkage system, *in* 'Pacific-Asia Conference on Knowledge Discovery and Data Mining' (PAKDD'04), Sydney, Springer LNAI 3056, pp. 638–647.

Christen, P. (2005), Probabilistic data generation for deduplication and data linkage, *in* 'International Conference on Intelligent Data Engineering and Automated Learning' (IDEAL'05), Brisbane, Springer LNCS 3578, pp. 109–116.

Christen, P. (2006), A comparison of personal name matching: techniques and practical issues, *in* 'Workshop on Mining Complex Data' (MCD), held at IEEE ICDM'06, Hong Kong.

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.

Churches, T., Christen, P., Lim, K. & Zhu, J.X. (2002), 'Preparation of name and address data for record linkage using hidden Markov models', *BioMed Central Medical Informatics and Decision Making*, vol. 2, no. 9.

Clarke, D.E. (2004), 'Practical introduction to record linkage for injury research', *Injury Prevention*, vol. 10, pp. 186–191.

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.

Cohen W.W., Ravikumar P. & Fienberg S.E. (2003), A comparison of string distance metrics for name-matching task, *in* 'IJCAI-03 workshop on information integration on the Web' (IIWeb-03), Acapulco, pp. 73–78.

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.

Fellegi, I.P. & Sunter, A.B. (1969), 'A theory for record linkage', *Journal of the American Statistical Society*, vol. 64, no. 328, pp. 1183–1210.

Gill, L. (2001), 'Methods for automatic record matching and linking and their use in national statistics', *National Statistics Methodology Series*, no. 25, National Statistics, London.

Goiser K. & Christen, P. (2006), Towards automated record linkage, *in* 'Australasian Data Mining Conference' (AusDM'06), Sydney, Conferences in Research and Practice in Information Technology (CRPIT), vol. 61, pp. 23–31.

Gu, L. & Baxter, R. (2006), Decision models for record linkage, *in* 'Selected Papers from AusDM', Springer LNCS 3755, pp. 146–160.

Kelman, C.W., Bass, J. & Holman, D. (2002), 'Research use of linked health data – A best practice protocol', *Aust NZ Journal of Public Health*, vol. 26, pp. 251–255.

Liu, B., Lee, W.S., Yu, P.S. & Li, X. (2002), Partially supervised classification of text documents, *in* 'International Conference on Machine Learning' (ICML'02), Sydney, Australia, pp. 387–394.

Liu, B., Dai, Y., Li, X., Lee, W.S. & Yu, P.S. (2003), Building text classifiers using positive and unlabeled examples, *in* 'IEEE International Conference on Data Mining' (ICDM'03), Melbourne, Florida, pp. 179–186.

McCallum, A., Nigam, K. & Ungar, L.H. (2000), Efficient clustering of high-dimensional data sets with application to reference matching, *in* 'ACM International Conference on Knowledge Discovery and Data Mining' (SIGKDD'00), Boston, pp. 169–178.

Nahm, U.Y., Bilenko, M. & Mooney, R.J. (2002), Two approaches to handling noisy variation in text mining, *in* 'ICML'02 workshop on text learning' (TextML'02), Sydney, Australia, pp. 18–27.

Neiling, M. (2005), Identification of real-world objects in multiple databases, *in* '29th Annual Conference of the Gesellschaft für Klassifikation e.V.', University of Magdeburg, pp. 63–74.

Nigam, K., McCallum, A.K., Thrun, S. & Mitchell, T. (2000), 'Text classification from labeled and unlabeled documents using EM', *Machine Learning*, vol. 39, no. 2, pp. 103–134.

Rahm, E. & Do, H.H. (2000), 'Data cleaning: Problems and current approaches', *IEEE Data Engineering Bulletin*, vol. 23, no. 4, pp. 3–13.

Sarawagi S., & Bhamidipaty A. (2002), Interactive deduplication using active learning, *in* 'ACM International Conference on Knowledge Discovery and Data Mining' (SIGKDD'02), Edmonton, Canada, pp. 269–278.

Tejada S., Knoblock C.A. & Minton S. (2000), Learning domain-independent string transformation weights for high accuracy object identification, *in* 'ACM International Conference on Knowledge Discovery and Data Mining' (SIGKDD'02), Edmonton, Canada, pp. 350–359.

Wang, G., Chen, H., Xu, J.J. & Atabakhsh, H. (2006), 'Automatically detecting criminal identity deception: An adaptive detection algorithm', *IEEE Transactions on Systems, Man and Cybernetics (Part A)*, vol. 36, no. 5, pp. 988–999.

Winkler, W.E. (2000), 'Using the EM algorithm for weight computation in the Fellegi-Sunter model of record linkage', *Technical report RR2000/05*, US Bureau of the Census.

Winkler, W.E. (2004), 'Methods for evaluating and creating data quality', *Information Systems*, Elsevier, vol. 29, no. 7, pp. 531–550.

Winkler, W.E. (2006), 'Overview of record linkage and current research directions', *Technical report RR2006/02*, US Bureau of the Census.

Yu, H., Han. J. & Chang, K.C.C. (2002), PEBL: Positive example based learning for Web page classification using SVM, *in* 'ACM International Conference on Knowledge Discovery and Data Mining' (SIGKDD'02), Edmonton, Canada, pp. 239–248.

Discovering Frequent Sets from Data Streams with CPU Constraint

Xuan Hong Dang, Wee-Keong Ng[1] Kok-Leong Ong[2] Vincent C S Lee[3]

[1] School of Computer Engineering
Nanyang Technological University, Singapore
Email: {dang0008,awkng}@ntu.edu.sg

[2] School of Engineering & IT
Deakin University, Australia
Email: leong@deakin.edu.au

[3] Clayton School of Information Technology
Monash University, Australia
Email: Vincent.Lee@infotech.monash.edu.au

Abstract

Data streams are usually generated in an online fashion characterized by huge volume, rapid unpredictable rates, and fast changing data characteristics. It has been hence recognized that mining over streaming data requires the problem of limited computational resources to be adequately addressed. Since the arrival rate of data streams can significantly increase and exceed the CPU capacity, the machinery must adapt to this change to guarantee the timeliness of the results. We present an online algorithm to approximate a set of frequent patterns from a sliding window over the underlying data stream – given *apriori* CPU capacity. The algorithm automatically detects overload situations and can adaptively shed unprocessed data to guarantee the timely results. We theoretically prove, using probabilistic and deterministic techniques, that the error on the output results is bounded within a pre-specified threshold. The empirical results on various datasets also confirmed the feasiblity of our proposal.

Keywords: Data Stream; Frequent Set Mining; Online Algorithm; Load Shedding; Error Approximation

1 Introduction

Data streams have recently become a novel data type that attracted much attention from the research community, ranging from works in data stream querying (Garofalakis et al. 2002), clustering (Guha et al. 2003), classification (Hulten et al. 2001) to frequent sets mining (Manku et al. 2002). They arise naturally in a number of applications, including financial analysis & stock trading, fraud detection, and IP & sensor network (Babcock et al. 2002, Garofalakis et al. 2002), which exhibit properties of online, huge volume, high arrival rates, and fast changing behaviors. In this novel setting, conventional methods that deal with persistently stored datasets become ineffective in streaming environments. It is therefore imperative to design new techniques that have the ability to compute the answer in an online fashion with only one scan on the streaming data whilst operating under the resource limitations (e.g., CPU cycles (Tatbul et al. 2003), memory space (Jeffrey et al. 2004), and bandwidth communication (Chi et al. 2005)).

For many stream applications such as stock monitors, telecom fraud detection or sensor network surveillance , the data arrival rate is rapid and mostly unpredictable (due to the dynamic changing behavior of data sources) (Das et al. 2003). Meanwhile, analyzing results of these streams usually requires delivery under time constraints to enable optimal decisions. Unfortunately, when the rate of the data stream significantly increases and exceeds the system capacity, overloading situations happen and the system may not be able to deliver the results within the given timeframe. Consequently, the quality of service can be degraded without bound. To cope with such situations, one may add more resources to handle the increased load or distribute the computation to multiple nodes. However, such an approach is expensive and generally infeasible in practice (Tatbul et al. 2003). Therefore, it has been recognized that dropping unprocessed data (thus, only approximate results can be obtained) to reduce workload and maintain the timeliness of output results is generally preferred.

In the literature, the problem of dropping some data in order to cope with high speed data streams is often known as *load shedding*. Currently, this problem has been intensively studied in data stream querying (Babcock et al. 2004, Gedik et al. 2005, Tatbul et al. 2003, 2006, Tu et al. 2006). However, it has not been well-addressed in the context of data stream mining (Chi et al. 2005), especially for the problem of finding frequent sets from transactional data streams. For this fundamental mining task, most algorithms reported so far focused on the issue of managing limited memory space (Lin et al. 2005, Giannella et al. 2003, Chang et al. 2003*a*, Teng et al. 2003, Jeffrey et al. 2004). They ignore the fact that CPU is also a bounded resource. It is important to note that the main difficulty in frequent set mining is the large number of itemsets whose frequencies need to be tracked. Given a transaction of size m, the number of itemsets is exponentially proportional to its size; i.e., for a transaction of size $m = 10$, the number of itemsets may be up to $2^{10} - 1$. This clearly makes the frequent set mining problem susceptible to overloading.

In this paper, we present an approximate algorithm to perform online computation of a set of frequent patterns from a transactional data stream with CPU as a bounded resource. We consider the most general mining model: the *time-based sliding window* model. It is important to note that, although frequent sets are computed in the sliding data window, no revisiting of expiring transactions is needed in our approach. Incoming transaction are processed online

and they will be adaptively dropped when overload situations are detected in order to guarantee the timeliness of the mining results. Furthermore, we prove that the error on the mining results is theoretically kept within a preset threshold by exploiting the Chernoff bound property (Chernoff 1952) and the deterministic technique (Manku et al. 2002). To verify the feasibility of our algorithm, experiments on various synthetic datasets have been performed where both the speed and the characteristics of data streams are changed from time to time.

The rest of the paper is organized as follows. In the following section, we formally formulate the problem of mining frequent sets from a window sliding on data streams given the limitation on CPU capacity. Section 3 describes our approach. The load detection and load shedding technique are addressed first, then the algorithm is represented and followed by the error analysis. Section 4 reports our experimental results. Related work is presented in Section 5. Finally, our conclusions are presented in Section 6.

2 Problem Definition

We consider a high speed data stream arriving as a time ordered series of transactions $\mathcal{DS} = \{t_1, t_2, ..., t_n, ...\}$. Each transaction t_i contains a set of items such that $t_i \subseteq I$, where $I = \{a_1, a_2, ..., a_m\}$ is a set of literals called items (or objects) and t_n is called the current transaction arriving on the stream.

We focus on *time-based* sliding window. In this model, let $TS_0, TS_1, \ldots, TS_{i-k+1}, \ \ldots, TS_i$ denote the time periods elapsed so far in the stream. Each time period (or time slot) contains multiple transactions arriving in that interval and thus, they form a partition of transactions in the stream. Given an integer k, the time-based sliding window is defined as the set of transactions arriving in the last k time periods and denoted by $\mathcal{SW} = \{TS_{i-k+1}, \ldots, TS_{i-1}, TS_i\}$. TS_i is called the latest time slot and TS_{i-k} is called the expiring one in the sliding window. When the time shifts to the new slot TS_i, the effect of all transactions in TS_{i-k} will be eliminated from the mining model.

Let n_w be the number of transactions arriving between TS_{i-k+1} and TS_i, the frequency of an itemset X, denoted by $freq(X)$, is defined as the number of transactions between TS_{i-k+1} and TS_i that contain X. Its support, denoted by $supp(X)$, is defined as the ratio between $freq(X)$ and n_w; i.e., $supp(X) = freq(X) \times n_w^{-1}$. Given $\sigma \in (0, 1]$ as a threshold for minimum support, X is called a frequent itemset if $supp(X) \geqslant \sigma$; it is called a maximal frequent itemset (MFI) if none of its immediate supersets are frequent.

Our problem is defined as follows: we are given a processing capacity (CPU) C of the mining system and a data stream $\mathcal{DS}$ with arbitrarily high arrival rates (the characteristics of $\mathcal{DS}$ also may change with time). Let $Load(\mathcal{DS})$ indicate the workload of the system, then a load shedding is invoked whenever $Load(\mathcal{DS}) > C$. The objective is to find all frequent patterns together with their estimated frequencies in the sliding window while guaranteeing that $Load(\mathcal{DS}) \leqslant C$.

3 Approximate Frequent Sets under bounded CPU

3.1 Workload Estimation

Since the behavior of data streams often changes over time, detecting overload situations is an important step in our algorithm. For the frequent set mining problem, the system workload may not be simply estimated by regularly checking the rate of transactions arriving in one time unit. Rather, it is essentially dependent on the number of itemsets containing in each transaction whose frequencies must be updated. Certainly, an accurate method to evaluate the system workload is to have an exact count of this number in each transaction. Unfortunately, this task is generally impossible since the system may not be able to fully process all incoming transactions under overload situations. Therefore, it is necessary to find a technique that is able to approximately estimate this number meanwhile it is efficient to compute. We propose such an estimate based on a small set of maximal frequent itemsets (MFIs). The key intuition behind using MFIs for this task is that the number of MFIs is exponentially smaller than the number of frequent itemsets, as shown by (Guizhen Yang 2004). Meanwhile, such a compact set completely captures the entire set of frequent itemsets. To give an explicit formula for this estimate, let us denote the number of MFIs in a transaction by m and let X_i be a MFI, $1 \leqslant i \leqslant m$. We assume the following estimated time (or load coefficient) to process one transaction:

$$L = \sum_{i=1}^{m} 2^{|X_i|} - \sum_{i,j=1}^{m} 2^{|X_i \cap X_j|} \qquad (1)$$

In this equation, the first summation estimates the number of frequent itemsets within each MFI. The second one estimates the number of itemsets overlapping between MFIs. Apparently, one may suppose that computing L is expensive and eventually defeats the purpose of quickly detecting CPU overload. Nevertheless, we do not explicitly compute L by finding all MFIs and the overlapping subsets among them. Instead, the set of MFIs is maintained in a prefix tree and the equation is computed by matching transactions to this structure to derive the number of distinct frequent sets. As we shall see from the empirical results in Section 4, this approach yields very good approximation while the computational time is negligible.

Given this computed statistics for each transaction, we measure it for n transactions over one time unit and let r be the current rate of the data stream. The following inequality imposes a constraint on load shedding decisions:

$$P \times r \times \frac{\sum_{i=1}^{n} L_i}{n} \leqslant C \qquad (2)$$

In this inequality, $P \in (0, 1]$ is a parameter adjusted adaptively to make the inequality hold. It also expresses the fraction of transactions that should be discarded. The expression $r \times \frac{\sum_{i=1}^{n} L_i}{n}$ gives the estimated system workload to process transactions arriving in one time unit. L_i is described in Equation 1; C, as formulated above, is the processing capacity of the mining system.

3.2 Probabilistic Technique to Shed Load

As we have analyzed above, when the system is overloaded, an immediate approach is to drop a fraction of the stream to reduce workload. Certainly, when all incoming data is not entirely processed (and dropped transactions are lost forever), one can expect some errors in the results. Our algorithm is designed to approximate this error in a precise manner. In other words, the error is guaranteed within some specific value. To achieve this, we approach the problem by utilizing a technique from probability, the Chernoff bound (Chernoff 1952). Such a theoretically sound

tool allows us to obtain a more accurate estimate on the mining error. To apply the Chernoff bound in our frequent set approximation, we clarify the following concepts. Let P be a value smaller than 1, then each coming transaction is chosen with probability P. For a set of N transactions arriving in the stream, n transactions are chosen randomly. Given an itemset X, we want to approximate how close is its computed frequency in n sampling transactions, as compared to its actual frequency p in N transactions.

Note that event X appearing in a transaction can be seen as a Bernoulli trial and thus, we denote random variable $A_i = 1$ if X appears in the ith transaction and $A_i = 0$ otherwise. Obviously, $\Pr(A_i = 1) = p$ and $\Pr(A_i = 0) = 1 - p$.[1] Hence, n randomly drawn transactions are viewed as n independent Bernoulli trials. Let r be the number of times that $A_i = 1$ occurs in these n transactions; r is called a *binomial* random variable and thus, its expectation is np. Then, the Chernoff bound states that given an error bound ϵ, $0 < \epsilon < 1$:

$$\Pr\{|r - np| \geqslant np\epsilon\} \leqslant 2e^{-np\epsilon^2/2} \quad (3)$$

Let us call $supp_E(X) = r/n$ the estimated support of X computed from n sampling transactions, then this equation gives us the probability that the true support $supp_T(X)$ deviates from its estimated support $supp_E(X)$ by an amount $\pm\epsilon$. If we want this probability to be no more than δ, then the required number of sampling transactions is at least (by setting $\delta = 2e^{-np\epsilon^2/2}$)

$$n_0 = \frac{2p\ln(2/\delta)}{\epsilon^2} \quad (4)$$

For example, consider $p = 0.002$, $\delta = 0.05$ and $\epsilon = 0.001$, then $n_0 \approx 15,000$. This means that for itemset X, if we sample 15,000 transactions from a partition of the stream, its true support $supp_T(X)$ in this partition is beyond the range of $[supp_E(X) - 0.001, supp_E(X) + 0.001]$ with probability 0.05. In other words, $supp_T(X)$ is within $\pm\epsilon$ of $supp_E(X)$ with a high confidence of 95%.

3.3 The Algorithm

In our framework, frequent patterns are discovered from a time-based sliding window where the window of interest is the set of transactions arriving in the last k time slots. We further divide the data stream into *con*ceptual win*dows* (condows) having the same size of Δ transactions and assume that each time slot consists of multiple condows. Accordingly, these condows form the basic units of each time slot (rather than individual transactions) in our model. The value of Δ is chosen to be equal to n_0 as computed in Equation 4.

When the window slides to a new time slot TS_i, the effect of all condows in the expired one, i.e., TS_{i-k} will be eliminated and thus, the current window now consists of condows arriving only in time slots $\mathcal{SW} = \{TS_{i-k+1}, ..., TS_i\}$. It is also important to note that since the rate of the data stream can change with time, the number of condows appearing in each time slot may not be the same.

In essence, not all itemsets appearing in the stream should be recorded and counted due to system resource limitations. By conceptually dividing the stream into condows, we can employ a deterministic threshold to filter itemsets whose frequencies are insignificant. In our algorithm, the threshold $\gamma(< \sigma)$ is used for this task. If a pattern has a support no more than γ, it is unlikely to be frequent in the near future and therefore, should be removed early from the results. Furthermore, our goal is not only to produce an approximate set of frequent patterns but also wants to deliver it within a precise error limit. This guarantee is a challenging task since our algorithm is designed to scan transactions online to find frequent sets; a potential candidate is generated only after all its subsets are found to be significantly frequent. In other words, longer patterns will suffer from a larger margin of error and as a result, it is not able to guarantee the same error for every pattern of different lengths. However, when partitioning the data stream into equal-sized condows, we can approximate the patterns' error based on their lengths and more importantly, it is able to tighten their upper error bound and the condition to generate potential candidates.

In our design, an array storing minimum frequency thresholds is utilized for this purpose. If m is the maximal size of frequent sets that the user wants to explore from the stream, then the only condition this array needs to satisfy is: $a[i] < a[j]$ for $1 \leqslant i < j \leqslant m$ and $a[m] \leqslant \gamma \times \Delta$. Therefore, an itemset of size j will be generated if its immediate subset frequencies in the current condow are above the threshold specified in $a[j]$; i.e., its subsets are significant in the condow. On the other hand, by choosing the size of each condow to be equal to Δ transactions, we can approximate

[1] We use Pr(.) to denote the probability of a condition being met.

Algorithm 1. Find frequent sets from a time-based sliding window.

Input:
(1) A processing capacity (CPU) C of the mining system;
(2) A window sized of k time slots sliding on data stream $\mathcal{DS}$;
(3) A minimum support threshold $\sigma \in (0, 1]$;
(4) A significant threshold $\gamma \in (0, \sigma)$ and condow size Δ;
Output: At anytime on demand, return all estimated frequent sets computed in the sliding window

```
1: TS = 0; w_o = 0; w_c = 1; W[mod(TS,k)] = w_c;
2: Periodically identify sampling rate P;
3: for each t_n ∈ DS, sampling it with probability of P and if
   t_n is chosen do
4:    for all X ⊆ t_n and X ∈ S do Ccnt(X)++
5:    for all X ⊆ t_n and X ∉ S do
6:       if X is 1-itemset then
7:          Insert X to S with X.Wid = w_c; Ccnt(X) = 1
8:       else
9:          if (for all Y ⊂ X, ∄Y ∈ S such that
              (Y.Wid < w_c and Ccnt(Y) ≤ arr[|X|]) or
              (Y.Wid = w_c and Ccnt(Y) ≤ arr[|X|] − arr[|Y|]))
            then
10:             Insert X to S with X.Wid = w_c; Ccnt(X) = 1;
11:         end if
12:      end if
13:      if X cannot be inserted into S then
14:         t_n = t_n − {Z ⊆ t_n | Z is a superset of X}
15:      end if
16:   end for
17:   for every Δ transactions sampled, scan S and do
18:      X.Freq[w_c] = Ccnt(X) × P^{-1}
19:      S = S − {X | X.Wid < w_o and Σ X.Freq[i] ⩽
         (w_c − w_o + 1) × a[|X|]}
20:      S = S − {X | X.Wid ⩾ w_o and Σ X.Freq[i] ⩽
         (w_c − X.Wid + 1) × a[|X|]}
21:      w_c = ⌈n/Δ⌉ + 1
22:   end for
23:   if A new time slot arrives and TS ⩾ k then
24:      TS++; w_o = W[mod(TS,k)]; W[mod(TS,k)] = w_c
25:      Remove X.Freq[i] such that i ⩽ w_o
26:   end if
27: end for
28: if mining results are requested then
29:   return {X | X.Wid < w_o and Σ X.Freq[i] ⩾ (w_c −
      w_o + 1) × Δ × σ}
30:   return {X | X.Wid ⩾ w_o and Σ X.Freq[i]+(X.Wid−
      w_o + 1) × a[|X|] ⩾ (w_c − w_o + 1) × Δ × σ}
31: end if
```

(in terms of probability) the true support of patterns within each time slot to be no more than ϵ when load shedding is involved. As will be shown in Section 3.4, the error on frequent sets is guaranteed to be within the preset error thresholds γ and ϵ.

Given the above analysis, we describe our algorithm as follows. We name the algorithm *L*oad *S*hedding for mining frequent sets from *S*liding *W*indows (LSSW) and its pseudo code is outlined in Algorithm 1. LSSW uses a prefix tree $\mathcal{S}$ to maintain frequent sets discovered from the sliding window. Each node in $\mathcal{S}$ corresponds to an itemset X having the following fields:

- *Item*: The last item of itemset X. Thus, X is represented by the set of items on the path from the root to the node.
- Wid: The index of the condow at which X is inserted into $\mathcal{S}$.
- $Freq[1..max]$: A circular queue storing frequencies counted at each condow in the sliding window.

We denote the latest (or current) condow in the sliding window by w_c, and the oldest with w_o. To efficiently eliminate expired time periods, the indexes of the first condows in each time slot are tracked globally and stored in a circular array W of k elements corresponding to the last k time slots. Accordingly, the value of w_o is always cleared each time the window slides. The time slot index TS is initialized to 0 when the algorithm starts and is incremented at each new time period. Let t_n be the new arriving transaction. LSSW consists of the following steps:

1. The system workload is periodically estimated to identify overloading. If such a situation occurs, the maximal value of P is identified via Equation 2. Otherwise, P is set to 1.
2. For each incoming transaction t_n, it is sampled with probability P. If t_n is chosen, the following steps are performed:
   - Increment: If an itemset X appearing in t_n is also currently maintained in $\mathcal{S}$, then its frequency in the current condow, denoted by $Ccnt(X)$, is increased by 1.
   - Insert: For each $X \subseteq t_n$ and $X \notin \mathcal{S}$, insert X into $\mathcal{S}$ with $X.Wid = w_c$ and $Ccnt(X) = 1$ if X is a singleton [2]; Otherwise, let Y be any immediate subset of X, then X is inserted into $\mathcal{S}$ if the following three conditions hold:
     - All immediate subsets of X are in $\mathcal{S}$;
     - $\nexists Y$ such that $Y.Wid < w_c$ and $Ccnt(Y) \leqslant arr(|X|)$; i.e., there is no Y being inserted into $\mathcal{S}$ from previous condows whose frequency in the current condow is insufficiently significant;
     - $\nexists Y$ such that $Y.Wid = w_c$ and $Ccnt(Y) \leqslant (arr(|X|) - arr(|Y|))$; i.e., there is no Y that has just been inserted into $\mathcal{S}$ in the current condow after which its frequency is no more than $(arr(|X|) - arr(|Y|))$.

     In cases where X is not inserted into $\mathcal{S}$, all its supersets in t_n need not be further checked.
   - Update frequency: After each Δ transactions have been sampled, the algorithm updates $Freq[w_c] = Ccnt(X) \times P^{-1}$. Note that to compensate for dropping transactions using sampling rate P, X's frequency is scaled up appropriately by P^{-1} to approximate its true frequency in the sampling part of the stream. At this step, LSSW also scans $\mathcal{S}$ to remove all but 1-itemsets that satisfy either of the two conditions:
     - If $X.Wid < w_o$ and $\sum X.Freq[i] \leqslant (w_c - w_o + 1) \times a[|X|]$
     - If $X.Wid \geqslant w_o$ and $\sum X.Freq[i] \leqslant (w_c - X.Wid + 1) \times a[|X|]$

     Certainly if an itemset is removed, all its supersets are also removed. Those itemsets recently inserted into $\mathcal{S}$ in the current condow will not be removed since they are generated after their immediate subsets became sufficiently frequent. After that, LSSW updates the index for the next condow by $w_c = \lceil n/\Delta \rceil + 1$.
   - Remove expiring time slot: In case the window slides to a new time slot, TS will be incremented by 1 and if $TS \geqslant k$, the algorithm further removes frequency counts in queue $X.Freq$ where indexes are smaller than value w_o stored in $W[mod(TS, k)]$. Then LSSW updates $W[mod(TS, k)] = w_c$ to register the first condow index of the new time slot.
3. At any instant upon the user's request, the algorithm scans $\mathcal{S}$ to output all itemsets satisfying either of the conditions below:
   - If $X.Wid < w_o$ and $\sum X.Freq[i] \geqslant (w_c - w_o + 1) \times \Delta \times \sigma$
   - If $X.Wid \geqslant w_o$ and $\sum X.Freq[i] + (X.Wid - w_o + 1) \times a[|X|] \geqslant (w_c - w_o + 1) \times \Delta \times \sigma$

   For the second condition, it is noted that $(X.Wid - w_o + 1) \times a[|X|]$ is X's maximal frequency error from w_o to $X.Wid$ (as will be proven in the following section).

3.4 Error Analysis

We prove the correctness of our algorithm in this section. For simplicity, we shall omit the notation X and re-denote its *true* frequency between two condows w_α and w_β (where $\alpha \leqslant \beta$) by $f_T(w_\alpha, w_\beta)$ and its *estimated* one by $f_E(w_\alpha, w_\beta) = \sum_{i=\alpha}^{\beta} X.Freq[i]$. Respectively, $s_T(w_\alpha, w_\beta)$ and $s_E(w_\alpha, w_\beta)$ denote its *true* and *estimated* supports in this period. In the proof of Lemmas 1 and 2, we ignore 1-itemsets and itemsets those have $X.Wid < w_o$ since their frequencies have been precisely counted in the sliding window.

Lemma 1 *Under no load shedding, if an itemset X is deleted at condow w_d, then*

1. *$f_T(w_s, w_d) \leqslant (w_d - w_s + 1) \times a[|X|]$; i.e., the true frequency of X between w_d and w_s – any condow at which X was previously inserted – is no more than $(w_d - w_s + 1) \times a[|X|]$; and*
2. *$f_T(w_d^p + 1, w_d) \leqslant (w_d - w_d^p) \times a[|X|]$, where w_d^p is any condow (previous w_d) at which X was also deleted.*

[2] Note that the immediate subsets of a 1-itemset is $\{\varnothing\}$ which appears in every transaction, all 1-itemsets are therefore inserted into $\mathcal{S}$ without conditions. For the same reason, they are also not pruned from $\mathcal{S}$.

Proof. Let us denote condow indexes of the two latest periods at which X is inserted and deleted by $w_s^{\ell-1}$, $w_d^{\ell-1}$ and w_s^{ℓ}, w_d^{ℓ} respectively.

When X is inserted at w_s^{ℓ}, its maximum error at this condow is $a[|X|]$, and $f_E(w_s^{\ell}, w_d^{\ell})$ is its frequency count since then to w_d^{ℓ}. Therefore $f_T(w_s^{\ell}, w_d^{\ell}) \leqslant f_E(w_s^{\ell}, w_d^{\ell}) + a[|X|]$. Together with the deletion rule at w_d^{ℓ}, we have $f_T(w_s^{\ell}, w_d^{\ell}) \leqslant (w_d^{\ell} - w_s^{\ell} + 1) \times a[|X|]$. On the other hand, X is not inserted anytime between $w_d^{\ell-1}+1$ and $w_s^{\ell}-1$, its true frequency in each of these condows is no more than $a[|X|]$. This means that $f_T(w_d^{\ell-1}+1, w_s^{\ell}-1) \leqslant (w_s^{\ell}-1-w_d^{\ell-1}) \times a[|X|]$. Summing up these 2 periods, $f_T(w_d^{\ell-1}, w_d^{\ell}) \leqslant (w_d^{\ell} - w_d^{\ell-1}) \times a[|X|]$ is derived.

Similar to above when X was inserted at $w_s^{\ell-1}$ and deleted at $w_d^{\ell-1}$, we have $f_T(w_s^{\ell-1}, w_d^{\ell-1}) \leqslant (w_d^{\ell-1} - w_s^{\ell-1} + 1) \times a[|X|]$. Together with the above result, $f_T(w_s^{\ell-1}, w_d^{\ell}) \leqslant (w_d^{\ell} - w_s^{\ell-1} + 1) \times a[|X|]$ is satisfied. By repeating the same calculation for every previous insertion and deletion periods, the lemma follows. □

Lemma 2 *Under no load shedding, if an itemset X is deleted at condow w_d, then its true frequency between w_d and any previous window w_i is no more than $(w_d - w_i + 1) \times a[|X|]$; i.e., $f_T(w_i, w_d) \leqslant (w_d - w_i + 1) \times a[|X|]$.*

Proof. We distinguish two cases. For the first case, where X's frequency is not counted at w_i. Then, from w_i to its next insertion at w_s^i (but not include w_s^i itself), its true frequency at each condow is no more than $a[|X|]$. Thus, $f_T(w_i, w_s^i - 1) \leqslant (w_s^i - w_i) \times a[|X|]$. On the other hand, by Lemma 1, its true frequency since w_s^i to w_d is bounded by $f_T(w_s^i, w_d) \leqslant (w_d - w_s^i + 1) \times a[|X|]$. Therefore, $f_T(w_i, w_d) \leqslant (w_d - w_i + 1) \times a[|X|]$.

For the second case where X's frequency is counted at w_i, we prove by contradiction. Assume that $f_T(w_i, w_d) > (w_d - w_i + 1) \times a[|X|]$. Let us denote the closest condows of w_i, at which X is inserted and deleted by w_s^i and w_d^i respectively (i.e., $w_s^i \leqslant w_i \leqslant w_d^i$). By Lemma 1, $f_T(w_d^i + 1, w_d) \leqslant (w_d - w_d^i) \times a[|X|]$. Meanwhile, we have assumed that $f_T(w_i, w_d) > (w_d - w_i + 1) \times a[|X|]$. Accordingly, its true frequency between w_i and w_d^i must be greater than $(w_d^i - w_i + 1) \times a[|X|]$. Together with the fact that X has been counting since w_s^i, we have $f_E(w_s^i, w_i - 1) > (w_i - 1 - w_s^i) \times a[|X|]$. Consequently $f_E(w_s^i, w_d^i) > (w_d^i - w_s^i) \times a[|X|]$, making X not be deleted at w_d^i. This contradicts the fact that X was deleted at w_d^i. The lemma is proved. □

Lemma 2 is important since it allows to approximate the maximal error of every itemset discovering within the sliding window regardless of the new index value of w_o. Note that the value of w_o is arbitrary since the number of condows within each time period is variable.

Theorem 1 *Let w_o and w_c be the oldest and current condows of the sliding window respectively. Under no load shedding, $s_T(w_o, w_c) \leqslant s_E(w_o, w_c) + \gamma$.*

Proof. If X is inserted at w_o, the first condow of the current sliding window, its maximum error at this condow is at most $a[|X|]$. Therefore, $f_T(w_o, w_c) \leqslant f_E(w_o, w_c) + a[|X|]$. Otherwise, X is possibly deleted some time earlier, as late as, at condow $w_i - 1$, and then inserted into $\mathcal{S}$ at condow w_i. By Lemma 2, $f_T(w_o, w_i - 1)$ is at most $(w_i - w_o) \times a[|X|]$ when such a deletion took place. Since $a[|X|]$ is the maximal error when X is inserted at w_i and $f_E(w_i, w_c)$ is its frequency count since then, it follows that $f_T(w_o, w_c) \leqslant f_E(w_i, w_c) + (w_i - w_o + 1) \times a[|X|]$.

On the other hand, let n_w be the number of transactions in the sliding window. Certainly, $\Delta \times (w_i - w_o + 1) \leqslant n_w$ (since $w_i \leqslant w_c$), we have $(w_i - w_o + 1) \times a[|X|] \leqslant \gamma \times n_w$. Therefore, dividing the inequality above for n_w, we derive $s_T(w_o, w_c) \leqslant s_E(w_o, w_c) + \gamma$. □

Theorem 2 *Under load shedding, $s_T(w_o, w_c) \leqslant s_E(w_o, w_c) + \gamma + \epsilon$ with a probability of at least $1 - \delta$.*

Proof. This theorem can be directly derived from the Chernoff bound and Theorem 1 above. At time slots where the load shedding happens, the true support of X is guaranteed within $\pm\epsilon$ of the counting support with probability $1 - \delta$ when the Chernoff bound is applied for sampling. Meanwhile, by Theorem 1, this counting support is limited by $s_T(w_o, w_c) \leqslant s_E(w_o, w_c) + \gamma$. Therefore, getting the upper bound (i.e., ϵ) in the Chernoff bound, we derive the true support of X to be no more then its estimated support by $\gamma + \epsilon$ with a confidence of $1 - \delta$. □

4 Experimental Results

We implemented our algorithm in C++ and performed experiments on a 1.9GHz Pentium machine with 1GB of memory running Windows XP. Three synthetic data streams are generated with a size of 3 million transactions each by using the IBM data generator (Agrawal et al. 1994) (we are not able to obtain any real data set that is large enough to simulate data streams). We obtained dataset T10.I6.D3000K by setting average transaction size T=10, average maximal frequent set size I=6. With similar setting, we obtained T8.I4.D3000K and T5.I3.D3000K. For all three datasets, the number of maximal potentially frequent itemsets L=2000 and the number of unique items N=10,000 (for more detail on dataset generation, we refer the reader to (Agrawal et al. 1994)). Since our approach is probabilistic, the precision and recall measures will be used to evaluate the effectiveness of our algorithm. For the same reason, each experiment is repeated 10 times for each parameter combination and the average results are reported.

4.1 Accuracy measurements

In our experiments, we fix $\epsilon = 0.001, \delta = 0.01$ and $p = 0.002$. Accordingly, the number of transactions for each condow as determined by the Chernoff bound is $\Delta \approx 20K$. Since $p = 0.002$, the minimum support threshold σ will be chosen to be no less than this value. Specifically, σ will be varied between 0.002 and 0.01. We also fix $\gamma = 0.1\sigma$ (except $\gamma = 0.25\sigma$ for $\sigma = 0.002$) and the maximal length of frequent patterns $m = 10$ (interval between $a[i]$ is $\gamma\Delta/m$). The sliding window consists of 10 time slots and each time slot receives $200K$ transactions (i.e., 10 condows). For simplicity, instead of varying loads, we fix the load and change the number of condows that the system can handle at each time period. For example, if only 2 of 10 condows can be processed in each time slot, this translates to an input stream rate that is 5 times higher than the CPU capacity and thus, the percentage of load shedding is 80%.

Figure 1 shows our experiment results on two datasets T8.I4.D3000K and T10.I6.D3000K where the

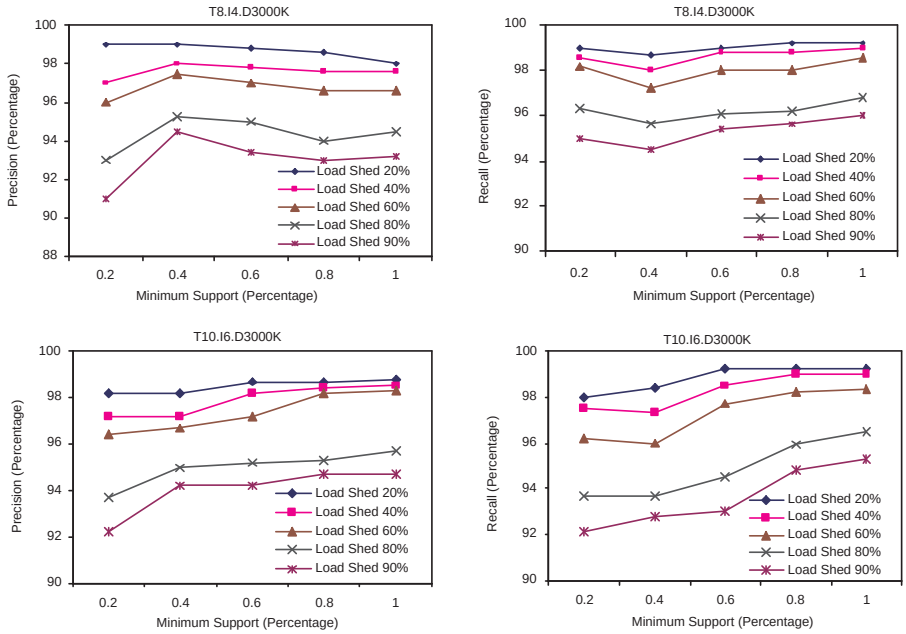


Figure 1: Accuracy on T8.I4.D3000K and T10.I6.D3000K

minimum support threshold is varied from 0.2% to 1% and the load shed is increased from 20% to 90%. We observed that at lower levels of system workload, the algorithm returns a higher number of true frequent itemsets as indicated by the higher value of recall, and a smaller number of false frequent itemsets from the precision measure. A larger gap in accuracy occurs when the load shed increases from 60% to 80% and 90%. At these levels, only 2 and 1 condows are processed (respectively) at each time slot. Nevertheless, the algorithm continues to find more than 92% of all true frequent itemsets (while keeping the false frequent itemsets at below 10%) even though the system workload has increased 10 times that of the CPU capacity, i.e., a load shed of 90%. And this is maintained across all levels of support thresholds.

More details are shown in Figure 2 when we plot the precision and recall for each itemset length. We report the results on T10.I6.D3000K with $\sigma = 0.2\%$ since at this threshold, it is possible to find frequent sets that has a length of up to 10. It can be observed that the precision decreases as the length of a frequent set increases. This happens because the longer patterns are generated only after *all* its subsets are found to be significantly frequent. Hence, the precision on longer itemsets is expected to be lower as a consequence of the larger margin error. Fortunately, distinguishing the error by its pattern length allows the algorithm to closely approximate the maximum lost for each pattern. As seen in Figure 2, the precision on 10-itemsets is maintained above 80%. Note that the recall is not affected by this approximation as the true frequency of itemsets is guaranteed by the Chernoff bound, which is generally dependent on the number of sampling transactions. We observed that other than very large load shedding (more than 80%), the recall is always found to be higher than 92% for every itemset length.

4.2 Adaptability

To test the adaptability of our algorithm, we use T5-8-10.D3000K where its first, second and third parts are made up of each 1000K transactions from T5.I3.D3000K, T8.I4.D3000K, and T10.I6.D3000K. To evaluate the workload using Equation 2, we first set the CPU capacity to be infinite and let the algorithm process the entire dataset. Figure 3(a) shows the relationship between the processing time and the statistics measured within each condow, where the range of support threshold varies between 0.2% and 0.6%. In the figure, σT and σT(w/o MFI) respectively indicates the time to process a condow when we enable and disable the function detecting workload using MFI tree. σS refers to the statistics computed in Equation 2 for each condow. As seen from the figure, σT and σT(w/o MFI) are very close to each other, indicating that the time to compute the statistics is very small compared to the time to find all frequent sets from the stream. When using MFI tree to evaluate workload, it only incurs more CPU usage when the algorithm updates this structure. However, this operation is only performed when a significant change in workload is detected in the stream or it is periodically performed for a set of condows. In the experiments presented, the MFI tree is updated at every time slot (or every 10 condows).

Also from Figure 3(a) we observe that the statistics computed is almost linearly proportional to the processing time across the support thresholds experimented. For instance, at $\sigma = 0.4\%$, the average statistics computed for one condow for transactions sized T5 is 160.7K, T8 is 514.3K and T10 is 792.5K; and the average time to process them are respectively 4.1s, 13.0s and 19.8s. What these statistics mean is that if we limit the CPU capacity and the data stream is initially sent to the system at a rate just below its CPU capacity, then the algorithm will have to shed load by sampling at a rate of 1/3 ($\approx \frac{160.7}{514.3} \approx \frac{4.1}{13}$) and 1/5 ($\approx \frac{160.7}{792.5} \approx \frac{4.1}{19.8}$) in the second and third parts of the stream. In other words, it is effective to use the computed statistics to identify the appropriate amount of data for shedding.

In Figure 3(b), we report the accuracy of our algorithm for this experiment. As we can observe, the recall at all minimum support thresholds is still very high ($\geqslant 94\%$) and is likely the same. Nevertheless, we find that the precision is slightly lower than that in

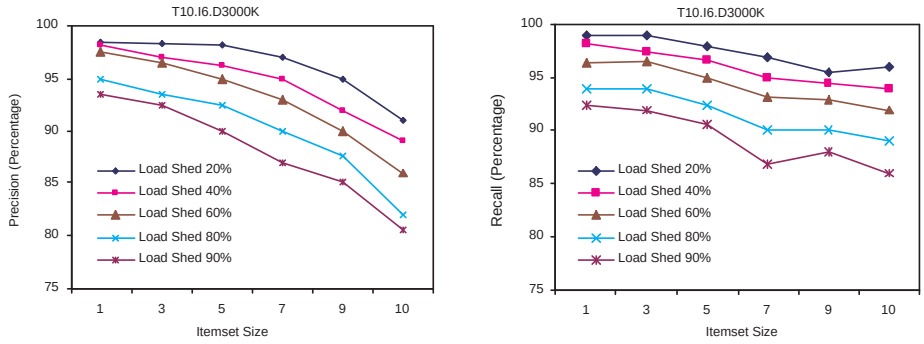


Figure 2: Accuracy vs. Itemset Length for T10.I6.D3000K

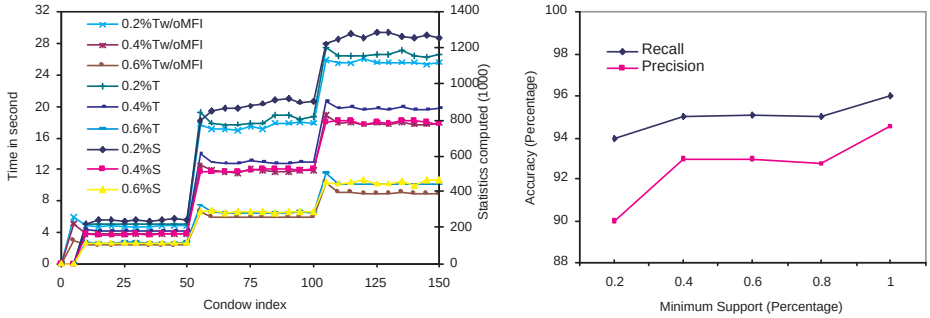


Figure 3: Adaptability on T5-8-10.D3000K

the pure dataset T10.I6.D3000K. This happens since we note that the last sliding window also includes 1000K transactions sized T8 and when the algorithm processes T10.I6.D1000K, more frequent itemsets are found in this part of the stream. According to our approximation (when estimating $(X.Wid - w_o) \times a[|X|]$ to be X's maximal frequency lost), a small fraction of those itemsets discovered in T10.I6.D1000K have over-estimated frequencies. This means that they were locally frequent in the third part of the stream, but not in the second part. However, it is interesting to note that at all support thresholds examined, the percentage of false frequent itemsets is consistently within 10%.

5 Related Work

The issue of dropping a fraction of data sets when the system is overloaded has been intensively studied in real-time databases (Wolfgang & Alexander 1994). Recently, this problem has been extended to data stream querying (Tatbul et al. 2003, Babcock et al. 2004, Tu et al. 2006, Tatbul et al. 2006). In this context, the problem of load shedding is defined as the process of finding an optimal plan for inserting dropping operations in a query network. The Aurora system is one of the first stream querying projects addressing this issue. This work utilizes various Quality of Service (QoS) graphs to represent different important levels of querying objects. Accordingly, when the system is overloaded, arriving transactions will be shed progressively starting from those containing less important objects. This project has recently developed to Borealis, a distributed stream processing system, where the load shedding problem is extended for sliding window model (Tatbul et al. 2006). The STREAM project (Babcock et al. 2004) is another work in this field. In STREAM, a load shedding scheme based on sampling is proposed for sliding window aggregate queries. This approach modifies the query network by inserting load shedder operators together with their sampling rates in such a way that the total sampling rate eliminates sufficient amount of dropping data. This work is similar to ours in that the random sampling is used as a means of load shedding. However, it addresses the problem of querying objects with optimized sampling rates in a query network while our work addresses the issue in the context of frequent set mining. In data stream mining, Loadstar (Chi et al. 2005) is recently reported as the first work focused on the load shedding problem for classification. In this work, various Quality of Decision (QoD) metrics is introduced to measure uncertainty levels in classification when exact values of the data are not available because of load shedding. A Markov model is utilized to predict the distribution of feature values and then classification decisions are made based on predicted values and QoD metrics.

On the other hand, recent work on mining frequent itemsets over data streams can be classified into three models. The first model is the *landmark* model where frequent sets are discovered between a particular point of time (called landmark) and the current time. Lossy Counting (Manku et al. 2002) and FDPM (Jeffrey et al. 2004) are typical algorithms in this group. The second model is the *time-fading* model where transactions are weighted on the time they arrive. This model gives more attention (fine granularity) to the recently arrived data and relaxes for the earlier ones (coarse granularity). Works that focus on this model include the estDec (Chang et al. 2003*a*) and FP-Streaming (Giannella et al. 2003) algorithms. The third model is the *sliding-window* model. Compared to the two previous ones, this model further considers the elimination of transactions. Frequent itemsets are found within a fixed portion of the stream

which is pre-specified by a period of time or a number of transactions. FTP-DS (Teng et al. 2003) (Frequent Temporal Patterns of Data Streams), estWin (Chang et al. 2003*b*), and Time-sensitive sliding window (Lin et al. 2005) are algorithms focusing on this model. A core idea underlying all these works is that they focus on designing methods that can efficiently summarize within limits of main memory. Our work (classified to the sliding window model) further addresses the load shedding problem when CPU capacity is not sufficient to manage all streaming transactions. More importantly, the mining results are approximated and guaranteed within a precise error threshold.

6 Conclusions

In this paper, we visit the issue of applications with data streams that have a variable arrival rate and data characteristics that could inadvertently push the workload above the system's capacity. To avoid such situations, it is important to design algorithms that can adapt to the underlying changes in the data characteristics to guarantee the timely delivery of results. Finding frequent patterns is an exemplar of a difficult problem in such a situation, and our paper presents the solution to this problem. An algorithm to approximate the set of frequent patterns from the data stream while taking the limitation of CPU capacity is proposed. Not only does our proposed algorithm perform well in overloaded situations, the results that it delivers are well guaranteed to be within the error bounds specified across patterns of different lengths. In addition to the theoretical proofs, our empirical evidences provided the verification that our proposal is effective in dealing with such data streams.

Acknowledgements. This work is supported in part by grant P0520095 from the Agency for Science, Technology and Research (A*STAR), Singapore.

References

Agrawal, R. & Srikant, R. (1994), Fast Algorithms for Mining Association Rules in Large Databases, *in* 'VLDB International Conference on Very Large Data Bases', pp. 489–499.

Babcock, B., Datar, M. & Motwani, R. (2004), Load Shedding for Aggregation Queries over Data Streams, *in* 'VLDB International Conference on Data Engineering', pp. 350–361.

Babcock, B., Babu, S., Datar, M., Motwani, R. & Widom, J. (2002), Models and Issues in Data Stream Systems, *in* 'ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems', pp. 1–16.

Lin, C., Chiu, D., Wu, Y. & Chen, A. (2005), Mining Frequent Itemsets from Data Streams with a Time-Sensitive Sliding Window, *in* 'SIAM International Conference on Data Mining', pp. 68–79.

Das, A., Gehrke, J. & Riedewald, M. (2003), Approximate join processing over data streams, *in* 'ACM SIGMOD International Conference on Management of Data', pp. 40–51.

Garofalakis, M., Gehrke, J. & Rastogi, R. (2002), Querying and mining data streams: you only get one look a tutorial, *in* 'ACM SIGMOD International Conference on Management of Data'.

Gedik, B., Wu, K.L., Yu, P.S. & Liu, L. (2005), Adaptive load shedding for windowed stream joins, *in* 'ACM CIKM International Conference on Information and Knowledge Management', pp. 171–178.

Giannella, C., Han, J., Pei, J., Yan, X. & Yu, P.S. (2003), *Mining Frequent Patterns in Data Streams at Multiple Time Granularities*, Next Generation Data Mining AAAI/MIT.

Guha, S., Meyerson, A., Mishra, N., Motwani, R. & O'Callaghan, L. (2003), Clustering Data Streams: Theory and Practice, *in* 'IEEE Transactions on Knowledge and Data Engineering', pp. 515–528.

Chernoff, H. (1952), A measure of asymptotic efficiency for thests of a hypothesis based on the sum of observations, *in* 'Annals of Mathematical Statistics', pp. 493–509.

Hulten, G., Spencer, L. & Domingos, P. (2001), Mining Time-Changing Data Streams, *in* 'ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', pp. 97–106.

Wolfgang, H. & Alexander, S. (1994), *Real time computing*, Springer-Verlag.

Chang, J.H. & Lee, W.S. (2003*a*), Finding recent frequent itemsets adaptively over online data streams, *in* 'ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', pp. 487–492.

Chang, J.H. & Lee, W.S. (2003*b*), EstWin: Adaptively monitoring recent change of frequent itemsets over online data streams, *in* 'ACM CIKM International Conference on Information and Knowledge Management', pp. 536–539.

Manku, G.S. & Motwani, R. (2002), Approximate Frequency Counts over Data Streams, *in* 'VLDB International Conference on Very Large Data Bases', pp. 346-357.

Tatbul, N. & Zdonik, S.B. (2006), Window-Aware Load Shedding for Aggregation Queries over Data Streams, *in* 'VLDB International Conference on Very Large Data Bases', pp. 799-810.

Tatbul, N., Çetintemel, U., Zdonik, S.B., Cherniack, M. & Stonebraker, M. (2003), Load Shedding in a Data Stream Manager, *in* 'VLDB International Conference on Very Large Data Bases', pp. 309-320.

Teng, W.G., Chen, M.S., & Yu, P.S. (2003), A Regression-Based Temporal Pattern Mining Scheme for Data Streams, *in* 'VLDB International Conference on Very Large Data Bases', pp. 93-104.

Tu, Y.C., Liu, S., Prabhakar, S. & Yao, B. (2006), Load Shedding in Stream Databases: A Control-Based Approach, *in* 'VLDB International Conference on Very Large Data Bases', pp. 787-798.

Guizhen Yang (2004), The complexity of mining maximal frequent itemsets and maximal frequent patterns, *in* 'ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', pp. 344-353.

Chi, Y., Yu, P.S., Wang, H. & Muntz, R.R. (2005), Loadstar: A Load Shedding Scheme for Classifying Data Streams, *in* 'SIAM International Conference on Data Mining', pp. 346–357.

Yu, J.X., Chong, Z., Lu, H. & Zhou, A. (2004), False Positive or False Negative: Mining Frequent Itemsets from High Speed Transactional Data Streams, *in* 'VLDB International Conference on Very Large Data Bases', pp. 204-215.

SemGrAM - Integrating Semantic Graphs into Association Rule Mining

John F. Roddick[1] **and Peter Fule**[1,2]

[1] School of Informatics and Engineering,
Flinders University of South Australia,
PO Box 2100, Adelaide, South Australia 5001
Email: roddick@infoeng.flinders.edu.au

[2] Defence Science and Technology Organisation
PO Box 1500, Edinburgh, South Australia 5111
Email: Peter.Fule@dsto.defence.gov.au

Abstract

To date, most association rule mining algorithms have assumed that the domains of items are either discrete or, in a limited number of cases, hierarchical, categorical or linear. This constrains the search for interesting rules to those that satisfy the specified quality metrics as independent values or as higher level concepts of those values. However, in many cases the determination of a single hierarchy is not practicable and, for many datasets, an item's value may be taken from a domain that is more conveniently structured as a graph with weights indicating semantic (or conceptual) distance. Research in the development of algorithms that generate disjunctive association rules has allowed the production of rules such as $Radios \vee TVs \rightarrow Cables$. In many cases there is little semantic relationship between the disjunctive terms and arguably less readable rules such as $Radios \vee Tuesday \rightarrow Cables$ can result. This paper describes two association rule mining algorithms, SemGrAM$_G$ and SemGrAM$_P$, that accommodate conceptual distance information contained in a semantic graph. The SemGrAM algorithms permit the discovery of rules that include an association between sets of cognate groups of item values. The paper discusses the algorithms, the design decisions made during their development and some experimental results.

Keywords: Association Mining, SemGrAM, SemGrAM$_G$, SemGrAM$_P$, Disjunctive Rules, Semantic Graphs.

1 Introduction

Current association rule mining algorithms make a number of assumptions about the domains over which items are defined. In early work, the domains were assumed to be binary – the existence (or not) of an item in a transaction (Agrawal et al. 1993). This was extended to handle discrete domains (often by simply qualifying the item with the attribute name) and hierarchical domains (Han & Fu 1999, Lu 1997, Shen & Shen 1998, Suk & Park 1999). Categorical and linear domains have also been accommodated (Lent et al. 1997, Gray & Orlowska 1998) as have fuzzy data (Kuok et al. 1998), spatial data (Koperski & Han 1995) and temporal data (Roddick & Spiliopoulou 2002). Ignoring such domain structure constrains the search and may result in missed rules – assuming discrete values, for example, means that item values must satisfy the quality metrics as independent values. To our knowledge, a single algorithm capable of handling more than one type of domain structure has not been developed.

The accommodation of hierarchies allows higher level concepts of those values through predefined or dynamically generated concept trees. For example, rules such as

$$Sunday, Coffee \rightarrow Croissant$$

may be found where

$$Coffee \supseteq \{Cappuccino, Latte, Macchiato\}$$

Unfortunately, this may convey the impression that *Latte* contributes to the rule when it may not. Moreover, in many cases the determination of a single hierarchy is not possible. Indeed, for many datasets, hierarchies may be imposed when it would be more appropriate to define the domain over a graph with weights indicating semantic (or conceptual) distance (see Figure 1). Apart from those domains that lend themselves to graph representation, directed graphs have the advantage that they subsume other domain structures.

Disjunctive association rule generation algorithms (Nanavati et al. 2001) aim to create rules that include disjunctive combinations of terms such as:

$$Sunday \vee Tuesday, Macchiato \rightarrow Croissant$$

Disjunctive rules are flexible in that no domain knowledge is required and perform well in many domains, particularly where Zipf's Law is evident such as in market basket data. However, often the rules produced contain disjunctions between unrelated terms, for example:

$$Sunday \vee Water, Macchiato \rightarrow Croissant$$

This mixing of concepts can reduce the readability of the results. Moreover, the disjunctive rule generation techniques outlined to date might combine two items, such as *Sunday* and *Tuesday* and omit *Monday* which might, for this dataset, just fall short of the metrics specified – that is, the semantic proximity of items is not taken into account when the disjunctive sets are formed.

One associated issue for data mining is the problem of scaling effects which occur particular for spatial and temporal data but can occur more pervasively. Essentially, many analyses are sensitive to the

length, interval, area, volume or metric over which a variable is distributed. Often termed the *Modifiable Areal Unit Problem* (MAUP), or the *Ecological Fallacy*, it is an important characteristic in many problems (Openshaw 1983). Put simply, the granularity chosen for data collection determines which spatial or other phenomena can be identified. If spatial data are aggregated then the larger the unit of aggregation the more likely attributes will be correlated. Moreover, by aggregating into different groups you can get different results. Importantly for data mining, attributes which exist in the data at one level of support can vanish at coarser or finer scales, or other orientations. Thus the development of an algorithm capable of aggregating data *at the level of significance* is important.

In this paper we propose two algorithms[1], SemGrAM$_G$ and SemGrAM$_P$, that are able to use conceptual distance information, contained in one or more semantic graphs, within an association rule mining system to produce association rules with a new type of item grouping. The algorithms dynamically join elementary items into composite *itemgroups* within the itemsets. The itemgroup thus formed represents a disjunctive aggregation of a number of items that are similar, as determined by the semantic graph. The increased support of *itemgroups*, and that of the resulting itemset, can be calculated to find association rules from the itemset. In effect, *itemgroups* allow for specialised disjunctions of similar items in a single association – a particular form of disjunctive rule (cf. (Nanavati et al. 2001)). For example:

$$[Tuesday, Wednesday], Cappuccino \rightarrow Croissant$$

In SemGrAM, more than one semantic graph can be used. In order to disambiguate the reason for an itemgroup's construction, where there is need, the graph is noted. For example,

$$[Adelaide, California]_{Weather} \rightarrow Wine$$

or

$$[Montana, Idaho]_{Proximity} \rightarrow BlackBear$$

The advantages of such an approach include the following:

- A finer granularity of the items able to be included in the source data set without loss of succinctness in the resulting rule. Unlike presupplied hierarchies, the items included in an itemgroup are determined dynamically and thus can be constructed to include only items that contribute to the rule in a meaningful way,
- The ability to adopt different sets of conceptual distances for different tasks over the same dataset,
- The capture of itemsets that are otherwise below threshold but nevertheless contain useful information when items are joined,
- The ability to incorporate domains that are more complex than those already accommodated. That is, to capture semantics of trees, circular lists, and so on as a result of the subsuming semantics of graphs. Moreover, there is the ability to accommodate multiple domain structures such as lists of trees,
- Its use in text mining to consolidate terms with similar meaning but differing representation (qv. (Mooney et al. 2006)), and
- The clustering of rules with spatial attributes within them (accommodating some of the advantages of the work of Lent et al. (1997)).

Figure 1 shows some examples of semantic graphs that present domain knowledge. Using the Stock Description graph (Figure 1(a)), consider a set of results containing one itemset that includes ottoman and a second including sofa with the remainder of the itemset in common. If both itemsets fall below the support threshold we can use the information in the semantic graph to determine that an ottoman is similar to a sofa and create a new itemgroup – [sofa, ottoman] – as an abstraction of the two similar items. The new itemset incorporating the itemgroup will have a higher support, possibly meeting the support threshold. That itemset, and any resulting rules, effectively captures information about a concept made from the itemgroup formed by a joining the two items.

2 Related Work

This approach differs from previous research. For example, the work of Srikant & Agrawal (1997) uses an unweighted *is-a* (directed acyclic graph) hierarchy. In this work we allow the use of weighted graphs and remove the acyclic condition. The difference is significant and results not only in a different algorithm being needed but also in rules possessing a different semantic structure.

Multi-level association rule algorithm research (Han & Fu 1999, Shen & Shen 1998, Ong et al. 2001) also differs from that outlined here. Han and Fu, for example, develop a top-down approach using *a priori* supplied hierarchies. While our approach requires the semantic graph to be supplied, the itemgroups discussed here are developed dynamically and, significantly, for SemGrAM$_G$ may differ from rule to rule. There is further discussion of related work in Section 2. Informally, the process we adopt is the aggregation of items into *itemgroups* in cases where two or more k-itemsets do not possess the required support by themselves but where all the items in the itemsets are either identical or pairwise conceptually similar. This is discussed in detail in Section 3.3.

The idea of grouping association rules has been explored previously. For example, Lent *et al.* cluster association rules to find more general associations (Lent et al. 1997). Their system provides for association rules that contain quantitative attributes which are clustered, as a individual points, on a two dimensional plane. The plane is constructed from the quantitative linear attributes found in the rules with the clusters representing the groups of rules. The research presented in this paper clusters both categorical and non-categorical attributes within the itemsets and thus the work of Lent *et al.* is complementary to this work.

An algorithm for creating disjunctive association rules has been presented by Nanavati et al. (2001). Their work inspects itemsets creating generalised disjunctions without using semantic graphs. As such, their work is on the one hand more flexible (as no graph is needed) but may also be too general as it produces rules without reference to the conceptual distance between items, and thus may group dissimilar items.

Mining with level wise abstraction (Han & Fu 1999, Ceglar et al. 2005) is also similar in that the rules that are created contain items that are an abstraction of lower level items. The level wise approach

[1] Where there is no necessity to distinguish between SemGrAM$_G$ and SemGrAM$_P$ we simply refer to them as SemGrAM.

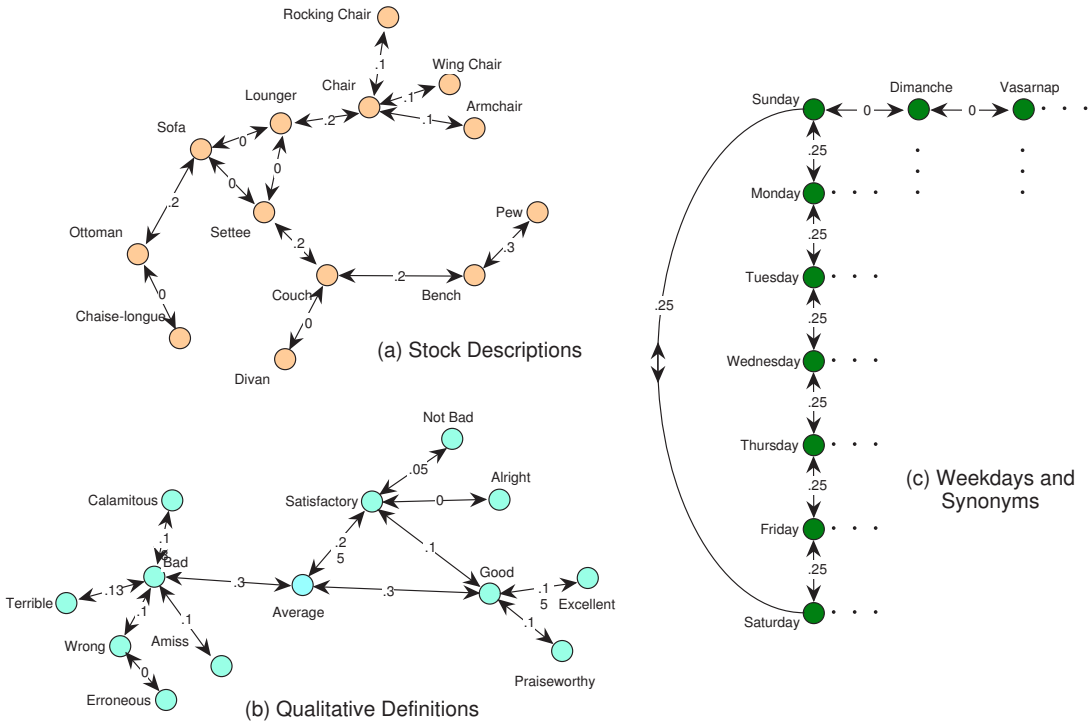


Figure 1: Situations in which graphs are used can be common.

requires a hierarchy containing the items as leaf nodes and adjusts the support threshold for the level of abstraction of the component items. The research presented here uses a semantic graph to combine items dynamically, so that they rise to meet an unchanging support threshold. Hierarchies are, of course, special cases of semantic graphs and some of the ideas presented by Han, Fu and others are applicable here also.

In other work, Han et al. (1997) present a system that constructs a hypergraph based on the confidence of association rules, then clusters the items by partitioning the graph. Further research by Guha et al. (1999) indicates that the algorithm breaks down in certain cases. Guha *et al.* present the ROCK algorithm for clustering transactions. The links counting algorithm they present could have been used to generate the distance matrix presentation for the SemGrAM algorithm discussed later.

In terms of other *higher order* algorithms, Kosters et al. (1999) describe a system that clusters transactions based on the association rules generated from them. It takes the rules with the highest support, using the antecedents as the clusters for the transactions, to create a hierarchical clustering scheme for the data set. Similarly, Ertöz et al. (2002, 2003) present a nearest shared neighbour approach to clustering documents which is based on an algorithm by Jarvis & Patrick (1973).

Finally, Mazlack & Coppock (2002) present research into data preparation techniques involving the partitioning of the values of the input data set to help produce better results. The ideas presented focus on attributes with qualitative values and the best methods of partitioning those values globally. The work presented here partitions categorical attributes into new items that best suit the generation of new results *for each subset of itemsets*, although it should be noted that the granularity of partitioning of non-categorical attributes influences the results of the algorithm presented here.

3 Algorithm Design and Description

3.1 Semantic Graphs

The advantage of graphs is that they subsume all other structures including lists and trees with weighted uni-directional graphs being the most general. Importantly, although they are not used widely, semantic graphs are not uncommon – WordNet (Fellbaum 1998, Budanitsky & Hirst 2000), Roget's Thesaurus (Jarmasz 2003), colour chart comparisons (CMYK / RBG / websafe / proprietary descriptions ...), geographic features, and so on provide a substantial resource and are readily available. Importantly, many such resources are not readily accommodated in a hierarchy and thus multi-level association rule mining solutions cannot be employed.

In addition, the MAUP (discussed earlier in Section 1) is accommodated by allowing graphs of different scales to be used with a fixed support threshold. That is, regardless of the granularity, rules with the predefined interest level will be reported.

For the purposes of this work graphs are assigned to a *family* with those in the same family able to be combined when creating an itemgroup. For example, consider the three semantic graphs – html colours, colour descriptions and geographic markers. A node in the html colour graph, say `xFFA500`, is comparable to the description `orange` and thus distances in the two graphs are aggregative. Such graphs are considered to be in the same *family*. Values in the third, geographic markers, are not comparable and would thus be placed in a different family.

In SemGrAM we allow graphs to be combined within each family as long as the edge weights can be normalised. In practice, we assume all graphs to be in different families unless two points of contact are specified between two graphs. Using these two points, the relativities between the weights used in each graph can be checked against the value of the *traversal threshold* τ (discussed in the next section).

3.2 Terminology

The input dataset consists of a finite number of transactions each containing a subset of the finite number of items S,

$$S = \{i_1, i_2, \ldots, i_n\}$$

Each item i_j is provided either as a simple value (eg. *Blue*) or as an attribute.value pair (eg. *Colour.Blue*). If the former is used, the prefix can be used to determine which families of graphs are appropriate for that item. In SemGrAM, we correlate graph families to attribute names through a simple list.

The input data is provided as transactions, with each transaction T_i containing a variable number of non-repeated items from S,

$$T_x = \{i_{x_1}, i_{x_2}, \ldots, i_{x_q}\} : \forall i_j \in S,\, q \geq 1$$

Each transaction generally contains significantly fewer items than are present in S, i.e. $q \ll n$.

An *itemgroup* G_i is a set of elementary items $[i_1 \ldots i_n]_\Gamma$ grouped for the purposes of itemset formation by virtue of their proximity in semantic graph Γ. The *itemgroup* G_i is then considered an atomic item within any *itemset* I_j[2]. Itemsets can consist of either or both of elementary items or itemgroups, (i.e. $I_j = \{i_1 \ldots i_m, G_1 \ldots G_n\}$) however, once grouped, an itemgroup is treated as an atomic item for all subsequent purposes with respect to that itemset. Thus for the example above, the itemgroup [sofa, ottoman]$_{Stock}$ might be contained within the itemset {Green, [sofa, ottoman]$_{Stock}$}. Note that the grouping of items $[i_1 \ldots i_n]$ as an itemgroup in I_j does not imply that they will be grouped in the same way in some different itemset I_k.

A semantic graph Γ_i is defined as a weighted, directional graph. Formally, each graph is a 4-tuple

$$\Gamma = \{S, E, \tau, \Phi\}$$

where a subset of the items in S are represented by nodes in the graph. The set of edges

$$E = \{e_1, e_2, \ldots, e_k\}$$

with each edge

$$e = \{i_x, i_y, d\} : \forall i_j \in S,\, 0 \leq d \leq \tau$$

between the nodes represents a semantic distance between the items in the context of that graph. Each edge has a distance d representing the strength of the relationship between the two items it connects, higher values indicating a more distant or weaker relationship and zero indicating a synonym[3]. Any edge with a traversal distance greater than the maximum defined *traversal threshold* τ is excluded from Γ_i. Each graph is assigned to a family Φ_i.

Items omitted from the graph (or included without a connecting edge) are assumed to be dissimilar (i.e. to have infinite distance between them). The *traversal threshold* τ is used to normalise distances across multiple graphs, making the scale used in the construction of the graph unimportant.

Semantic graphs are created either from expert knowledge of the context from which input dataset S is taken or are extracted from generally available knowledge. In SemGrAM, the graphs are stored as a dataset of triples $< i_x, i_y, d >$, from which transitive distances are obtained recursively.

[2]For clarity we use square brackets for itemgroups and curly brackets for itemsets. Where obvious, the suffix indicating the graph is omitted.

[3]The semantic graph traversal concepts are explained in more detail elsewhere (Roddick et al. 2003).

3.3 The SemGrAM Algorithms

While the ideas behind SemGrAM are common, this section describes two distinct algorithms, SemGrAM$_G$ and SemGrAM$_P$ and discusses some of the design decisions. SemGrAM$_G$ is a flexible, but greedy algorithm while SemGrAM$_P$ is more efficient but imposes some constraints on the ruleset discovered. Specifically,

SemGrAM$_G$ operates in a greedy manner by aggregating appropriate itemsets that have a support that falls just under the minimum support threshold. As a result, SemGrAM$_G$ is able to use the semantic information to combine itemsets in different ways for different sets of rules. It is also independent of the underlying itemset generation algorithm.

SemGrAM$_P$ operates parsimoniously by amending FP-Trees and is thus more efficient but results in a ruleset where the same merger of items into an itemgroup may appear in multiple rules. SemGrAM$_P$ is at present based on the manipulation of FP-Trees and thus tied to FPGrowth (Han et al. 2000).

As for all association rule mining routines, the SemGrAM algorithms mine transactions to find common and significant co-occurrences of items[4]. Association rule mining routines typically utilise, *inter alia*, a support metric σ, which indicates the frequency of the co-occurrence of the items contained within each itemset,

$$I_x = \{i_{x_1}, i_{x_2}, \ldots, i_{x_m} \mid \forall i \in S,\, m \geq 1\},\, \sigma$$

where m indicates the cardinality of the itemset. The itemset can be viewed as an intersection of the items it contains where the support indicates the strength of the intersection.

SemGrAM uses three user defined support thresholds.

1. The traditional support threshold (σ) that applies to all itemsets. If the support of any itemset is less than this threshold then the itemset is not used for rule production and thus not reported in the final set of results.

2. A *near support threshold* (β) to partition itemsets of low cardinality, with itemsets that have support between σ and β termed *near support itemsets* or *nsi*'s. The range of support values between the normal and near support values is termed the *near support range*[5].

3. An *itemgroup cohesion threshold* (γ). When an itemgroup is created the cohesion of the group is assessed, and if below γ is removed from consideration. Finding itemgroups is an optimisation problem that balances the potential gain in support through grouping the items with the loss of semantic precision (the *cohesion*) as items are added. For example if an item was defined over a graph of colour hues, red would be similar to crimson and vermilion, and may be grouped if the circumstances suggested it. If the itemset containing this itemgroup was still unable to reach the regular support, it may need to widen the semantics of the itemgroup by using other higher support items that were conceptually more distant. If pink, for example, had a high support

[4]For a full survey of association mining algorithms see the recent survey by Ceglar & Roddick (2006).

[5]The concept of *nsi*'s has already been investigated for other purposes in research into incremental association rule mining (Cheung et al. 1996, Rainsford et al. 1997, Kouris et al. 2003, Lee et al. 2005).

it may be beneficial to include it in the itemgroup to help the itemset reach the normal support threshold, but it would start to stretch the semantic cohesion of the itemgroup; γ controls this semantic spread.

The thresholds have been adopted because the cognitive and computational complexity of merging low cardinality itemsets can be high. β and γ thus enable the user to manage the scope of the additional itemsets[6].

3.3.1 SemGrAM$_G$

The mining algorithm used as a base for SemGrAM$_G$ could have been chosen from any of the existing algorithms including, for example, Apriori (Agrawal et al. 1993), FPGrowth (Han et al. 2000) or Eclat (Zaki et al. 1997), as long as the algorithm is capable of supporting the multiple thresholds. In our implementation (see Section 4) we use the FPGrowth algorithm.

Algorithm 3.1 SemGrAM$_G$ β-graph construction

```
1: Generate FP-tree
2: for each itemset I_j : β ≤ support(I_j) < σ do
3:    add I_j node to β-graph;
4:    for each item I_k ∈ β-graph do
5:       if |I_j| = |I_k| and diff( I_j , I_k ) = 1 then
6:          x, y = differing values
7:          for each graph family Φ_i do
8:             if graph Φ_i is applicable to both x and y then
9:                weight = ∞
10:               for each graph Γ_j ∈ Φ_i do
11:                  weight = min(weight, dist(x,y,Γ_j) / τ_Γj)
12:               end for
13:               if weight ≤ γ then
14:                  create edge e_{j,k} between I_j between I_k in β-
                     graph labelled with weight
15:               end if
16:            end if
17:         end for
18:      end if
19:   end for
20: end for
```

Broadly, SemGrAM$_G$ re-examines the *nsi*'s in conjunction with information in the semantic graph with a view to forming new itemsets that will meet the normal support threshold. This is accomplished by constructing a β-graph for all *nsi*'s in which the itemsets are nodes and the edges indicate their similarity according to a family of graphs as shown in Figure 2 and as outlined in Algorithm 3.1. Note that there may be more than one edge between a pair of nodes if more than one family of graphs is applicable.

The function diff in Algorithm 3.1 operates in the same way as the *confusion matrix* of Oommen & Loke (1995), Oommen & Zhang (1996)[7] to examine two same-length itemsets returning the number of differences between them. dist(x, y, Γ_i) returns the semantic distance calculated (perhaps transitively) between the two nodes x and y in Γ_i. Following the construction of the β-graph, SemGrAM recursively searches the graph and combines the closest nodes.

In the current algorithm, once an itemset's support reaches σ it is removed from further merges (see Algorithm 3.2#13-15). This keeps the cohesion of the itemgroups as tight as possible. If these lines are omitted, the algorithm will produce more rules may have higher support at the expense of items with less semantic precision. Finally, rule production is relatively easy as given in Algorithm 3.3.

[6]It is possible that the β and γ thresholds could be merged (for example, the support for an itemset might be deprecated as the cohesion decreases) but how this is achieved is large application domain specific and we have chosen to retain the two independent thresholds.

[7]Oommen uses a confusion matrix to determine the probability of striking a wrong key on a keyboard, which can then be incorporated into an edit distance function.

Algorithm 3.2 SemGrAM$_G$ β-graph traversal

```
1: while edges left in β-graph do
2:    for each edge e_{j,k} between I_j between I_k in ascending order
      of weight do
3:       combine node I_new creating a new itemgroup containing
         x and y
4:       for each all other edges to I_other from I_j or I_k do
5:          weight = average of weights from I_other to I_j and I_other
            to I_k
6:          if weight ≤ γ then
7:             create new edge between I_new and I_other in β-graph
               labelled with weight
8:          end if
9:          delete edge between I_other and I_j and I_other and I_k
10:      end for
11:      supp(I_new) = supp(I_j) + supp(I_k) − supp(I_j ∩ I_k)
12:      if support(I_new) ≥ σ then
13:         Remove all edges connected to I_new
14:      end if
15:   end for
16: end while
```

Algorithm 3.3 Rule Production

```
1: extract rules from FP-tree as per (Han et al. 2000)
2: for each nodes I_j in β-graph do
3:    if support(I_j) ≥ σ then
4:       extract rule
5:    end if
6: end for
```

3.4 SemGrAM$_P$

This version of SemGrAM was written to investigate the utility of providing a more efficient algorithm at the expense of some semantic flexibility. SemGrAM$_P$ inspects the FP-Tree and merges branches that are between σ and β (ie those that would result in *nsi*'s. The effect of merging them in the FP-tree means that the SemGrAM$_P$ runs more efficiently (in terms of both time and space) that SemGrAM$_G$ as can be seen from the experimental results. The algorithm is in two parts - first the FP-tree is modified as outlined in Algorithm 3.3. Second, rules are produced as per (Han et al. 2000). The first part looks for sections of the FP-tree such that can be merged.

Algorithm 3.4 SemGrAM$_P$ FP-tree traversal

```
1: recursively search the FP-Tree
2: if
   (a) The weight of a node (in the context of what comes above
   it) is greater than σ
   (b) There are at least two children (n_1 ... n_i) of that node such
   that
        i. they are within the threshold semantic distance γ (ie as
   per Algorithm 3.1#7-17)
              AND
        ii. they have weights between σ and β then
3:    Create a new item representing the itemgroup [n_1 ... n_i]
4:    Merge subtrees of those children using the new item as root
5: end if
```

SemGrAM$_P$ is considerably simpler, both to code and execute but it should be noted that the same merging of subtrees may result in the same itemgroup being used in a number of rules.

4 Evaluation of Proof-of-Concept System

To demonstrate the concept, we implemented SemGrAM$_G$ and SemGrAM$_P$ (and for comparison, FPGrowth) in Java and ran experiments on a 1.5GHz Mac PowerPC G4. This implementation has shown it to be tractable and to reveal interesting rules that would otherwise not be reported. Note that the ability to dynamically create itemgroups means that

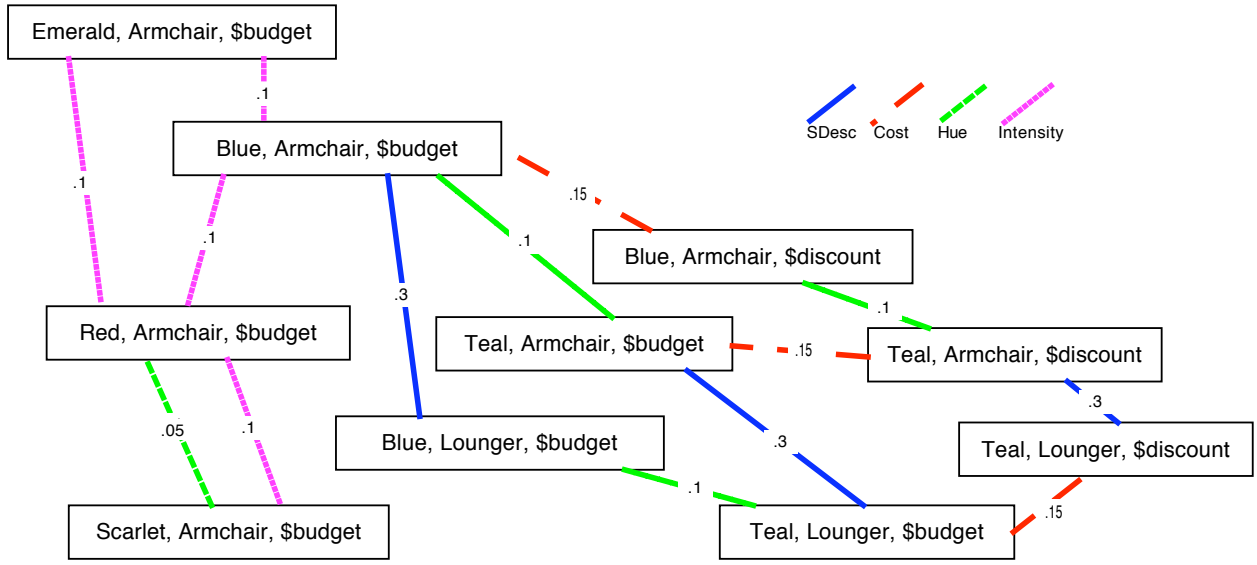


Figure 2: Example β-graph. Note that the different graphs for Hue and Intensity means that `< Red, Armchair, $budget >` cannot be put in the same itemgroup as `< Teal, Armchair, $budget >`.

items can be specified at a lower granularity (although not without affecting performance).

A 487Kb, 10,000 transaction synthetic dataset was constructed together with semantic graphs that covered 25%, 50%, 75% or 100% of the items listed. Results are shown in Figure 3. Two important points to note are that the premium for handling semantic graphs is currently up to 52% (although for a lower coverage and a larger dataset the premium is more reasonable at below 10%). This makes the concept tractable although further efficiencies may be useful. Secondly, the SemGrAM algorithms are linear in the time taken to process each itemset regardless of input file size. On all datasets tested, both SemGrAM$_G$ and SemGrAM$_P$ have shown that they scale satisfactorily. Moreover, as the dataset size increases, the cost per itemset reduces. Finally, the effects of changes to the *itemgroup cohesion threshold* (γ) can be large. As γ increases the interconnection between items accelerates showing the *double jump* phenomenon reported elsewhere (Spencer 2001).

Currently we limit SemGrAM$_G$ to looking for itemsets that vary by a single item. It can easily be seen that there would be cases where itemsets are similar but vary by more than a single item. In principle, the algorithm could be modified to assess the distances between a number of items in a set of itemsets and find the combination of items that creates the minimum distance. For example, the itemsets `< Blue, Armchair, $Budget >` and `< Teal, Lounger, $Budget >` in Figure 2 might be considered mergable. The issues arise in recording which items made the itemsets similar and making judgments at to whether the itemgroups created make sense semantically. There is, for example, a chance that inappropriate inferences may result.

5 Conclusions and Further Research

This paper has outlined two new algorithms for accommodating semantic graphs within association rule mining. In so doing it not only accommodates graph structured domains but also those for which a weighted, directed graph can be used to simulate other domain structures (such as lists and hierarchies). In some respects this work can thus be considered to subsume some earlier work in these areas.

The focus of the proof-of-concept implementation was not on performance but on proving that that the design decision were sound. Nevertheless, the implementation shows that even in this implementation, the premium paid for the extra processing is not excessively high, even in the case of SemGrAM$_G$. In practice, one of the major advantages will be that items can be specified at a lower granularity with the algorithm selecting the most appropriate aggregations.

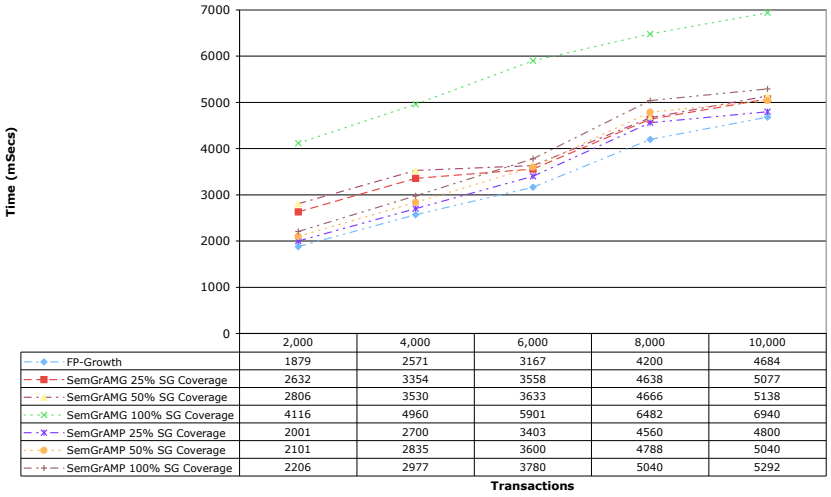
Further work can be envisaged for the algorithm, some of which is discussed in Section 4. In particular, the work of Nanavati et al. (2001) is complimentary to our work and it is possible that both ideas could be combined in a single algorithm.

The algorithms are currently dependent on the prior definition of the semantic graph. We have not yet investigated automatic generation of the graph or the use of functional (as opposed to enumerated) descriptions of graphs. However, there is pre-existing work in this area which could be utilised.

Since the overhead involved in the user's comprehension of the rules produced can significantly outweigh variations in processing time, it would potentially be useful to provide a good user-interface to the system allowing the full exploration of results through the semantic graph structure. For instance, if a result includes an itemgroup it could allow a one click look-up of the items that are a part of that itemgroup and show their contributions to the support of that itemset. It should be able to display the semantic graph clustering with varying levels of cohesion, possibly with varying clustering algorithms.

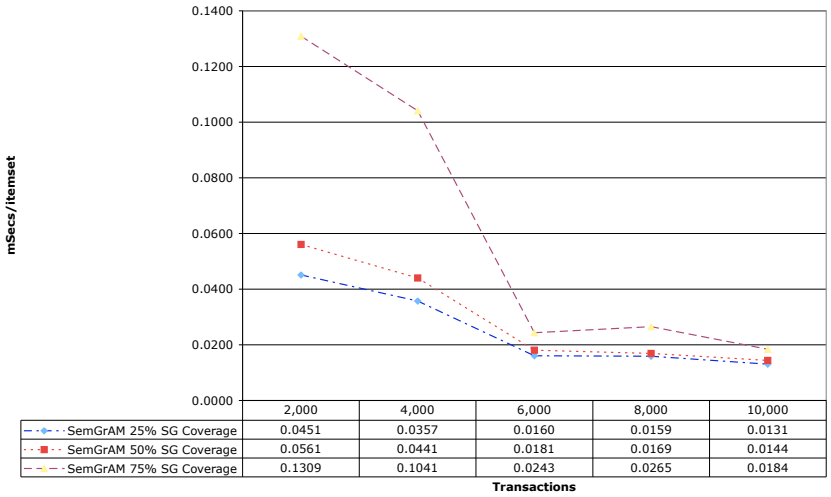
The system may also be applicable to the problem of clustering of association rules (cf. Lent et al. (1997)). Specifically, if one of the items is a spatial attribute, such as zip code, then we could potentially generate clusters of rules.

On a broader level, semantic graphs have not been accommodated in many areas of data mining to date and a wider program of research could be considered for which this research would be an example. In particular, sequential pattern mining (Agrawal & Srikant 1995, Srikant & Agrawal 1996) offers an opportunity for enhancement.



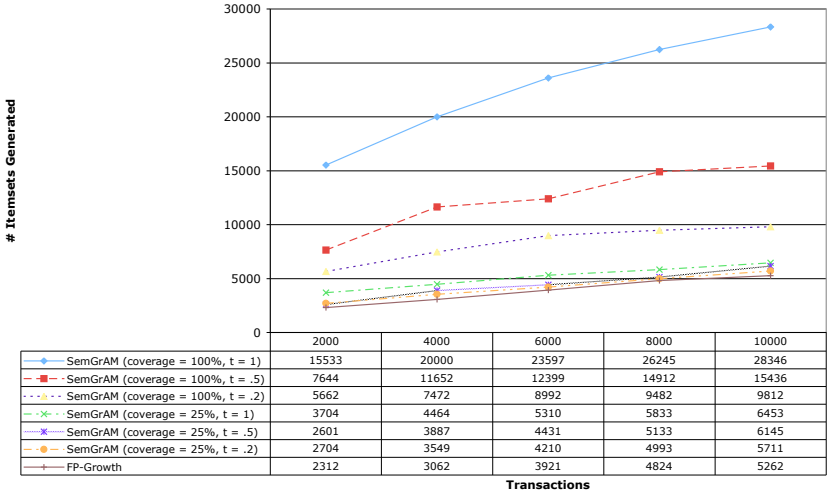
	2,000	4,000	6,000	8,000	10,000
FP-Growth	1879	2571	3167	4200	4684
SemGrAMG 25% SG Coverage	2632	3354	3558	4638	5077
SemGrAMG 50% SG Coverage	2806	3530	3633	4666	5138
SemGrAMG 100% SG Coverage	4116	4960	5901	6482	6940
SemGrAMP 25% SG Coverage	2001	2700	3403	4560	4800
SemGrAMP 50% SG Coverage	2101	2835	3600	4788	5040
SemGrAMP 100% SG Coverage	2206	2977	3780	5040	5292

(a) Effect of varying coverage of semantic graph



	2,000	4,000	6,000	8,000	10,000
SemGrAM 25% SG Coverage	0.0451	0.0357	0.0160	0.0159	0.0131
SemGrAM 50% SG Coverage	0.0561	0.0441	0.0181	0.0169	0.0144
SemGrAM 75% SG Coverage	0.1309	0.1041	0.0243	0.0265	0.0184

(b) Scalability of SemGrAM



	2000	4000	6000	8000	10000
SemGrAM (coverage = 100%, t = 1)	15533	20000	23597	26245	28346
SemGrAM (coverage = 100%, t = .5)	7644	11652	12399	14912	15436
SemGrAM (coverage = 100%, t = .2)	5662	7472	8992	9482	9812
SemGrAM (coverage = 25%, t = 1)	3704	4464	5310	5833	6453
SemGrAM (coverage = 25%, t = .5)	2601	3887	4431	5133	6145
SemGrAM (coverage = 25%, t = .2)	2704	3549	4210	4993	5711
FP-Growth	2312	3062	3921	4824	5262

(c) Effects of varying τ

Transaction file size	Distinct Itemsets	Av. time (mSecs)	μs/trans	μs/itemset
2,000	15,533	4,116	2.058	0.265
4,000	20,000	4,960	1.240	0.248
6,000	23,597	5,901	0.983	0.250
8,000	26,245	6,482	0.810	0.247
10,000	28,346	7,170	0.717	0.253

(d) Experimental Timing (SemGrAM$_G$)

Figure 3: Experimental Results

References

Agrawal, R., Imielinski, T. & Swami, A. (1993), Mining association rules between sets of items in large databases, *in* P. Buneman & S. Jajodia, eds, 'ACM SIGMOD International Conference on the Management of Data', ACM Press, Washington DC, USA, pp. 207–216.

Agrawal, R. & Srikant, R. (1995), Mining sequential patterns, *in* P. Yu & A. Chen, eds, '11th International Conference on Data Engineering (ICDE'95)', IEEE Computer Society Press, Taipei, Taiwan, pp. 3–14.

Budanitsky, A. & Hirst, G. (2000), Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures, *in* 'Workshop on WordNet and Other Lexical Resources, in the North American Chapter of the Association for Computational Linguistics (NAACL-2000)', Pittsburgh, PA, USA.

Ceglar, A. & Roddick, J. F. (2006), 'Association mining', *ACM Computing Surveys* **38**(2).

Ceglar, A., Roddick, J. F., Calder, P. & Rainsford, C. P. (2005), 'Visualising hierarchical associations', *Knowledge and Information Systems* **8**(3), 257–275.

Cheung, D., Han, J., Ng, V. & Wong, C. (1996), Maintenance of discovered association rules in large databases: an incremental updating technique, *in* S. Su, ed., '12th International Conference on Data Engineering (ICDE'96)', IEEE Computer Society, New Orleans, Louisiana, USA, pp. 106–114.

Ertöz, L., Steinbach, M. & Kumar, V. (2002), A new shared nearest neighbor clustering algorithm and its applications, *in* R. Grossman, J. Han, V. Kumar, H. Mannila & R. Motwani, eds, '2nd SIAM International Conference on Data Mining (SDM'02)', SIAM, Arlington, VA, USA.

Ertöz, L., Steinbach, M. & Kumar, V. (2003), Finding topics in collections of documents: A shared nearest neighbor approach, *in* W. Wu, H. Xiong & S. Shekhar, eds, 'Clustering and Information Retrieval 2003', Kluwer, pp. 83–104.

Fellbaum, C., ed. (1998), *WordNet: An Electronic Lexical Database*, Bradford Books.

Gray, B. & Orlowska, M. (1998), CCAIIA: Clustering categorical attributes into interesting association rules, *in* X. Wu, K. Ramamohanarao & K. Korb, eds, '2nd Pacific-Asia Conference on Research and Development in Knowledge Discovery and Data Mining (PAKDD-98)', Vol. 1394 of *LNAI*, Springer, Melbourne, Australia, pp. 132–143.

Guha, S., Rastogi, R. & Shim, K. (1999), Rock: A robust clustering algorithm for categorical attributes, *in* '15th International Conference on Data Engineering', IEEE Computer Society Press, Sydney, Australia, pp. 512–521.

Han, E.-H., Karypis, G., Kumar, V. & Mobashar, B. (1997), Clustering based on association rule hypergraphs, *in* 'Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'97)', Tucson, Arizona.

Han, J. & Fu, Y. (1999), 'Mining multiple-level association rules from large databases', *IEEE Transactions on Knowledge and Data Engineering* **11**(5), 798–804.

Han, J., Pei, J. & Yin, Y. (2000), Mining frequent patterns without candidate generation, *in* W. Chen, J. Naughton & P. Bernstein, eds, 'ACM SIGMOD International Conference on the Management of Data (SIGMOD 2000)', ACM Press, Dallas, TX, USA, pp. 1–12.

Jarmasz, M. (2003), Roget's Thesaurus as a Lexical Resource for Natural Language Processing, Masters, University of Ottawa.

Jarvis, R. A. & Patrick, E. A. (1973), 'Clustering using a similarity measure based on shared nearest neighbors', *IEEE Transactions on Computers* **C-22**(11).

Koperski, K. & Han, J. (1995), Discovery of spatial association rules in geographic information databases, *in* '4th International Symposium on Large Spatial Databases', Maine, pp. 47–66.

Kosters, W. A., Marchiori, E. & Oerlemans, Ard, A. J. (1999), Mining clusters with association rules, *in* D. Hand, J. Kok & M. Berthold, eds, '3rd International Symposium on Advances in Intelligent Data Analysis, IDA-99', Vol. 1642 of *LNCS*, Springer, Amsterdam, p. 39.

Kouris, I. N., Makris, C. & Tsakalidis, A. K. (2003), An improved algorithm for minign association rules using multiple support values, *in* I. Russell & S. Haller, eds, '16th Florida International Artificial Intelligence Research Society Conference', AAAI Press, St. Augustine, Florida, USA, pp. 309–313.

Kuok, C., Fu, A. & Wong, M. H. (1998), 'Mining fuzzy association rules in databases', *ACM SIGMOD Record* **27**(1), 41–46.

Lee, Y.-C., Hong, T.-P. & Lin, W.-Y. (2005), 'Mining association rules with multiple minimum supoports using maximum constraints', *International Journal of Approximate Reasoning* **40**(1-2), 44–54.

Lent, B., Swami, A. & Widom, J. (1997), Clustering association rules, *in* A. Gray & P.-A. Larson, eds, '13th International Conference on Data Engineering', IEEE Computer Society Press, Birmingham, UK, pp. 220–231.

Lu, Y. (1997), Concept Hierarchy in Data Mining: Specification, Generation and Implementation, Master of science, Simon Fraser University.

Mazlack, L. & Coppock, S. (2002), Granulating data on non-scalar attribute values, *in* 'IEEE International Conference on Fuzzy Systems', Honolulu, pp. 944–949.

Mooney, C. H., De Vries, D. & Roddick, J. F. (2006), A multi-level framework for the analysis of sequential data, *in* S. Simoff & G. Williams, eds, 'Data Mining: Theory, Methodology, Techniques, and Applications', Vol. 3755 of *LNAI*, Springer, Heidelberg, Germany, pp. 229–243.

Nanavati, A., Chitrapura, K. P., Joshi, S. & Krishnapuram, R. (2001), Mining generalised disjunctive association rules, *in* '10th International Conference on Information and Knowledge Management (CIKM'01)', ACM Press, Atlanta, Georgia, USA, pp. 482–489.

Ong, K. L., Ng, W. K. & Lim, E. P. (2001), Large mining multi-level rules with recurrent items using fp-tree, *in* '3rd IEEE Conference on Information, Communications and Signal Processing (ICICS'2001)', Springer, Singapore.

Oommen, B. J. & Loke, R. K. S. (1995), Pattern recognition of strings with substitutions, insertions, deletions and generalized transpositions, *in* 'IEEE International Conference on Systems, Man and Cybernetics', Vol. 2, IEEE, pp. 1154–1159.

Oommen, B. J. & Zhang, K. (1996), 'The normalized string editing problem revisited', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**(6), 669–672.

Openshaw, S. (1983), *The Modifiable Areal Unit Problem (Concepts and Techniques in Modern Geography)*, Geo Books, Norwich, UK.

Rainsford, C., Mohania, M. & Roddick, J. F. (1997), A temporal windowing approach to the incremental maintenance of association rules, *in* J. Fong, ed., '8th International Database Workshop, Data Mining, Data Warehousing and Client/Server Databases (IDW'97)', Springer, Hong Kong, pp. 78–94.

Roddick, J. F., Hornsby, K. & De Vries, D. (2003), A unifying semantic distance model for determining the similarity of attribute values, *in* M. Oudshoorn, ed., '26th Australasian Computer Science Conference (ACSC2003)', Vol. 16 of *CRPIT*, ACS, Adelaide, Australia, pp. 111–118.

Roddick, J. F. & Spiliopoulou, M. (2002), 'A survey of temporal knowledge discovery paradigms and methods', *IEEE Transactions on Knowledge and Data Engineering* **14**(4), 750–767.

Shen, L. & Shen, H. (1998), Mining flexible multiple-level association rules in all concept hierarchies, *in* G. Quirchmayr, E. Schweighofer & T. Bench-Capon, eds, '9th International Conference on Database and Expert Systems Applications, DEXA'98', Vol. 1460 of *LNCS*, Springer, Vienna, Austria, pp. 786–795.

Spencer, J. (2001), *The Strange Logic of Random Graphs*, Springer.

Srikant, R. & Agrawal, R. (1996), Mining sequential patterns: generalisations and performance improvements, *in* P. M. G. Apers, M. Bouzeghoub & G. Gardarin, eds, 'International Conference on Extending Database Technology, EDBT'96', Vol. 1057 of *LNCS*, Springer, Avignon, France, pp. 3–17.

Srikant, R. & Agrawal, R. (1997), 'Mining generalized association rules', *Future Generation Computer Systems* **13**(2-3), 161–180.

Suk, C.-Y. & Park, E. (1999), An approach to intensional query answering at multiple abstraction levels using data mining approaches, *in* '32nd Annual Hawaii International Conference on Systems Sciences, HICSS-32', IEEE Comput. Soc, Los Alamitos, CA, USA.

Zaki, M. J., Parthasarathy, S., Ogihara, M. & Li, W. (1997), New algorithms for fast discovery of association rules, *in* '3rd International Conference on Knowledge Discovery and Data Mining (KDD-97)', AAAI Press, Newport Beach, CA, USA, pp. 283–286.

Are Zero-suppressed Binary Decision Diagrams Good for Mining Frequent Patterns in High Dimensional Datasets?

Elsa Loekito and James Bailey

NICTA Victoria Laboratory
Department of Computer Science and Software Engineering
University of Melbourne, Australia
Email: {eloekito, jbailey}@csse.unimelb.edu.au

Abstract

Mining frequent patterns such as frequent itemsets is a core operation in many important data mining tasks, such as in association rule mining. Mining frequent itemsets in high-dimensional datasets is challenging, since the search space is exponential in the number of dimensions and the volume of patterns can be huge. Many of the state-of-the-art techniques rely upon the use of prefix trees (e.g. FP-trees) which allow nodes to be shared among common prefix paths. However, the scalability of such techniques may be limited when handling high dimensional datasets. The purpose of this paper is to analyse the behaviour of mining frequent itemsets when instead of a tree data structure, a canonical directed acyclic graph namely Zero Suppressed Binary Decision Diagram (ZBDD) is used. Due to its compactness and ability to promote node reuse, ZBDD has proven very effective in other areas of computer science, such as boolean SAT solvers. In this paper, we show how ZBDDs can be used to mine frequent itemsets (and their common varieties). We also introduce a weighted variant of ZBDD which allows a more efficient mining algorithm to be developed. We provide an experimental study concentrating on high dimensional biological datasets, and identify indicative situations where a ZBDD technology can be superior over the prefix tree based technique.

Keywords: data mining, association rule mining, frequent patterns, frequent itemset mining, Binary Decision Diagrams (BDDs), Zero-suppressed Binary Decision Diagrams (ZBDDs), high dimensional datasets.

1 Introduction

Mining frequent patterns such as frequent itemsets is a fundamental and well studied problem in data mining. It has a number of useful applications such as association rule mining and market basket data analysis. Frequent itemsets correspond to combinations of items (or attribute values) which occur frequently in the dataset. Thus, mining them in a high dimensional dataset can be challenging, since the search space is exponential in the number of dimensions.

State-of-the-art frequent itemset mining techniques such as those found in FIMI Repository (Goethals 2004) have made attempts to address this issue by making use of prefix tree data structures, or combinations of prefix trees with other data structures, to compress the data representation. Prefix trees, however, limit node sharing to common prefixes which may limit the scalability of a frequent itemset mining algorithm when handling high dimensional datasets. As we will analyse in this paper, sharing of common suffixes, too, can be useful for mining frequent itemsets, which is made possible using a graph data structure called the Zero-suppressed Binary Decision Diagrams (ZBDDs) (Minato 1993).

In particular, microarray datasets are one of the most challenging kinds of high dimensional datasets for pattern mining. They typically consist of only a few number of samples (i.e. rows), but they have very high dimensionality (i.e. thousands of attributes). The number of patterns in a microarray dataset can be enormous (Creighton & Hanash 2003), and hence, mining them requires considerable time as well as space. Other works such as (Rioult et al. 2003, Li et al. 2003) have also studied other itemset mining problems in microarray datasets.

Binary Decision Diagrams (BDDs) (Bryant 1986) are a compact canonical graph representation of boolean formulae. They are a directed acyclic graph (DAG) data representation, similar to binary decision trees, except identical sub-trees are merged. There exist efficient BDD library routines which promote their canonicity and allow intermediate computation results to be reused. Furthermore, Zero-suppressed Binary Decision Diagrams (ZBDDs) are a special type of BDDs which were introduced for efficient manipulation of sparse item combinations (Minato 2001, Minato & Arimura 2005). There exist efficient library routines for manipulating ZBDDs in (Mishchenko 2001), and they have been shown effective for solving problems in other computer science areas such as boolean SAT solvers (Aloul et al. 2002) and solving graph optimization problems (Coudert 1997). There are only a few works that use ZBDDs in a data mining context, such as (Loekito & Bailey 2006, Minato 2005, Minato & Ito 2007, Minato & Arimura 2006).

A vast number of techniques for mining frequent itemsets and their common varieties have been proposed. A survey can be found in (Zaki & Goethals 2003, Zaki et al. 2004). FP-growth* (Grahne & Zhu 2003) is one of the strongest frequent itemset mining algorithms, which is based on the use of prefix trees such as FP (frequent pattern)-trees. FP-growth* follows a pattern growth framework (Han et al. 2004) which recursively creates database projections, and it uses FP-trees to represent the intermediate databases. Other implementations of pattern growth such as AFOPT (Liu et al. 2003) and LCMv3 (Uno et al. 2005) are also based on the use of modified FP-trees, or prefix trees combined with arrays and bitmap data structures.

The purpose of this paper is to analyse the behaviour of frequent itemset mining when canonical DAGs such as ZBDDs, instead of trees, are used as a

primary data structure. Prefix trees have been used as a means for achieving data compression through allowing node sharing of common prefixes. ZBDDs are different from prefix trees in the sense that node fan in as well as fan out is allowed, and multiple identical sub-trees do not exist. More specifically, by sharing the identical sub-trees in a ZBDD, higher data compression can potentially be achieved and efficient ZBDD library can be employed in the mining procedure. Therefore, ZBDDs are seemingly attractive for mining frequent itemsets in sparse, and challenging, high dimensional datasets.

In this paper, we show how ZBDDs can be used to mine frequent itemsets and their maximal and closed variants, concentrating on high dimensional biological datasets. We consider mining in a column-wise pattern growth framework, and a row-wise framework (Rioult et al. 2003, Pan et al. 2004) (introduced for mining closed frequent itemsets). Our objective is to identify and explain situations where ZBDDs are advantageous compared to FP-trees. In particular, we aim to address questions such as:

1. Does the canonical property of ZBDDs allow a scalable and efficient algorithm for frequent itemset mining to be developed ?
2. How much data compression can a ZBDD achieve compared to an FP-tree ?
3. Does the use of a more compact data structure always mean that mining is more efficient?

Our main contributions in this paper are threefold:

- We present an algorithm that can mine (maximal/closed) frequent itemsets, based on the use of a ZBDD as the primary data structure and a supplementary bitmap (similar to (Burdick et al. 2001)) for support checking. A particularly attractive feature of our technique is the use of multiple shared-ZBDDs to represent the input database, the intermediate databases, as well as the final output, allowing them to share common sub-trees. This is something which is not possible in prefix-tree-based techniques like (Grahne & Zhu 2003, Liu et al. 2003, Pietracaprina & Zandolin 2003). We also show how the ZBDD mining framework can be adapted to the row-wise mining approach (Pan et al. 2003, 2004).
- We introduce an edge-weighted variant of ZBDDs, whose structure is similar to Edge-Valued Binary Decision Diagrams (Vrudhula et al. 1996) which were proposed for efficient representation of discrete functions. Our weighted ZBDDs allow itemsets and their corresponding frequencies to be compactly represented. Hence, support counting can be performed more efficiently compared to when the bitmap is used. Moreover, their canonical property allows a more efficient mining technique to be developed, which is achieved through re-using intermediate results. It is advantageous especially for mining large and dense datasets, in which bitmap manipulations may be costly, and a significant portion of the intermediate computations may share results.
- We experimentally investigate the behaviour of our techniques, according to various characteristics of high dimensional biological datasets. Our techniques are compared against the state-of-the-art FP-tree-based technique FP-growth* (Grahne & Zhu 2003). Our results show a number of situations where the use of a ZBDD (either weighted or not) is able to give improvement over FP-growth*.

2 Preliminaries

In this section we provide background knowledge of the pattern growth framework, which is employed by the existing prefix-tree based mining algorithms, and an overview of Zero-suppressed Binary Decision Diagrams. Firstly though, we define some terminologies which will be used in the remainder of this paper.

Assume we have a dataset D defined upon a set of k attributes. For every attribute A_i, $i \in \{1, 2..k\}$, the domain of its values (or items) is denoted by $dom(A_i)$. Let $\mathcal{I}$ be the aggregate of the domains items across all the attributes, i.e. $\mathcal{I} = \bigcup_{i=1}^{k} dom(A_i)$. An itemset is a subset of $\mathcal{I}$. Let p and q be two itemsets. We say p contains q if q is a subset of p, i.e. $q \subseteq p$. A dataset is a collection of transactions, where each transaction is an itemset. The support of an itemset p in dataset $\mathcal{D}$, i.e. $support(p)$, is the fraction of the transactions in $\mathcal{D}$ which contain p ($0 \leq support(p) \leq 1$). Given a dataset $\mathcal{D}$, and a support threshold α, an itemset p is **frequent** if it satisfies the constraint: $support(p) \geq \alpha$. Furthermore, p is a **maximal frequent itemset** if p is not contained in any other frequent itemset. p is a **closed frequent itemset** if p is not contained in any other frequent itemset which has the same support.

2.1 Pattern growth framework for mining frequent itemsets

The pattern growth framework for mining frequent patterns grows prefixes by recursively projecting intermediate databases. For each item x, an *x-conditional DB* is induced. It contains itemsets in the input database which contain x. Then, frequent itemsets which contain prefix $\{x\}$ are grown by recursively creating further projections from the x-conditional DB. FP-growth is one of the strongest techniques that follows this pattern growth approach. In particular, FP-growth (Han et al. 2004) uses frequent pattern (FP) trees for storing the input, the output patterns, and the conditional DBs. An FP-tree is created afresh for each of those databases.

FP-growth uses a dynamic ordering of the items, such that items in an FP-tree are ordered by decreasing frequency, the most frequent item at the top. The FP-trees which are created throughout mining may use different item orderings. One of the most efficient implementations of FP-growth grows prefixes by traversing the database in a bottom-up manner (There exist its variations which use an inverse item ordering and perform a top-down traversal, such as that in (Liu et al. 2003)).

An FP-tree example is shown in Figure 1(a). Each node in an FP-tree contains an item and a value which represents the frequency of that node's prefix path. As a secondary data structure, a header table is used for storing the total frequency of each item. In this example, the first prefix is grown using item d (the least frequent item). The FP-tree representation for d-conditional DB is shown in Figure 1(b).

FP-growth* (Grahne & Zhu 2003) is an optimised implementation of FP-growth which reduces the number of database projections based on a technique called *bi-level projection*, and minimises memory usage of the algorithm by using a *pseudo-projection* instead of physically creating the conditional databases. For further implementation details, readers are referred to that paper.

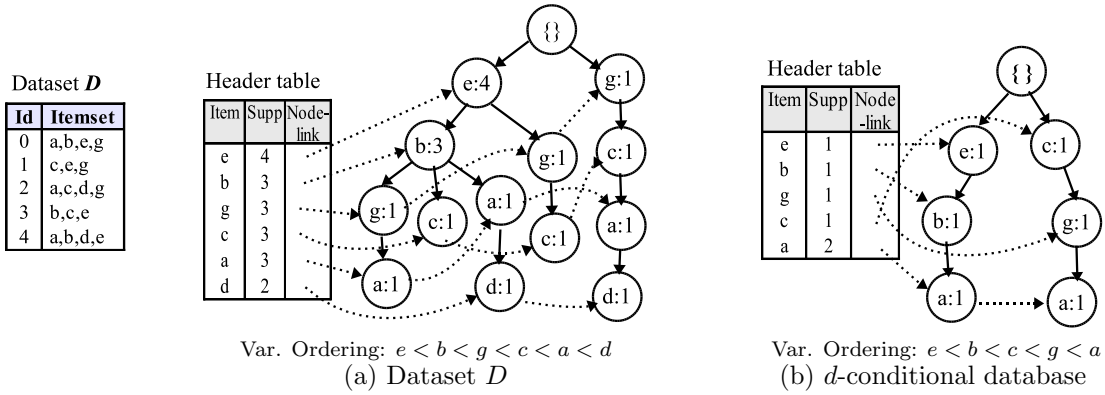


Figure 1: FP-tree representations for sample dataset D

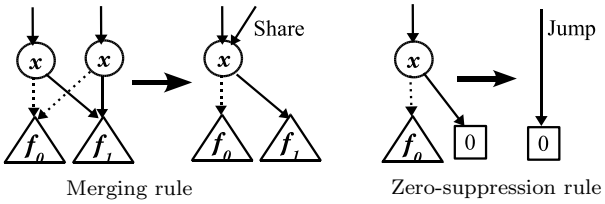


Figure 2: ZBDD Reduction Rules

2.2 Zero-suppressed Binary Decision Diagram(ZBDD):

Binary Decision Diagrams (BDDs) (Bryant 1986) are canonical directed acyclic graphs (DAGs). Their canonical property allows boolean formulae to be compactly represented and their logical operations (AND, OR, XOR, etc.) to be performed in polynomial time with respect to the number of nodes. A Zero-suppressed BDD (ZBDD) (Minato 1993) is a special type of BDD which was introduced for efficient manipulation of sparse combinations. However, it has received little attention in data mining except a few works in (Loekito & Bailey 2006, Minato 2005, Minato & Ito 2007, Minato & Arimura 2006). A survey on other, non data-mining, ZBDD applications can be found in (Minato 2001).

More formally, a BDD is a canonical DAG of labeled nodes. It consists of one source node, multiple internal nodes, and two *sink* nodes which are labeled as 0 and 1 respectively. Each internal node may have multiple parent nodes, but it can only have two child nodes, which we call 0-*child* and 1-*child* nodes. By canonical, it means that multiple identical nodes are not allowed. Two nodes are identical if they have the same label, and their respective child nodes are also identical. The nodes are ordered so that the label of any internal node must be of higher index (i.e. appear earlier in the variable ordering) than the label of its children.

An internal node N with a label x, denoted $N = node(x, N_1, N_0)$, encodes the boolean formula $N = (x \wedge N_1) \vee (\overline{x} \wedge N_0)$. We call N_1 (resp. N_0) the 1-child (resp. 0-child) of N. The edge connecting a node to its 1-child (resp. 0-child) is also called the *true*-edge (resp. *false*-edge). In the illustrations shown shortly, solid lines correspond to true-edges whilst dotted lines correspond to false-edges. Each path from the root to sink-1 (resp. sink-0) gives a true (resp. false) assignment for the represented function. For some node N, with label x, the outgoing true-edge of N represents a true assignment for variable x, whereas the outgoing false-edge represents a false assignment for variable x.

BDDs have two important properties (Bryant 1986):

1. Equivalent subtrees are shared (*canonical*);
2. Computation results are stored for future reuse (referred as BDD's *caching principle*).

These properties make the worst-case complexity of most BDD operations polynomial with respect to the number of nodes. The caching principle allows the result from intermediate computations to be reused if the same computation is re-visited in the future. This utility is particularly effective if many subtrees are being shared within the input BDD, as the same subtree may be encountered multiple times (under the same computation) through out its manipulation.

A Zero-suppressed BDD (ZBDD) is a special kind of BDD which was introduced for efficient combinatorial itemset analysis (Minato 1993, Minato & Arimura 2005). More specifically, a ZBDD is a BDD with two reduction rules (Their illustrations are shown in Figure 2):

1. **Merging rule**: merge identical subtrees (to obtain canonicity);
2. **Zero-suppression rule**: delete nodes whose 1-child is the sink-0, and replace them with their 0-child.

By utilising these rules, a sparse collection of item combinations, which can be seen as a boolean formula, can be represented with high compression, i.e. for an n variable formula, the possible space of truth values is 2^n, however the corresponding BDD/ZBDD can have exponentially fewer nodes.

We follow the ZBDD encodings for representing a collection of itemsets as in (Minato & Arimura 2005). An itemset p can be represented by a n-bit binary vector $(x_1, x_2, \ldots, x_n)$, where $x_i = 1$ if item i is contained in p. A set $\mathbf{S}$ of itemsets can be represented by a characteristic function $\mathcal{X}_S : \{0,1\}^n \rightarrow \{0,1\}$ where $\mathcal{X}_S(p) = 1$ if $p \in \mathbf{S}$ and 0 otherwise. In ZBDD semantics, set $\mathbf{S}$ such that $\mathbf{S} = \mathbf{S_0} \cup (\mathbf{S_1} \times \{x\})$ can be represented by a ZBDD node $N = (x, N_1, N_0)$ where $\mathbf{S_1}$ (resp. $\mathbf{S_0}$) is the set of itemsets encoded by N_1 (resp. N_0). An itemset p in S is interpreted as a conjunction of the items contained in p and yields a true assignment for the boolean formula encoded by N. A sink-0 node encodes an empty set ($\emptyset$), and sink-1 node encodes a set of an empty itemset ($\{\emptyset\}$). For clarity, sink-0 nodes may be omitted from the illustrations shown in this paper. Table 1 lists some pre-defined ZBDD library operations (Minato & Arimura 2005, Mishchenko 2001) which we use in our algorithms.

Variable Ordering: The number of nodes in a ZBDD, and thus, the efficiency of its manipulations may be highly sensitive to its variable ordering. Work in (Minato 2001) shows that a good variable ordering for compact BDDs should have two properties:

Table 1: Primitive ZBDD operations

sink-0	The empty set, $\emptyset$
sink-1	The set of an empty itemset, $\{\emptyset\}$
getNode(x, N_1, N_0)	Create $node(x, N_1, N_0)$ and apply ZBDD reduction rules
$P \bigcup_{max} Q$	Maximal set-union between itemsets in P and Q
$P \setminus Q$	Itemsets of P which do not exist in Q
CrossProd(P, Q)	Pair-wise intersection between itemsets in P and itemsets in Q

E.g.: 1. $P = \{\{a, b, d\}, \{b, c\}\}$, $Q = \{\{b, c, d\}, \{a, c, d\}\}$, $P \bigcup_{max} Q = \{\{a, b, d\}, \{b, c, d\}, \{a, c, d\}\}$
2. $P = \{\{b, c, d\}, \{a, b, d\}\}$, $Q = \{\{a, b, d\}, \{a, c\}\}$, $P \setminus Q = \{\{b, c, d\}\}$
3. $P = \{\{a, d\}, \{b, c\}\}$, $Q = \{\{b, d\}, \{a, b\}\}$, CrossProd$(P, Q) = \{\{a, b, d\}, \{b, c, d\}, \{a, b, c, d\}\}$

1) groups of closely related variables should be kept near to each other; 2) variables that greatly affect the function should be located at higher positions. For our purpose, we use some heuristics described shortly, based on the frequency of the variables in the input dataset, that aims to maximise sharing of sub-structures across the auxiliary ZBDDs and allow efficient mining.

3 Frequent Itemset Mining Based on the Use of ZBDDs

In this section we present our ZBDD-based mining techniques for mining frequent itemsets, and mining their maximal and closed variants.

As a general overview, our techniques adopt the pattern growth framework, but instead of using FP-trees, we use ZBDDs as a primary data structure, and ZBDD library routines are used to compute the conditional DBs. More specifically, we use multiple-shared ZBDDs, which means canonicity is maintained (i.e. node sharing is allowed) across the multiple databases. Unlike in FP-trees, support information is not stored inside the nodes, instead, we use a secondary data structure *bitmap* for support counting.

The core operations in our framework, such as creating database projections and maintaining the output patterns, employ efficient ZBDD library functions which cache computation results. This means, intermediate redundant computations can be avoided if the same database (or substructure of the database) is re-visited through out mining. Due to the recursive nature of the pattern growth algorithm, multiple conditional databases are likely to contain many similar sub-structures. In particular, for a given prefix, its conditional DB contains subsets of itemsets from the conditional DB projected by some subset of that prefix.

Variable Ordering: Items in the ZBDDs in our mining technique are ordered by increasing frequency. This ordering has the following three objectives: 1) to obtain smaller conditional DBs in the first recursions, 2) to allow early pruning of infrequent prefixes, 3) to increase node-sharing across the databases. This ordering allows a very high data compression to be achieved, and increases the effectiveness of ZBDD's caching principle during database projections. Figure 3(a) shows the ZBDD which represents the input dataset $\mathcal{D}$ given in Figure 1. In this example, the ZBDD contains only 10 nodes (excluding sink nodes), whereas the FP-tree contains 14 nodes.

3.1 Frequent Itemset (FI) Miner

Our algorithm for mining frequent itemsets is labeled as **FIMiner**. The algorithm is shown in Algorithm 1. It is invoked by calling FIMiner(Z_D, α, $prefix$) where Z_D contains the input itemsets and $prefix$ is initially empty (i.e. $prefix = \{\}$). Frequent itemsets are grown from the given prefix, using the input ZBDD which is traversed in a top-down fashion. Let x be the top-node's label. For a given input ZBDD, item x is firstly used to grow the current prefix since no computation is needed to create the conditional DB for this item. More specifically, an x-conditional DB can be found in the top 1-child node, which contains all itemsets containing x in Z_D. This routine finds the patterns which contain x. Patterns which do not contain x are grown later, after removing x from the input Z_D and applying the same mining procedure. The output ZBDD is incrementally built from each recursion step, using the same variable ordering as the input ZBDD which allows both ZBDDs to share nodes.

In addition to the standard ZBDD primitive operations, we push the anti-monotonic support constraint deep inside the routine using an *infrequent prefix pruning* strategy (line 4-5) which is based on the anti-monotonic property of *support*. Here, the *bitmap* data representation of the input database is needed to calculate the support of each prefix. For a given prefix P, bitmap(P) refers to the bit-vector of the occurrence of P in the initial input dataset. We compute support(P) by counting the number of 1's in bitmap(P), denoted as |bitmap(P)|. Hence, the support of $prefix \cup \{x\}$ in FIMiner (line 4) can be computed incrementally by re-using bitmap($prefix$) which has been computed in previous mining iteration, and taking its bit-wise intersection with bitmap($\{x\}$).

When a new prefix which is grown using item x is infrequent, it is deleted from the output by returning the sink-0 node and employing ZBDD's zero-suppression rule (line 5). This automatically deletes node x from the output and replaces it by other patterns which do not contain item x, which will be computed later in line 9. Otherwise, the new prefix can be grown further using x-conditional DB (line 7). To remove x from the remaining routines and grow prefixes using the remaining items, the two child-nodes of Z_D are merged using a set-union operation, which is a ZBDD primitive operation (line 9). For mining efficiency and data compression purposes, the non-maximal itemsets are removed from the merged database, which can be obtained by using ZBDD's $\bigcup_{max}$ operation. We refer to this operation as *DB-merging*. Since primitive ZBDD operations store computation results in a cache, DB-merging can be computed efficiently if many of the conditional DBs share common subtrees. Finally, the recursion terminates when the induced database is empty (line 1-2).

Once mining in both the x-conditional database and the merged database have been completed, the output node is created by having item x as its label and the patterns found from the two sub-tasks are assigned to its 1-child and 0-child respectively. Since, this procedure is performed at each recursion level, it incrementally builds the output ZBDD in a bottom-up fashion. More specifically, the frequent patterns which are found from the x-conditional database be-

Algorithm 1 FIMiner(Z_D,α, $prefix$)

Input: Z_D: the database induced by $prefix$
α: minimum support threshold ,
$prefix$: prefix itemset
Output: Z_{FI}: the frequent itemsets in Z_D
Procedure:
1: **if** (Z_D is a sink node) **then**
2: *Terminal case:*
return Z_D
3: **end if**
Let $Z_D = node(x, Z_{D_x}, Z_{D_{\overline{x}}})$,
$prefix_x = prefix \cup \{x\}$
4: **if** (support($prefix_x$) $< \alpha$) **then**
5: *Infrequent prefix pruning*:
$Z_{FI_x} = 0$
6: **else**
7: *Grow new prefix* $prefix_x$ *and mine FIs*:
Z_{FI_x} = FIMiner(Z_{D_x}, α, $prefix_x$)
8: **end if**
9: *DB-merging and grow prefix without* x:
$Z_{FI_{\overline{x}}}$ = FIMiner($Z_{D_{\overline{x}}} \bigcup_{max} Z_{D_x}$, α, $prefix$)
10: **return** Z_{FI} = getNode($x, Z_{FI_x}, Z_{FI_{\overline{x}}}$)

Note:
support($prefix_x$) = |bitmap($prefix$) $\cap$ bitmap($\{x\}$)|.

come the 1-child node, which are seen as patterns which contain item x, and the frequent patterns which are found in the merged database become the 0-child. When creating such a node, ZBDD's library checks whether the node has been existed. If it has, then the existing node is shared.

Let us consider again the sample dataset in Figure 1. Figure 3(a) shows the conditional DB for the first item, i.e. *d*-conditional DB which is given by the 1-child of the top-node. Figure 3(b) illustrates the *DB-merging* operation. The merged database contains the set-union between itemsets in the two child-nodes of Z_D. In order to achieve higher data compression and mining efficiency, the non-maximal itemsets are simultaneously removed while computing the set-union. Identical nodes in the merged ZBDD are shared with the input database, as well as the other databases (nodes which are newly created for the merged ZBDD are drawn with a bold outline).

3.2 Maximal Frequent Itemset (MFI) Miner

We now describe some optimisations that can be applied to our FI mining technique for mining maximal frequent itemsets (MFIs). We call the algorithm **MFI-Miner**. Using the same core operations as FI-Miner, MFI-Miner has an additional procedure for removing the non-maximal patterns. This is performed using a progressive focusing technique (Burdick et al. 2001), which removes the locally non-maximal patterns from each conditional DB. It can be computed using a ZBDD primitive set-subtraction routine i.e. $Z_{FI_{\overline{x}}} \setminus Z_{FI_x}$, where Z_{FI_x} and $Z_{FI_{\overline{x}}}$ are computed as in Algorithm 1. This subtraction operation removes the frequent extensions of $prefix$ which also occur as frequent extensions of $prefix_x$ since they are non-maximal local to the current database.

Additionally, our framework can adopt some advanced pruning techniques such as those used in (Burdick et al. 2001, Grahne & Zhu 2003, Wang et al. 2003). For this purpose, an itemset *tail* is maintained for each conditional DB. It contains the items that occur in the relevant database, and ZBDD library functions are employed for manipulating each database with its *tail*. For instance, to remove infrequent items from a database D, *crossProd* can be employed upon the ZBDDs of D and $freqTail$, where $freqTail$ is obtained from *tail* by removing the infrequent items. Since $freqTail$ is a single itemset, then the *crossProd* operation returns the intersection between each itemset in D with $freqTail$.

3.3 Closed Frequent Itemset (CFI)-Miner

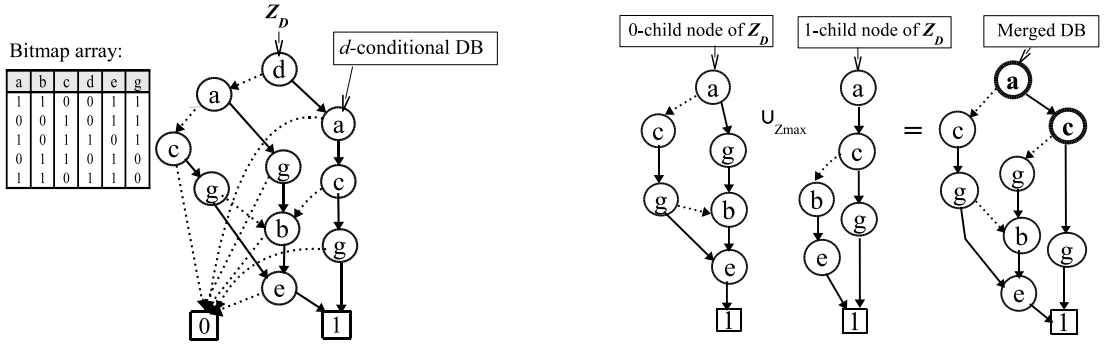
Our algorithm for mining closed frequent itemsets namely **CFI-Miner** has a similar framework as MFI-Miner, which uses a progressive focusing technique for removing the non-closed patterns. However, the closed constraint requires the support of an itemset to be compared against its subset(s). Thus, the support information has to be represented in the output data structure, which was not necessary for FI/MFI-Miner. Note that the support information does not need to be represented in the input ZBDD as it may reduce its compactness. Additionally, CFI-Miner can also adopt the more advanced pruning techniques found in existing algorithms, using a similar mechanism to MFI-Miner which maintains a *tail* itemset.

To represent the patterns' support in the ZBDD output, additional variables are used, which are appended to each pattern. We refer to these extended pattern representations as *item-support-sets*. In order to achieve higher compression, we use the binary representation of the support values. For a database of n transactions, $log_2(n)$ binary variables are reserved. Suppose the database containing 5 transactions, 3 support-encoding variables are reserved. Let r_0 r_1 r_2 be the support-encoding binary variables, such that $r_2 = 0$, $r_1 = 0$, $r_0 = 1$ represents a support of 1, $r_2 = 0$, $r_1 = 1$, $r_0 = 0$ represents 2, etc. For instance, the *item-support-set* representation of itemsets (with their corresponding frequencies) {*be:3, abe:2, ab:2*} is $\{ber_1r_0, aber_1, abr_1\}$. Furthermore, itemsets in the maximal item-support-sets correspond to closed itemsets. Item-support-set abr_1 is not maximal since $aber_1$ exists. However, $aber_1$ is maximal, and *abe* is a closed itemset.

4 Weighted Zero-suppressed Binary Decision Diagrams

Support counting using bitmap in our above-described techniques can be costly, especially in dense high dimensional datasets which contain long patterns. To eliminate this overhead, we introduce a weighted variant of ZBDD, namely **Weighted Zero-suppressed Binary Decision Diagram (WZDD)** which allows the support values to be represented using edge-weights and in turn allows a more efficient frequent itemset mining. There exist other weighted types of Decision Diagrams (Bryant & Chen 1995, Vrudhula et al. 1996, Ossowski & Baier 2006) for manipulating pseudo-boolean functions, but the semantics are different.

In a WZDD, every edge is attributed by a positive integer value. Formally, we define an internal WZDD node as a pair of $\langle \varphi, \vartheta \rangle$, where φ is the total weight of this node's outgoing edges, ϑ is a ZBDD node. The edge-weights are anti-monotonic, with their values descreasing as the nodes are positioned lower in the structure. The weight on its incoming edge corresponds to the total support of itemsets being represented(see Figure 4(a)). WZDDs are also canonical, that is, nodes which contain the same set of itemsets with the same corresponding supports are merged. Consequently, ZBDD's set-union routine needs to be adapted for WZDDs to add the supports of any itemset which occurs in both of its operands. Figure 4(b) shows an example of a WZDD which represents itemsets (with their corresponding support) : {*ace:1, abe:2, ab:2*}. The weight on node *a*'s incoming

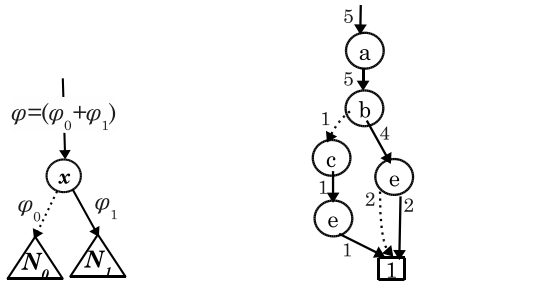


Var. ordering: $d < a < c < g < b < e$

(a) Z_D = ZBDD representation for dataset $\mathcal{D}$, d-conditional DB is Z_D's 1-child node

(b) DB merging (nodes marked with bold lines are the newly created nodes)

Figure 3: ZBDD representations for sample dataset $\mathcal{D}$



WZDD node

(a)

Var. ordering: $a < b < c < d < e$

WZDD representation for $\{abe : 2, ace : 2, ab : 1\}$

(b)

Figure 4: Weighted ZBDDs

link is 5, i.e. the total support of the itemsets. The two node e's are not shared since their edges have different weights, respectively.

Although WZDDs may use more nodes than the ZBDDs, due to their weighted-edges, they allow further use of the caching utility for mining frequent itemsets. More specifically, WZDDs allow more efficient mining algorithm to be developed by caching the computed patterns from each conditional DB. In WZDDs, any two identical nodes contain the same set of itemsets and their corresponding support, thus, the two nodes also contain the same set of frequent patterns. Suppose different prefixes project the same conditional DB, patterns from the earlier computation can be reused and redundant computation can be avoided.

5 Row-wise Mining of Closed Frequent Itemsets using ZBDDs

Let us now describe how our ZBDD-based mining framework can be adopted to the row-wise mining framework for finding CFIs which have been introduced in (Pan et al. 2003, Liu et al. 2006, Pan et al. 2004). In this framework, the patterns are mined by searching for possible row (instead of item) combinations. We can employ the column-wise mining framework described earlier, except now the ZBDD variables correspond to row-IDs (instead of items). Before we describe our technique, we firstly provide some background and terminology.

Let R be the set of row-ids. A *rowset* is a subset of R. For a given rowset X, $row\_support\_set(X)$ is the maximal itemset which occurs in all the rows in X. Moreover, for a given item x, we call the set of row-ids in which x occurs as x's *bit_support*. By definition, every *row_support_set* is a closed itemset. Works in (Pan et al. 2003, Liu et al. 2006) proposed a row-wise mining framework which is efficient for mining of CFIs in microarray datasets. It performs a depth-first search in the lattice of row-id (instead of item) combinations. We will describe a pattern growth, bottom-up algorithm (Pan et al. 2003) based on the use of ZBDD, although adapting it to the top-down algorithm (Liu et al. 2006) is certainly possible.

Our row-wise mining technique adopts the algorithm in (Pan et al. 2004), using a similar approach to FIMiner for creating database projections and performing a top-down traversal of the input ZBDD. Here, the input ZBDD is used to generate possible row combinations. A supplementary bitmap data structure is used accordingly for computing the *row_support_set* for each row combination. Unlike in our column-wise mining algorithms, there is no sharing between the input and the output ZBDDs in a row-wise mining since now they contain different sets of variables. To maximise node sharing within the output ZBDD which stores the closed frequent itemsets, we use the same variable ordering as in our column-wise algorithm i.e. by increasing frequency. The mining procedure is summarised as the following. Due to lack of space, we omit the implementation details of our algorithm.

The input ZBDD initially contains the *bit_support* of each item in the input dataset. Rowset prefixes are grown using the first variable (i.e. row id) in the ZBDD, which has a similar mechanism as growing itemset prefixes in our column-wise framework, while the minimum support threshold has not yet been reached by the rowset prefix or while the *row_support_set* is not empty. More specifically, as each rowset is being grown, its *row_support_set* is incrementally computed from the bitmap data representation. When the length of the prefix reaches the minimum support, its *row_support_set* is a fully-grown CFI, and it is inserted into the output ZBDD using ZBDD's set-union operation.

6 Performance Study

In this section, we analyse the performance of our ZBDD-based techniques for mining maximal frequent itemsets (MFIs) and closed frequent itemsets (CFIs). The algorithms were implemented in C++ using the BDD library, JINC, which was developed by the author of (Ossowski & Baier 2006) and used in their

study of another weighted variant of BDD. All experiments were performed on four 4.0 GHz CPUs, 32 GB RAM, running RedHat Linux 5, with a CPU-timeout limit of 100,000 seconds per mining task. We used two gene-expression datasets: Leukaemia ALL-AML [1], and lung cancer[2] which was previously studied in (Pan et al. 2004). The ALL-AML dataset contains 72 samples (i.e. transactions), each sample is described by 7129 genes (i.e. attributes). The lung cancer dataset contains 32 samples, described by 12533 genes.

Continuous attribute values are discretised using an entropy discretisation method, then, the discretised attributes are ordered according to their entropy values from highest to lowest (i.e. attr.1 has highest entropy reduction value). Entropy-based discretisation is the commonly used method for discretising microarray datasets (Creighton & Hanash 2003). To obtain results for low support threshold values, we performed our experiments using the first 100 attributes from ALL-AML dataset (All methods could not complete mining at low support thresholds when all of the attributes were used due to the CPU time out constraint). For a similar reason, we used the first 750 attributes from the lung cancer dataset. In the following discussions, we refer to the respective datasets as ALLAML-100 and lung-cancer-750.

We performed experiments for mining CFIs and MFIs in both datasets, but we do not include the results from mining MFIs in the ALLAML-100 dataset in this paper due to the similar output characteristics between the MFIs and CFIs. The pattern characteristics from both datasets will be shown shortly. We did some experiments for mining CFIs in the row-wise mining framework, but we do not include the results in this paper. Overall, our ZBDD-based method outperforms the FP-tree based method (Pan et al. 2004) when mining at low support thresholds.

In the item-wise framework, we implemented our techniques, i.e. *ZBDDMiner*, *WZDDMiner*, which are our ZBDD and WZDD based algorithms, using only the basic infrequent prefix pruning, and *ZBDDMiner** and *WZDDMiner** which use the more advanced pruning techniques. Their performance is compared against the state-of-the-art FP-tree based algorithms for mining CFIs and MFIs, i.e. *FP-close** and *FP-max** (Grahne & Zhu 2003)[3], which performed best on dense datasets. From each algorithm, we measure (i) the CPU time which is the total time spent for mining, (ii) the size of the output FP-trees, ZBDDs, or WZDDs which store the mined patterns in terms of the number of nodes, and (iii) the total nodes usage which is the total number of nodes used through out mining, which include the data structures for storing the input database, the intermediate databases, and the final output patterns. In ZBDDs or WZDDs, shared nodes are counted only once. When WZDDs are used for mining MFIs, storing support values in the output is not necessary, thus, they are removed from the output WZDDs

6.1 Patterns Distribution

In the ALLAML-100 dataset, the length distribution of closed frequent itemsets at support threshold 40% is shown in Figure 5(a). It shows that there are millions of relatively long patterns (the longest pattern contains 32 items). In lung cancer dataset, the length distribution of the closed frequent itemsets given support threshold of 40% is shown in Figure 5(b), and the length distribution of the maximal frequent itemsets is shown in Figure 5(c). Both output characteristics show there are only a relatively small number of patterns in this dataset, compared to the patterns contained in ALLAML-100 dataset. Thus, we categorise ALLAML-100 dataset as a dense dataset, and lung cancer-750 as a sparse dataset.

6.2 Mining CFI in a Sparse Dataset

Firstly, let us observe the performance comparison between our ZBDD-based and WZDD-based algorithms for mining CFIs in lung cancer-750 dataset against FP-close*. Our WZDD-based algorithms could not complete mining for support threshold $< 40\%$, whilst the ZBDD-based algorithms could not complete mining for support threshold $< 50\%$, due to the CPU timeout constraint.

Even though this dataset is relatively sparse, in Figure 6(a) it is shown that the ZBDD representations (with support encoding variables) for storing the output patterns are smaller than the FP-trees for support threshold $< 60\%$, with the weighted ZBDDs being the most compact. Figure 6(b) shows that FP-close* has the fastest running times, and the WZDD based algorithms are faster than the ZBDD based algorithms. This is expected from this dataset due to its sparse characteristics, in which there is less likely that many nodes are shared across the conditional databases. Now let us look closer at the performance of our algorithms which use advanced pruning strategies, i.e. ZBDDMiner* and WZDDMiner*. It shows that ZBDDMiner* improves the efficiency of ZBDDMiner, but WZDDMiner* does not improve WZDDMiner except for the low support threshold of 36%. When the support threshold is low, many conditional databases are being induced, and WZDDMiner* benefits from its ability to re-use intermediate pruning computations.

The total nodes usage of all algorithms is shown in Figure 6(c). It shows that FP-close* uses the least number of FP-tree nodes for representing all the databases (including the input, the output patterns, and the intermediate structures). However, the discrepancies with our ZBDD and WZDD based algorithms are decreasing as the support threshold decreases, with the WZDDs having a similar nodes usage to FP-trees at support threshold of 37%.

Furthermore, ZBDDMiner* uses about 5 times fewer nodes than ZBDDMiner, which shows the effectiveness of the advanced pruning strategies for reducing the size of the intermediate structures, and in turn results in a more efficient mining of ZBDDMiner* over ZBDDMiner. On the other hand, WZDDMiner* has a slightly more nodes usage than WZDDMiner. This shows that there are more nodes being shared between the input database and the conditional databases when no advanced pruning is used. Given high support threshold, moreover, there are not many patterns in this sparse dataset, hence, not much nodes are shared across the conditional databases.

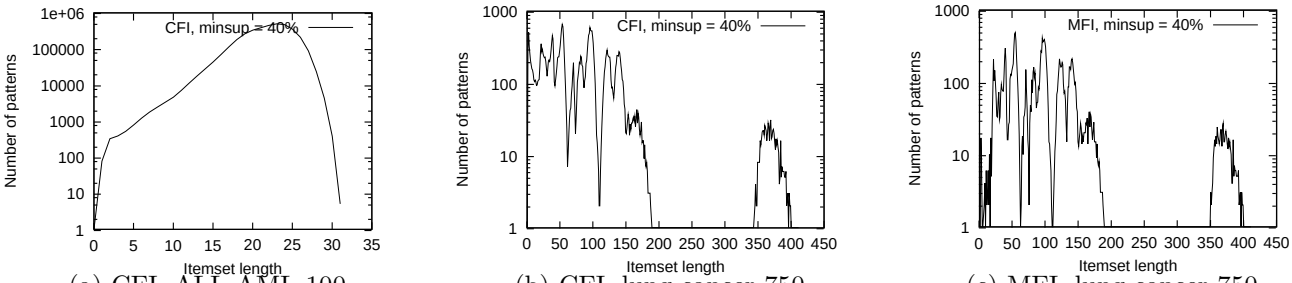
6.3 Mining CFI in a Dense Dataset

When mining CFIs in ALLAML-100 dataset, ZBDDMiner could not complete within the CPU time limit for support threshold $< 50\%$, and FP-close* exceeds the memory limits for support threshold as low as 27%. The WZDDs for storing the output patterns are significantly smaller than FP-trees, as shown in Figure 7(a), which demonstrates the ability of WZDDs to compactly represent huge volume of patterns. The ZBDD (with additional support-encoding variables) representations are about 100 times larger than WZDDs.

[1] http://research.i2r.a-star.edu.sg

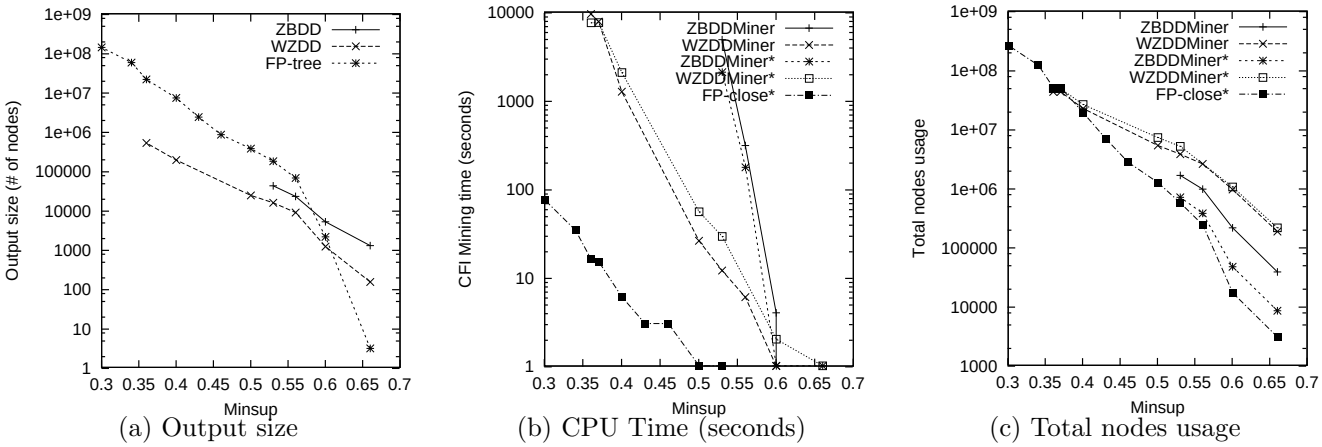
[2] http://www-genome.wi.mit.edu/cgi-bin/cancer

[3] FP-close* is a variant of FP-growth* for mining CFIs, FP-max* is a variant of FP-growth* for mining MFIs. Their implementation can be found in (Goethals 2004)



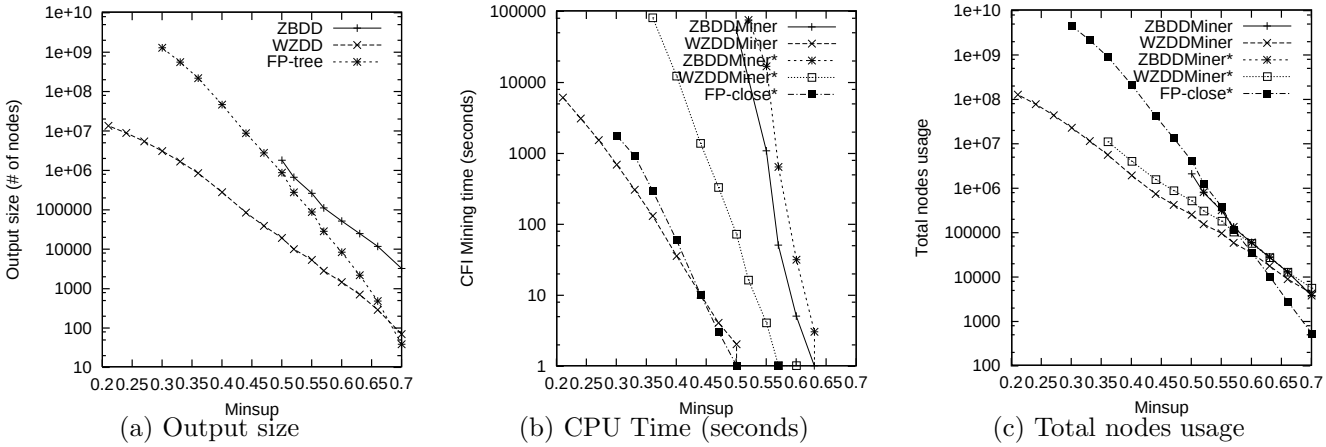
(a) CFI, ALL-AML-100 (b) CFI, lung-cancer-750 (c) MFI, lung-cancer-750

Figure 5: Patterns Length Distribution



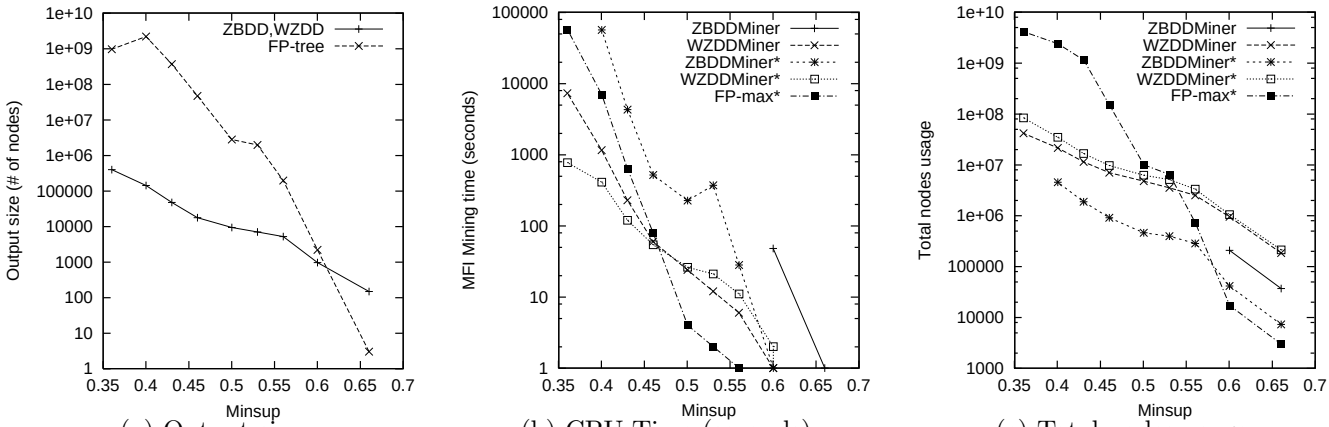
(a) Output size (b) CPU Time (seconds) (c) Total nodes usage

Figure 6: Results from mining CFIs in lung-cancer-750 dataset



(a) Output size (b) CPU Time (seconds) (c) Total nodes usage

Figure 7: Results from mining CFIs in ALL-AML-100 dataset



(a) Output size (b) CPU Time (seconds) (c) Total nodes usage

Figure 8: Results from mining MFIs in lung-cancer-750 dataset

Figure 7(b) shows the runtime comparison between the algorithms. WZDDMiner has the fastest runtime for low support thresholds , i.e. $< 40\%$. When advanced pruning strategies are used, WZDDMiner* is up to 500 times slower than WZDDMiner, but on the other hand, ZBDDMiner* can achieve up to 100 times speedup factor over ZBDDMiner.

Furthermore, Figure 7(c) shows WZDDMiner has the least total nodes usage compared to all the other algorithms except for support threshold $\geq 60\%$ for which the FP-trees have the least total nodes usage.

6.4 Mining MFI in a Sparse Dataset

When mining MFIs in this sparse dataset, Figure 8(a) shows that the ZBDD representations for storing the output patterns are smaller than the FP-trees for support threshold $< 60\%$, being up to 1000 times smaller at support threshold of 40%. This shows the significant data compression being achieved by ZBDDs over FP-trees.

Figure 8(b) shows the runtime comparison between our algorithms against FP-max*. It shows that for high support threshold, i.e. $\geq 50\%$, FP-max* is the fastest, but WZDDMiner* is the fastest for lower support threshold. Unlike in mining CFIs, WZDDMiner* does improve the mining efficiency of WZDDMiner for low support threshold, which indicates that in this dataset, re-using the pruning computations from multiple conditional databases in WZDDMiner* is beneficial.

Although ZBDDMiner* is slower, but it uses the least number of nodes through out mining for support threshold $< 60\%$, which is shown in Figure 8(c). More specifically, ZBDDMiner* could achieve about 50 times data compression over both WZDDMiner and WZDDMiner*, FP-max* uses up to 100-1000 times more nodes than both WZDDMiners and ZBDDMiners when support threshold is low, i.e. $< 50\%$.

7 ZBDD VS FP-tree

In this section we discuss some advantages and disadvantages of using ZBDDs (and their weighted variants) in an FI mining framework, based on our experimental results, with respect to a number of dimensions: the compactness of ZBDD data structure, the effectiveness of pruning in a ZBDD-based technique, and the effectiveness of ZBDD's caching utility.

7.1 ZBDD's canonicity

The canonical structure of a ZBDD is a a powerful feature which we have shown useful for not only compressing high dimensional output patterns, but also for compressing the intermediate structures used for mining them, by allowing the various databases to share nodes. This compression has been proven in our experimental results when mining MFIs and CFIs at relatively low support threshold, for which the FP-trees had billions of overall nodes usage. In particular, in the dense dataset, WZDDs are able to achieve up to 500 times overall data compression over FP-trees, as shown in the total nodes usage comparison. In such a circumstance, many long patterns exist and many sub-trees may be shared across multiple databases (including the conditional DBs) which is not possible using FP-trees. On the other hand, in the sparse dataset in which not too many conditional DBs are being projected, ZBDDs allow a higher overall data compression than WZDDs, being able to share more nodes when representing the intermediate databases but they require the use of *bitmap* for support counting as a tradeoff.

Furthermore, when ZBDDs are used for computing CFIs and the output data structure contains additional support-encoding variables, we found that although the size of the ZBDDs is increased, they can still contain fewer nodes than FP-trees when the support threshold is relatively low.

7.2 Pruning effectiveness

Our techniques use a basic infrequent prefix pruning as well as some advanced pruning strategies. We will discuss the effectiveness of each type of pruning shortly. Firstly, let us consider the infrequent prefix pruning. FP-trees may allow earlier pruning of infrequent prefixes by allowing different item orderings to be used for different database projections. On the other hand, ZBDDs and WZDDs use static item ordering as a tradeoff for achieving data compression. This explains a number of situations in our experimental results when ZBDDs and WZDDs are less efficient than FP-trees, when mining CFIs in the sparse dataset or mining MFIs/CFIs in the other dataset given a high support threshold, in which a large search space can be pruned earlier by the FP-trees.

Secondly, the effectiveness of advanced pruning strategies rely upon the amount of search space reduction over the overhead of performing the pruning routines. As part of the advanced pruning routines, ZBDDMiner* uses a bit-wise-and operation for support counting, whereas WZDDMiner* has to traverse the database which is an expensive computation to perform in high-dimensional datasets. Based on our experiments, ZBDDMiner* does improve mining efficiency of ZBDDMiner when the total size of the conditional DBs is significantly reduced, such as when mining MFIs/CFIs in the sparse dataset.

WZDDMiner*, on the other hand, although it has a higher computation overhead for performing the advanced pruning, it is able to use ZBDD's caching utility for remembering the set of frequent items in each database. It is therefore beneficial when there is a large number of nodes being shared across the various conditional databases, as shown in our experiments when mining MFIs/CFIs at low support threshold in the sparse dataset. In other circumstances, WZDDMiner* is less efficient than WZDDMiner. For instance when the support threshold is relatively high, only a few database projections are required (and the databases are less likely to share many nodes), for which ZBDD's ability to re-use the intermediate pruning computations is not beneficial. Also, in the dense dataset, since a large volume of patterns exist, the effects of pruning do not greatly reduce the size of the databases, thus, the overhead of performing advanced pruning in WZDDMiner* is not compensated.

7.3 Cached computation results

Another powerful feature of ZBDDs is their caching principle. The ZBDD routines used in our framework allow the result from intermediate computations, to be cached. More particularly in the WZDD-based framework, the output patterns from every conditional DB are cached and it has been proven useful to allow more efficient mining. In dense datasets, or in sparse datasets with low support threshold, millions of long patterns exist and different prefix itemsets may project similar conditional DBs (and thus, similar patterns). This has proven a significant time improvement by WZDDs over FP-trees, despite the use of static item ordering which may hurt its efficiency as we have discussed earlier. In particular, our experimental results show the WZDD-based technique (without advanced pruning) outperforms the FP-tree based technique (with advanced pruning)

when mining huge volume of MFIs (indicated by the number of nodes in the output data structures) at low support threshold, and it has similar time performance as the FP-tree based technique for mining CFIs in the relatively denser dataset.

Summary of Performance: We now return to the questions which were posed at the beginning of the paper:

1. *Does the canonical property of ZBDDs allow an efficient and scalable algorithm for frequent itemset mining to be developed ?*

   As we have seen in our experimental results, the WZDD based algorithm is superior over both ZBDDs and FP-trees for mining MFIs at low support threshold, or mining CFIs in a dense dataset. In such a circumstance, millions of auxiliary conditional DBs are induced, as indicated by the volume of patterns. Their canonical WZDD representations (which are substantially smaller than FP-trees) allow the computed patterns and other intermediate results to be reused, which in turn allow mining to be performed efficiently.

2. *How much data compression can a ZBDD achieve compared to an FP-tree ?*

   (a) When mining CFIs, the total size of non-weighted ZBDDs used throughout mining may be larger than the FP-trees, this being a result of having the extra support encoding variables on the output. They, however, could achieve slightly higher overall data compression by a factor of 2 compared to FP-trees, as shown in our results when mining CFIs at low support in dense datasets.

   More specifically, for storing the output CFIs, ZBDDs with support-encoding variables increase the size of the traditional ZBDDs (without support-encoding variables) by 100 times, yet, they may contain fewer nodes than FP-trees when mining CFIs at low support threshold.

   (b) When mining MFIs, non-weighted ZBDDs are able to achieve further overall data compression up to 100 times more compact than WZDDs, as found in our experiments. This shows WZDDs achieve lower data compression than ZBDDs due to their weighted edges. However, WZDDs are still much more compact than FP-trees for representing all of the databases created throughout mining, more particularly if the support threshold is not too high, as shown in our results that they used up to 1000 times fewer nodes than the FP-trees

3. *Does the use of a more compact data structure really mean that mining is more efficient?*

   Not always. When compared to FP-trees, there are situations where the intermediate ZBDDs or WZDDs are highly compressed and the total nodes usage is much less, for which intermediate results can be reused effectively and mining speedups are increased over the FP-tree based technique. More particularly, such situations were found in our experimental results when mining huge volume of MFIs/CFIs. On the other hand, if the ZBDD/WZDD data representations were of similar size than FP-trees, the FP-tree based techniques are more efficient as they use a more flexible variable ordering which allows earlier search space pruning.

   Moreover, if we compare ZBDDs against the weighted ZBDDs, although the ZBDD representations in the intermediate computations are often smaller than the WZDDs, based on our findings, it does not always mean a more efficient mining was obtained. This can be explained by the extra cost of using a secondary data bitmap with (non-weighted) ZBDDs, which may require expensive computation in high dimensional datasets.

8 Related Work

We are aware of several other works which use ZBDDs for itemset mining (Minato & Arimura 2006, Minato & Ito 2007). Work in (Minato & Ito 2007) demonstrated that ZBDDs are useful for mining patterns in high dimensional amino acid datasets.

Work in (Minato & Arimura 2006) proposed a ZBDD-based pattern growth mining of frequent itemsets, which uses a different data representation schema to that used in our proposed technique. Its optimised variant for mining closed frequent itemsets is proposed in (Minato & Arimura 2007). Their ZBDD representation of the databases encode the support values by storing the itemsets in multiple ZBDD functions based on their binary support values, whose representations are referred as *tuple histograms*. Their experiments show their technique outperforms FP-growth (Han et al. 2004) for mining huge volume of patterns in the traditional, low dimensional, type of datasets. Such a representation is less compact and requires more complex routines to construct, instead of the simpler, and faster, basic ZBDD routines used in our framework. Therefore, their technique does not seem scalable for mining the more challenging microarray datasets which have exponentially large search space. Furthermore, we have shown our technique can be adapted to the row-wise mining framework.

A more recent work in (Iwasaki et al. 2007) proposed a method for choosing a good variable ordering for ZBDDs in data mining applications. The method computes the variable ordering after the ZBDD has been built, and re-arranges the nodes. This method is different to ours which decides the variable ordering prior to constructing the ZBDDs. Our variable ordering heuristics aim to achieve an efficient mining of frequent itemsets, as well as to achieve an overall compact data representation across multiple intermediate data structures used throughout mining, whereas their method finds a variable ordering which is optimised for a particular ZBDD.

Other tree data structures have been proposed in other frequent itemset mining techniques (Zaki et al. 2004, Liu et al. 2003, Pietracaprina & Zandolin 2003). None of them, however, allows node sharing across the auxiliary databases, which is a key feature of our technique. AFOPT (Liu et al. 2003) is a prefix-tree structure which is designed to work for a top-down traversal in projecting the conditional DBs similar to our traversal strategy. However, it cannot achieve much data compression. Patricia trie (Pietracaprina & Zandolin 2003) combines prefix trees with array-list to achieve a more compressed structure but it does not allow sharing among multiple databases, and it does not yet have optimisations for mining maximal/closed frequent itemsets.

Work in (Loekito & Bailey 2006) demonstrated that ZBDDs can be used to efficiently mine contrast patterns, which include two support constraints on two respective datasets, one being anti-monotonic and the other monotonic. Their technique also uses a supplementary bitmap data representation for sup-

port counting in a similar mechanism to the ZBDD-based techniques proposed in this paper.

9 Conclusions and Future Work

In this paper, we have examined the use of advanced data structures, ZBDDs, for mining (maximal/closed) frequent itemsets, and identified situations where they are superior over state of the art FP-tree-based technique. Overall, we found that ZBDD allows much higher data compression for storing huge volume of long patterns as well as the intermediate structures used in mining them. We also introduced a weighted type of ZBDD, which is able to improve mining efficiency than the classic ZBDD. Although our ZBDD-based framework is not uniformly superior than FP-trees, it can sometimes allow more efficient mining in relatively dense high dimensional datasets at low support thresholds. We believe this result suggests that ZBDDs can be a very valuable tool in data mining. We would like to further investigate their use in other scenarios, considering more complex constraints and other types of patterns.

Acknowledgement

We would like to thank Jian Pei, one of the authors of the FP-growth technique, for his useful comments on the compactness of a ZBDD compared to an FP-tree, and the efficiency of the mining algorithm between those two database representations.

We also would like to thank Joern Ossowski for providing us the JINC package and for his helpful comments on the implementation of our WZDD data structure which relies on JINC's core library.

This work is partially supported by National ICT Australia. National ICT Australia is funded by the Australian Government's Backing Australia's Ability initiative, in part through the Australian Research Council.

References

Aloul, F. A., Mneimneh, M. N. & Sakallah, K. (2002), ZBDD-based backtrack search SAT solver, *in* 'International Workshop on Logic Synthesis', University of Michigan.

Bryant, R. E. (1986), 'Graph-based algorithms for boolean function manipulation', *IEEE Transactions on Computers* **35**(8), 677–691.

Bryant, R. E. & Chen, Y.-A. (1995), Verification of arithmetic circuits with Binary Moment Diagrams, *in* 'DAC'95: Proceedings of the 32nd ACM/IEEE Conference on Design Automation', pp. 535–541.

Burdick, D., Calimlim, M. & Gehrke, J. (2001), MAFIA: A maximal frequent itemset algorithm for transactional databases, *in* 'International Conference on Data Engineering (ICDE'01)', pp. 443–452.

Coudert, O. (1997), 'Solving graph optimization problems with ZBDDs', *In Design, Automation and Test in Europe* pp. 224–228.

Creighton, C. & Hanash, S. (2003), 'Mining gene expression databases for association rules', *Bioinformatics* **19**(1), 79–86.

Goethals, B. (2004), 'Frequent itemset mining implementations (FIMI) repository'.
URL: *http://fimi.cs.helsinki.fi/*

Grahne, G. & Zhu, J. (2003), Efficiently using prefix-trees in mining frequent itemsets, *in* 'Proceedings of the 1st IEEE ICDM FIMI'03 Workshop on Frequent Itemset Mining Implementations'.

Han, J., Pei, J., Yin, Y. & Mao, R. (2004), 'Mining frequent patterns without candidate generation: An FP-Tree approach', *Data Mining and Knowledge Discovery* **8**(1), 53–87.

Iwasaki, H., Minato, S. & Zeugmann, T. (2007), A method of variable ordering for Zero-suppressed Binary Decision Diagrams in data mining applications, *in* 'Proceedings of the 3rd IEEE International Workshop on Databases for Next-Generation Researchers, SWOD 2007', pp. 85–90.

Li, J., Liu, H., Downing, J. R., Yeoh, A. & Wong, L. (2003), 'Simple rules underlying gene expression profiles of more than six subtypes of Acute Lymphoblastic Leukemia (ALL) patients', *Bioinformatics* **19**, 71–78.

Liu, G., Lu, H., Yu, J. X., Wei, W. & Xiao, X. (2003), AFOPT: An efficient implementation of pattern growth approach, *in* 'Proceedings of the 1st IEEE ICDM FIMI'03 Workshop on Frequent Itemset Mining Implementations'.

Liu, H., Han, J., Xin, D. & Shao, Z. (2006), Top-down mining of interesting patterns from very high dimensional data, *in* 'Proceedings of the 22nd International Conference on Data Engineering (ICDE'06)', p. 114.

Loekito, E. & Bailey, J. (2006), Fast mining of high dimensional expressive contrast patterns using ZBDDs, *in* 'Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'06)', pp. 307–316.

Minato, S. (1993), Zero-suppressed BDDs for set manipulation in combinatorial problems, *in* 'Proceedings of the 30th International Conference on Design Automation', pp. 272–277.

Minato, S. (2001), 'Zero-suppressed BDDs and their applications', *International Journal on Software Tools for Technology Transfer (STTT)* **3**(2), 156–170.

Minato, S. (2005), Finding simple disjoint decompositions in frequent itemset data using Zero-suppressed BDD, *in* 'Proceedings of IEEE ICDM'05 Workshop on Computational Intelligence in Data Mining', pp. 3–11.

Minato, S. & Arimura, H. (2005), Combinatorial item set analysis based on Zero-suppressed BDDs, *in* 'IEEE Workshop on WIRI 2005', pp. 3–10.

Minato, S. & Arimura, H. (2006), Frequent pattern mining and knowledge indexing based on Zero-suppressed BDDs, *in* 'The 5th International Workshop on Knowledge Discovery in Inductive Databases (KDID'06)', pp. 83–94.

Minato, S. & Arimura, H. (2007), 'Frequent closed item set mining based on Zero-suppressed BDDs', *Transaction of the Japanese Society of Artificial Intelligence* **22**(2), 165–172.

Minato, S. & Ito, K. (2007), 'Symmetric item set mining method using Zero-suppressed BDDs and application to biological data', *Transaction of the Japanese Society of Artificial Intelligence* **22**(2), 156–164.

Mishchenko, A. (2001), 'An introduction to Zero-suppressed Binary Decision Diagrams'.
URL: *http://www.ee.pdx.edu/ãlanmi/research.htm*

Ossowski, J. & Baier, C. (2006), Symbolic reasoning with weighted and normalized decision diagrams, *in* 'Proceedings of the 12th Symposium on the Integration of Symbolic Computation and Mechanized Reasoning', pp. 35–96.

Pan, F., Cong, G., Tung, A. K. H., Yang, J. & Zaki, M. (2003), CARPENTER: Finding closed patterns in long biological datasets, *in* 'Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03)', pp. 637–642.

Pan, F., Cong, G. & Tung, A. K. H.and Tan, K. (2004), Mining frequent closed patterns in microarray data, *in* 'Proceedings of the 2nd IEEE ICDM FIMI'04 Workshop on Frequent Itemset Mining Implementations', pp. 363–366.

Pietracaprina, A. & Zandolin, D. (2003), Mining frequent itemsets using patricia tries, *in* 'Proceedings of the 1st IEEE ICDM FIMI'03 Workshop on Frequent Itemset Mining Implementations'.

Rioult, F., Boulicaut, J., Crémilleux, D. & Besson, J. (2003), Using transposition for pattern discovery from microarray data., *in* 'Proceedings of 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD'03)', pp. 73–79.

Uno, T., Kiyomi, M. & Arimura, H. (2005), LCM ver. 3: Collaboration of array, bitmap and prefix tree for frequent itemset mining, *in* 'Proceedings of ACM SIGKDD Open Source Data Mining Workshop on Frequent Pattern Mining Implementations (OSDM'05)', pp. 77–85.

Vrudhula, S., Pedram, M. & Lai, Y. (1996), 'Edge-valued binary decision diagram', *Representation of Discrete Functions* pp. 109–132.

Wang, J., Han, J. & Pei, J. (2003), CLOSET+: Searching for the best strategies for mining frequent closed itemsets, *in* 'Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03)', pp. 236–245.

Zaki, M., Goethals, B. & Bayardo, R., eds (2004), *Proceedings of the 2nd IEEE ICDM FIMI'04 Workshop on Frequent Itemset Mining Implementations*, Vol. 126 of *CEUR Workshop Proceedings*.

Zaki, M. & Goethals, B., eds (2003), *Proceedings of the 1st IEEE ICDM FIMI'03 Workshop on Frequent Itemset Mining Implementations*, Vol. 80 of *CEUR Workshop Proceedings*.

PCITMiner – Prefix-based Closed Induced Tree Miner for finding closed induced frequent subtrees

Sangeetha Kutty Richi Nayak Yuefeng Li

Faculty of Information Technology
Queensland University of Technology
GPO Box 2434, Brisbane Qld 4001, Australia

{kutty,r.nayak,y2.li}@qut.edu.au

Abstract

Frequent subtree mining has attracted a great deal of interest among the researchers due to its application in a wide variety of domains. Some of the domains include bio informatics, XML processing, computational linguistics, and web usage mining. Despite the advances in frequent subtree mining, mining for the entire frequent subtrees is infeasible due to the combinatorial explosion of the frequent subtrees with the size of the datasets. In order to provide a reduced and concise representation without information loss, we propose a novel algorithm, PCITMiner (Prefix-based Closed Induced Tree Miner). PCITMiner adopts the prefix-based pattern growth strategy to provide the closed induced frequent subtrees efficiently. The empirical analysis reveals that our algorithm significantly outperforms the current state of the art algorithm, PrefixTreeISpan(Zou, Lu, Zhang, Hu and Zhou 2006b).

Keywords: Frequent subtree mining, closed, induced trees, subtrees, frequent mining

1 Introduction

Recently, there have been an increasing number of researches in frequent subtree mining due to the simplicity of the mining process and the potential of its application in various domains. Some of the domains include bio informatics, XML processing, database management, and web usage mining (Tatikonda, Parthasarathy and Kur 2006). Additionally, frequent subtree mining serves as the kernel function for other data mining techniques such as association rules mining, classification and clustering.

A number of algorithms have been proposed to extract frequent subtrees efficiently from a given tree dataset. However, there exists an immense disadvantage faced by these algorithms. For instance, there are often situations in which the number of frequent subtrees increases exponentially with the size of the tree dataset causing difficulties for the end-user to analyse the results (Chi, Yang, Xia and Muntz 2004b).

Due to the overwhelming number of frequent subtrees, the frequent subtree mining algorithms fail to provide a complete output. In order to provide a feasible solution as well as to improve the performance, the frequent subtree mining algorithms have focused on generating a concise but lossless representation of frequent subtrees. Two such popular representations of frequent subtrees are closed and maximal representations. One popular technique for generating the closed and maximal frequent subtrees is the CMTreeMiner (Chi et al. 2004b). However, this algorithm employs the "*generate-and-test*" technique to generate the closed frequent subtrees. Popularized by apriori-based algorithms(Agrawal, Mannila, Srikant, Toivonen and Verkamo 1996), the *generate-and-test* technique basically involves two processes namely the candidate generation from the smaller sized frequent subtrees and then the testing of the candidate frequent subtrees against the tree dataset. Unfortunately, the *generate-and-test* technique is an expensive operation when the numerous candidate frequent subtree checks are required (Termier, Rousset, Sebag, Ohara, Washio and Motoda 2005).

On the other hand, studies on the frequent itemset and sequential mining have clearly demonstrated that the pattern-growth technique provides improved performance in comparison to the *generate-and-test* technique to generate the frequent itemsets or sub-sequences(Pei 2002; Han, Pei and Yan 2005)especially on dense datasets. Hence, in this paper we propose an efficient algorithm called *PCITMiner* (**P**refix-based **C**losed **I**nduced **T**ree Miner), which combines the strengths of pattern growth technique and the closure property to discover the frequent closed induced subtrees. The experimental results indicate that our proposed approach outperforms the base algorithm *PrefixTreeISpan(Zou et al. 2006b)* by resulting in a reduced number of frequent subtrees and with an improved time performance in generating the frequent induced subtrees. The experimental results also manifests that our proposed approach is a better performing algorithm for heavily branched trees.

To the best of our knowledge, there exists no subtree mining algorithm using the pattern growth technique to identify the closed frequent induced subtrees. Hence, in this paper we develop the novel algorithm *PCITMiner* (**P**refix-based **C**losed **I**nduced **T**ree Miner) using the pattern growth technique for providing the closed frequent induced subtrees in a computationally efficient manner.

The rest of the paper is organized as follows. In Section 2, we will be presenting the background of the frequent subtree mining and the problem definitions. Section 3 contains a review of the frequent subtree mining algorithms. Section 4 details the description of *PCITMiner*, while Section 5 presents a performance comparison of running the PCITMiner algorithm against an implementation of PrefixTreeISpan algorithm(Zou et al. 2006b). PrefixTreeISpan was chosen as the baseline as it is the state-of-the-art algorithm in the area of frequent subtree mining using the pattern-growth technique.

2 Background concepts and Problem Definition

Before, explaining about the frequent subtree mining process we will look into what is meant by trees and the types of trees and subtrees.

A tree is denoted as $T = (V,v0, E, f)$, where V is the set of nodes; $v0$ is the root node which does not have any edges entering into it; E is the set of edges in the tree T; f is a mapping function $f: E \rightarrow V \times V$.

There are different types of trees namely *free trees* or *rooted trees*, *ordered* or *unordered trees* and *labelled* or *unlabelled trees*. If a given tree T has a root node, $v0$, then T is called as a *rooted tree* otherwise a *free tree*. An *ordered tree* is a tree which preserves a pre-defined ordering such as left-to-right among the set of nodes. Finally, if a tree T has labels for its edges then T is a *labelled tree*. The proposed technique will be applied on the *labelled rooted ordered trees.*

In the frequent mining of trees, it has been noted that often the entire tree will not be frequent rather there is a good possibility that parts of the tree are frequent. The parts of such trees are referred to as *subtrees*. A tree T' is a subtree of T if there exists a subtree isomorphism from T' to T. This implies that there is a one-to-one mapping from the vertices of T' to the vertices of T and it preserves the vertex labels, edge labels and adjacency then T' is a subtree of T. There exist different perspectives about subtrees and the two popular types of subtrees are the induced and embedded subtrees(Chi, Nijssen, Muntz and Kok 2005).

Embedded subtree

For a tree T with an edge set E and a vertex set V, a Tree T' with a vertex set V' and an edge set E' is an embedded subtree of T iff (1) $V' \subseteq V$; (2) $E' \subseteq E$; (3) the labelling of nodes of V' in T' is preserved in T; (4) $(v1,v2) \in E$ where $v1$ is the parent of $v2$ in T' iff $v1$ is the parent of $v2$ in T; and (5) for $v1,v2 \in V'$, preorder($v1$) < preorder($v2$) in T' iff preorder($v1$) < preorder($v2$) in T. An embedded subtree T' preserves the ancestor-descendent relationships among the vertices of the tree, T.

Induced subtree

For a tree T with an edge set E and a vertex set V, a Tree T' with a vertex set V' and an edge set E' is an induced subtree of T iff (1) $V' \subseteq V$; (2) $E' \subseteq E$; and (3) the labelling of the nodes of V' and E' in T' is preserved in T. An induced subtree T' is a subtree which preserves the parent-child relationships among the vertices of the tree, T.

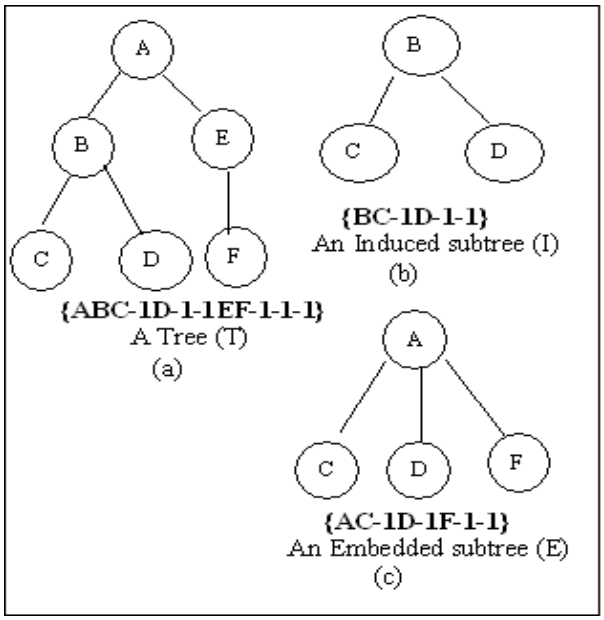


Figure 1: (a) A Tree, T (b) Induced subtree, I (c) Embedded subtree, E with their respective pre-order string in curly braces

For a given Tree, T (in Figure 1(a)), Figures 1(b) and 1(c) show the induced and embedded subtrees of T. It can be seen that the induced subtree, I preserves the parent-child relationships, and the embedded subtree, E preserves only the ancestor-descendent relationships. Hence, though A is not the parent of C, D and F, the embedded subtree, E has A as the root node as it is the ancestor of the three nodes C, D and F.

The curly braces in Figure 1 below the Tree, T and the subtrees, I and E, are their respective pre-order string format (as defined in (Chi et al. 2005)). The pre-order string format represents the pre-order traversal of a tree in a string like format where every node has a "–1" as its end flag. For a rooted ordered tree T with only one node r, the pre-order string of T is $S(T) = l_r –1$ where l is the label of the root node r. On the other hand, for multiple nodes for the rooted ordered tree T, where r is the root node and the children nodes of r are $r_1,...,r_k$ preserving left to right ordering. Then the pre-order string for T is $S(T)= l_r\ S(T_{r_1}) \ldots S(T_{r_k})\text{-}1$.

Having explained the background concepts about trees and subtrees, we will now detail the frequent subtree mining problem.

Problem definition for the frequent subtree mining

Given a tree dataset $D = \{T_1, T_2, T_3, \ldots, T_n\}$ with n number of trees, there exists a subtree $T' \subseteq T_k$ preserving the relationships (either the parent-child relationship or the ancestor-descendent relationship) among the nodes as that of the tree T_k. Support(T') (or frequency(T')) is defined as the percentage (or the number) of trees in D where T' is a subtree. A subtree T' is frequent if its support is not less

than a user-defined minimum support threshold. In other words, T' is a frequent subtree of the trees in D such that (*frequency* $(T')/|D|) \geq min\_supp$, where *min_supp* is the support threshold and $|D|$ is the number of trees in the tree dataset D.

Mining for the frequent induced subtrees from a huge tree dataset causes difficulties in analysing the result due to the combinatorial explosion in the number of frequent subtrees generated at lower support, consequently the frequent subtree mining algorithms become intractable. Hence, it is essential to control the number of subtrees generated. In order to reduce the number of subtrees without any information loss, two concise representations of frequent subtrees namely maximal and closed were proposed(Chi et al. 2004b).

Problem definition for Closed and Maximal subtree

In a given tree dataset, $D = \{T_1, T_2, T_3, ..., T_n\}$, if there exists two frequent subtrees T' and T'', T' is said to be maximal of T'' iff $\forall\ T' \supseteq T''$, $supp(T') \leq supp(T'')$; and a frequent subtree T' is closed of T'' iff for every $T' \supseteq T''$, $supp(T') = supp(T'')$. The latter property is called as *closure*. Based on the definition, it can be said that $M \leq C \leq F$, where

M = Number of Maximal frequent subtrees

C = Number of Closed frequent subtrees

F =Number of Frequent subtrees

Let us analyse the two concise representations, maximal and closed frequent subtrees. Firstly, we will apply closure on frequent subtrees generated from a given tree dataset D. Before that, we will define the frequent subtrees for a given *min_supp* of k. Let us assume that the frequent subtree mining result set $O = \{T'_1, T'_2, T'_3\}$ contains three frequent subtrees having a support of k, $k+1$ and k respectively. Also consider, $T'_1 \subseteq T'_3$, $T'_2 \subseteq T'_3$ and T'_3 does not have any superset. Applying the definition of the closed frequent subtrees on the frequent subtrees result set, O, it is found that T'_1 and T'_3 have the same support and $T'_3 \supseteq T'_1$. As a result, T'_1 is not closed and it can be removed from the output as its superset, T'_3, includes the information contained in T'_1. Also, there exists no superset of T'_3 therefore T'_3 is closed. Hence, T'_2 and T'_3 are the two closed frequent subtrees.

On the contrary, let us check whether T'_2 and T'_3 are maximal or not. We have not included T'_1 as the number of maximal frequent subtrees is less than the number of closed frequent subtrees and hence T'_1 cannot be maximal as it is not closed. According to the definition of maximal frequent subtrees, T'_3 is the maximal frequent subtree due to the reason that $T'_3 \supseteq T'_2$. There is only one maximal frequent subtree, which is a reduced number in comparison to the closed subtrees (that is two). The total number of output patterns are less considering the maximal frequent subtree representation, however, this representation suffers from information loss. Considering the above example, the frequent subtree T'_2 has a support of k+1 which implies that T'_2 occurs more than T'_3 but, based on the final output maximal pattern, it can be inferred that the support of T'_2 is k as the extra occurrence information has not been included in the output.

Therefore, the comparison of these two concise representations reveals that though the maximal frequent subtree representation provides reduced pattern set it results in information loss. Alternatively, the closed frequent subtree representation provides a concise pattern set without any information loss as the closure property eliminates only the redundant information. This reason is attributed for the popularity of closed frequent subtrees over the maximal frequent subtrees.

3 Related Research

Research on the frequent subtree mining spans from the frequent mining on different types of subtrees namely induced and embedded using various performance tuning techniques to the recent researches focusing on efficiently generating concise representations such as closed and maximal.

The seminal work on the embedded subtrees were conducted by the TreeMinerV(Zaki 2005), TreeMinerH(Zaki 2005) to generate the frequent embedded subtrees. Some of the earlier works on the frequent induced subtree mining include TreeFinder(Termier, Rousset and Sebag 2002), Freqt(Asai, Abe, Kawasoe, Arimura, Satamoto and Arikawa 2002), uFreqt(Asai, Arimura, Uno and Nakano 2003), HybridTreeMiner (Chi, Yang and Muntz 2004a), Unot(Asai et al. 2003) to generate the frequent induced subtrees.

The frequent subtree mining algorithms can be broadly classified into two groups namely (1) *generate and test* and (2) *pattern-growth* based on the strategy they adopt to identify the frequent subtrees. Algorithms such as TreeMiner(Zaki 2002) and Freqt(Asai et al. 2002) fall into the category of *generate-and-test* in which the candidate frequent subtrees are generated from the previous step and tested whether they are frequent or not using the tree dataset. These algorithms adopt a *level-wise* mining methodology where at each level the size of the newly discovered subtree is increased by one by joining or extending frequent subtrees generated from the previous level. For instance, the *(K+1)-Length* candidate frequent subtrees are generated by extending or joining the frequent *K–Length* subtrees where *K* refers to the number of levels in the subtree. A candidate *(K+1)-Length* frequent subtree is tested against the dataset to verify whether the candidate *(K+1)-Length* subtree is frequent or not. If it is frequent, then the *(K+1)-Length* subtree is included in the result set and it is used to generate *(K+2)-Length* subtree and this process is repeated until there are no more candidate subtrees that could be found.

The *Pattern-growth* strategy was adopted by Xspanner(Wang, Hong, Pei, Zhou, Wang and Shi 2004), Chopper(Wang et al. 2004), PrefixTreeISpan(Zou et al. 2006b) and PrefixTreeESpan(Zou, Lu, Zhang and Hu 2006a). These algorithms utilize the strategy of extending the discovered subtrees recursively until there are no frequent subtrees that could be found. As the extension of the discovered subtree is conducted by identifying the

extensions from the tree dataset, pattern-growth strategy does not involve any generation and testing of candidates. Though Xspanner and Chopper are included in the pattern-growth category, it adopts the *generate-and-test* techniques to generate candidates and hence it cannot be considered as a true candidate of the pattern-growth strategy. Experimental results of the frequent subtree mining algorithms such as PrefixTreeISpan utilizing the *pattern-growth* strategy shows improved runtime over the *generate-and-test* algorithms such as FreqT(Zou et al. 2006b).

With the increase in the dataset size there is an explosion in the number of frequent subtrees generated. In order to reduce the number of subtrees without any information loss, two popular concise representations namely maximal and closed were proposed. Some of the algorithms which generate these concise representations are PathJoin (Xiao and Yao 2003) and CMTreeMiner(Chi et al. 2004b). PathJoin (Xiao and Yao 2003) utilize a compact data structure called FST-forest to generate only the maximal frequent subtrees. PathJoin (Xiao and Yao 2003) faces a serious disadvantage as it does not utilize the maximal subtree generation to improve the performance. This is due to the fact that the pruning of frequent subtrees is applied as a post-processing step. On the other hand, CMTreeMiner(Chi et al. 2004b) was proposed to generate both closed and maximal frequent structures efficiently. As indicated by (Termier et al. 2005) that CMTreeMiner fails to produce satisfactory results for trees that have the high branching factor.

A parallel field to closed frequent subtree mining is the closed frequent subsequences. These closed frequent subsequences are extracted by mining sequential datasets. Some of the popular closed frequent sequence mining algorithms are BIDE(Wang and Han 2004) and CloSpan(Yan, Han and Afshar 2003). In contrast to sequences, trees have branches and hence the techniques to employ the closure property for sequential patterns could not be applied to trees. The sequences include extensions only in either the forward or backward direction in the same branch, it is difficult to utilize sequential pattern mining techniques to trees as it contains several branches.

It is evident from the previous research works that the *generate and test* algorithms are not an efficient vehicle for exploiting the full strength of closure. Additionally the existing techniques for itemsets and sequential patterns frequent mining cannot be naturally extended to deploy the closure property due to the structure of the trees. The impressive performance improvement obtained for the pattern-growth based techniques, PrefixTreeISpan and PrefixTreeESpan, as reported in (Zou et al. 2006b; Zou et al. 2006a) over the generate-and-test based techniques, FreqT and TreeMiner, inspired us to investigate the effect of deploying closure on the pattern-growth technique.

In this paper, we will be focusing only on the frequent induced subtrees and not on the emedded subtrees. The reason for this choice is two fold: Firstly, the application of the closure property on the frequent embedded subtrees will not reduce the result set significantly as the ancestor-descendant relationship is maintained in the frequent embedded subtrees. Secondly, the application of the closure property incurs a huge overhead due to the numerous closure checks required. Hence, in this paper we will focus only on generating the closed frequent induced subtrees using the *pattern-growth* technique. The following section details about the description of our proposed method PCITMiner (**P**refix-based **C**losed **I**nduced **T**ree Miner).

4 PCITMiner - Closure using Prefix Pattern Growth

In this section, we introduce PCITMiner for mining the closed induced subtrees from a tree dataset, *D*. Before we go into the details of PCITMiner, we will briefly introduce how the frequent induced subtrees are generated using the pattern growth technique.

Tree Id	Pre-order string of Trees
1	ABC-1D-1-1EF-1-1-1
2	ABC-1-1D-1-1
3	ABC-1-1C-1-1
4	ABC-1-1C-1-1

Table 1: Tree dataset

We will be using the tree dataset, *D* provided in Table 1 as our running example in this paper. The tree dataset, *D* contains Tree Ids and the trees which are represented in the pre-order string format as defined in Section 2. The Tree Id in Table 1 is the unique id for the trees in *D*. In this tree dataset, D, we have four trees and they contain the nodes labelled as A, B, C, D, E and F. The pictorial representation of the tree with Tree Id 1 is illustrated in Figure 2(a). This tree dataset is mined for the frequent subtrees with a *min_supp* of 2.

4.1 The frequent subtree generation

As we are using the pattern growth technique for the frequent subtree generation, we will explain the prefix-based pattern growth technique where the patterns are subtrees. There are three phases involved in the prefix-based frequent subtree generation and they are:

1. The *1-Length* frequent subtree generation
2. Projecting the dataset using prefix trees
3. Mining the projected instances dataset

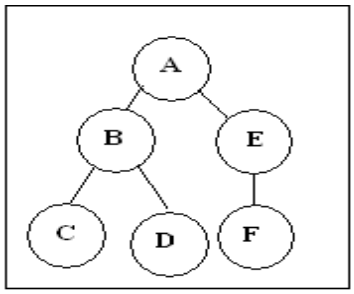
4.1.1 The *1-Length* frequent subtree generation

The prefix-based subtree growth technique starts with a scan of the tree dataset, *D* to determine the *1-Length* frequent subtrees. A *1-Length* frequent subtree contains only one node and is represented using the following format $Sub_X = (<X^a – 1>: Supp)$ where *X* represents the node label, the superscript '*a*' specifies the position in the pre-order traversal of the subpattern Sub_X, '*–1*' is used for the end flag for the node labelled *X,* and *Supp* is the support of the subtree. For a *K-Length* subtree, the representation can be obtained by replacing *X* in the Sub_X with the pre-order traversal and including a superscript for the pre-order positions. Hence the subtree representation is $(<X^a\ Y^b\ Z^c\ -1 \ldots K^n\ -1\ -1>: Supp)$ where

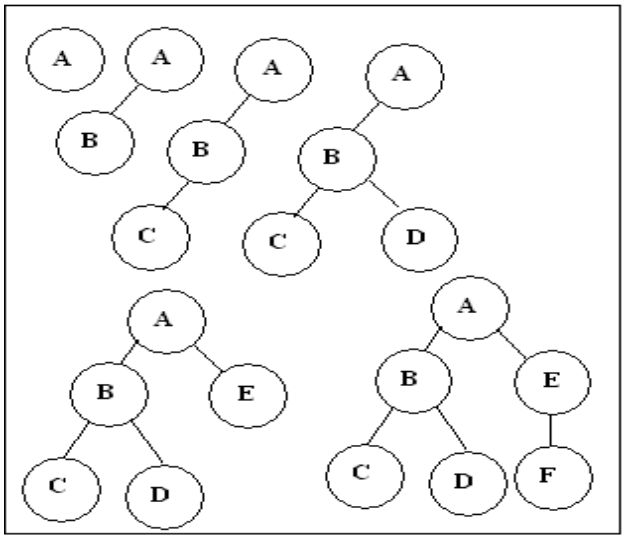
X, *Y*, *Z* and *K* are node labels and superscripts *a, b, c...n* are the increasing positions of nodes in the pre-order traversal of the subtree.

By scanning the tree dataset, D in Table 1 for a given *min_supp* of 2, the four *1-Length* frequent subtrees generated are (<A^1-1>:4), (<B^1-1>:4), (<C^1-1>:4), (<D^1-1>:2) where A, B, C and D are node labels. The superscript '1' on the node labels represents its position in the pre-order traversal of each of the subtrees. The '–1' in each of the subtrees represent the end flag for the node. Finally, the number after a ':'(colon) gives support of each of the subtrees. The subtrees having node labels A, B or C individually has a support of 4 and the subtree having node labelled D has a support of 2.

The tree dataset, D is scanned again and partitioned into four subsets using each of the four subtrees serving as the prefix-tree. Using the definition of the prefix-tree in (Zou et al. 2006b) we will explain what is meant by a prefix-tree using the first tree in our example dataset D from Table 1.



(a)



(b)

Figure 2: Prefix Trees of Tree T (in (a))

Definition 1 (Prefix-Tree)

Let there be a tree *T* with *m* nodes, *T′* be a tree with *n* nodes, where $n \leq m$. The pre-order scanning of tree *T* from its root until its *n-th* node results in a tree *T′*. If the tree *T′* is isomorphic to tree *T* then *T′* is called the prefix of Tree *T* (Zou et al. 2006b).

The Figure 2(b) shows the prefix-trees for the tree T illustrated in Figure 2(a). The 6 prefix-trees containing 1, 2, 3, 4, 5 and 6 nodes are identified for the tree T.

4.1.2 Projecting the dataset using the prefix trees

The next step in this process involves projecting the dataset using the prefix trees generated. Consider the example tree dataset, *D* in Table 1, there are four prefix-trees having a single node. Using these four 1-node prefix-trees, the dataset *D* is partitioned into four prefix-projected datasets namely (i) <A^1 -1 >:4) (ii) (<B^1-1>:4) (iii) (<C^1-1> :4) and (iv) (<D^1-1>:2). To build the prefix-projected dataset for a given prefix-tree *T′*, every tree in *D* is checked whether it contains the prefix-tree *T′*. If a tree *T* contains the prefix-tree *T′*, then the projected instance of T is included in the *T′*-prefix-projected instances dataset.

Definition 2 (The Prefix projected instance)

Consider a tree dataset $D = \{T_1, T_2, T_3, \ldots, T_n\}$ and a prefix subtree *T′* with *n* nodes. If there exists a tree $T_x \in D$ with *m* nodes which contains the prefix-tree *T′*, then the *T′*-prefix projected instance of T_x is the pre-order scanning of T_x from $n+1$ node to *m*.

Tree Id	Pre-order string of Trees
1	BC-1D-1 -1
2	BC-1-1 D-1
3 3	BC-1-1 C-1
4 4	BC-1-1 C-1

Table 2: <A^1-1> projected instances dataset

Tree Id	Pre-order string of Trees
1	C-1D-1
2	C-1
3	C-1
4	C-1

Table 3: <B^1-1> projected instances dataset

Tree Id	Pre-order string of Trees
1	C-1 D-1
2	C-1
3	C-1
4	C-1

Table 4: <A^1B^2-1-1> projected instances dataset

Tree Id	Pre-order string of Trees
1	C-1
2	C-1

Table 5: <A^1B^2-1C^3-1-1> projected instances dataset

Tables 2, 3, 4, and 5 provide the projected instances dataset of the prefix-trees <A^1-1>, <B^1-1>, <A^1B^2-1-1>, <A^1B^2-1C^3-1-1> respectively. To improve the efficiency,

the projected instances from the infrequent *Length-1* subtrees are eliminated. It can be noted in Table 2 that the tree with Tree Id1 does not contain the nodes E and F as they are infrequent and hence they were eliminated in projection. The generated projected instances are mined using the technique detailed in the following subsection.

4.1.3 Mining the projected instances dataset

As a next step in the prefix-pattern growth, each of the projected instances dataset is mined to identify the Growth Element (GE) (Zou et al. 2006b), which is defined as follows.

Definition 3: ***Growth Element (GE)***
Given two trees T' and T with m and m+1 nodes respectively, where T' is the prefix of T. If there occurs a node n in Tree T but not in T' then the node n is the Growth Element (GE) of T' w.r.to T.
If there is any *frequent* GE then the corresponding projection is partitioned and mined recursively until there are no more frequent GEs. For instance, for the partitioned dataset $<A^1$ -1 > provided in Table 2, the GEs are nodes labelled B, C and D as they occur as first nodes in the projected instance. The support of GEs, B, C and D is 4, 2 and 1 respectively in the $<A^1$ –1 > prefix-projected dataset. Hence only B and C are frequent GEs. Since the mining process outputs the induced subtrees, the position of nodes is important in counting the support. For example, the node labelled D occurs twice in the dataset in Table 2 but it occurs in different positions that is why it is not frequent. In other words, the subtrees should have parent-child relationship. In Tree Id 1, the parent of D is B and in the second tree (Tree Id 2), the parent of D is A. Hence, the support of D is 1 in the $<A^1$ –1 > prefix-projected dataset. Using the two GEs, B and C, two separate projections are constructed and mined for the frequent subtrees. Tables 4 and 5 give the projection for $<A^1B^2$ –1 –1 > and $< A^1B^2$-$1C^3$-1-1> respectively.

4.2 Closure

So far we have seen how the frequent subtrees are generated using the prefix-pattern growth technique. Table 6 lists the frequent subtrees using this approach. It can be seen from Table 6 that subtrees such as ($<A^1$-1 >:4), ($<B^1$-1>:4), ($<C^1$-1>:4), ($<A^1B^2$-1 –1>:4), ($<B^1C^2$-1-1>:4) are subsets of ($<A^1B^2C^3$-1-1-1>: 4) with the same support. Hence, instead of generating all the frequent subtrees, only a superset of the frequent subtrees with the same support can be represented as output. By doing so, the number of the frequent induced subtrees is reduced by eliminating only the redundant frequent subtrees and hence there is no information loss. This property of reducing the redundant frequent subtrees is called as the closure property as discussed in Section 2. From Table 6, the subtree ($<A^1C^2$-1-1>: 2) $\subseteq$ ($<A^1B^2C^3$-1 -$1C^4$-1-1>: 2), and hence using the closure property the subtree ($<A^1C^2$-1-1>: 2) can be safely removed from the result set. As the node labelled D is not included in other closed frequent induced subtrees, subtree ($<D^1$-1>:2) is included in the output.

Number of nodes	Frequent Subtrees
1	($<A^1$-1 >:4), ($<B^1$-1>:4), ($<C^1$-1>:4), ($<D^1$-1>:2)
2	($<A^1B^2$-1 –1>:4), ($<B^1C^2$-1-1>:4), ($<A^1C^2$-1-1>:2)
3	($<A^1B^2C^3$-1-1-1>: 4) ($<A^1B^2$-$1C^3$-1-1>: 2)
4	($<A^1B^2C^3$-1 -$1C^4$-1-1>: 2)

Table 6: Frequent induced subtrees from prefix-pattern growth algorithm

Table 7 summarizes the closed frequent induced subtrees with only 3 closed frequent induced subtrees in comparison to 10 frequent induced subtrees (as shown in Table 6). On comparing Tables 6 and 7 it is interesting to note that closure has reduced the number of frequent induced subtrees by three-fold.

Number of nodes	Frequent Subtrees
1	($<D^1$-1>:2)
3	($<A^1B^2C^3$-1-1-1>: 4)
4	($<A^1B^2C^3$-1 -$1C^4$-1-1>: 2)

Table 7: Closed frequent induced subtrees

Now the challenge is to impose closure on the frequent induced subtrees using the prefix-based subtree mining. A naïve approach to impose closure is to first generate all the frequent induced subtrees and then eliminate the subtrees based on their support by checking the closure property, as shown in Tables 6 and 7. It is an expensive task when there are a large number of frequent subtrees generated and hence, it is essential to identify an efficient method, which provides the closed result set. There are a number of approaches proposed in the frequent itemset and sequential mining (Wang and Han 2004; Yan et al. 2003). Unlike, the itemset or sequential mining, trees have branches and hence we cannot apply closure using these techniques. Hence, we propose two methods to apply closure efficiently and they are:

1. Search Space reduction using the backward scan
2. Bi-directional Extension Closure checking

4.2.1. Search space reduction using the backward scan

This technique does a backward scan to reduce the search space using the following lemma:

Lemma 1:
Let there be two *1-length* frequent subtrees L_k and L_k' in a given tree dataset, D. If L_k' is the parent node of L_k in all trees in D then the projection of L_k is stopped as the parent node L_k' will include all the subtrees generated using the prefix-tree L_k.
Using the running example tree dataset D in Table 1, we will explain, how to reduce the search space using the backward scan technique. This technique is applied after

the first scan of the dataset where the *1-Length* frequent subtrees are known. The *1-Length* frequent induced subtrees are (<A^1-1 >:4), (<B^1-1>:4), (<C^1-1>:4), (<D^1-1>:2). As the node labelled A is a root node in all the trees it is not checked for its parents. Hence, this technique is applied for the subtrees (<B^1-1>:4), (<C^1-1>:4) and (<D^1-1>:2).

The checking of the parent node of <B^1-1> in each of the trees in the tree dataset *D* reveals that the parent node is <A^1-1> in all the trees. This information state that the parent node of <B^1-1> (i.e. <A^1-1>) and <B^1-1> have the same support. Consequently, <B^1-1> can be pruned from growing since the projections for the parent node <A^1-1> will include the projections for <B^1-1>. By doing so, the number of subtrees and the number of projections required are reduced. Due to the reduced search space, the efficiency of the algorithm is improved.

4.2.2. The Bi-directional Extension Closure checking

After reducing the search space using the backward scan, there occurs some of the subtrees which are not closed. In order to check the closeness of the generated frequent subtrees, the bi-directional extension closure checking is performed.

According to the definition of a frequent closed induced subtree, a prefix-tree, $T_p=e_1,e_2,...e_n$ is non-closed if there exist at least one extension event, e' which can be used to create a prefix-tree T_p' having the same support as that of T_p. The prefix-tree T_p can be extended in the following ways:

1. *Predecessor node extension as in* $T_p' = e_1,e_2,...,e_n\, e'$
2. *Internal node extension as in* $T_p'= e_1,e_2\, e',...,e_n$
3. *Successor node extension as in* $T_p'= e_1,e_2,... e',e_n$

The bi-directional extension closure checking involves two events namely the *forward-extension event* and the *backward-extension event*. With reference to the event *n* given by e_n, in the situation 1, e' occurs after the event e_n and hence it is a *forward extension event*. On the other hand, in the situation 2 and 3, e' occurs before the event e_n and hence it is a *backward extension event*.

Theorem 1:

If there exists neither the forward-extension event nor the backward extension event in regard to a prefix-tree T_p' then T_p' must be a closed frequent subtree.

A naive approach to check whether there occurs any forward-extension closure checking is enumerating all the frequent sub-trees and then checking their support. However, this is an expensive operation due to very large number of checks required. The following lemma is utilised to check for the forward-extension event efficiently.

Lemma 2: Forward-extension event

For a prefix tree T_p', its complete set of forward-extension events is equivalent to the set of its frequent GEs whose supports are equal to the support of T_p'. If any of the GE for given projection has same support as T_p' then T_p' is not closed.

Using the running example provided in Table 1, the GE is *C* for the prefix-tree <A^1B^2-1-1>. The support of <A^1B^2-1-1> is 4 and the support of the GE *C* is 4 and hence <$A^1B^2C^3$-1-1-1> is not closed. On the other hand, consider the prefix-tree <$A^1B^2C^3$-1-1-1> having the GE *C*. The support of <$A^1B^2C^3$-1-1-1> is 4 and the support of the GE *C* is 2 and hence <$A^1B^2C^3$-1-1-1> may be closed. We say <$A^1B^2C^3$-1-1-1> may be closed, as we need to check for closure using the backward extension event to confirm the closure. This forward event checking is not a computational expensive step and hence it is used for reducing the number of closed frequent induced subtrees. In order to check for the backward extension event, the following lemma is used.

Lemma 3: Backward-extension event

If there exists a prefix-tree T_p with *m* nodes and a prefix-tree T_p' with the common *m* nodes and an additional node *b* having the same support as that of T_p then T_p is not closed and *b* is a backward extension event w.r.to T_p

There are two types of backward-extension events:

1. The parent extension of GE
2. The sibling extension of GE

The backward extension event to GE is the extension of the parent of GE and hence it is handled by the backward scan technique. On the other hand, the backward extension to sibling extension is the extension of sibling nodes of GEs. For instance, in the prefix-tree <A^1B^2-1C^3-1-1>, the sibling extension event is the node labelled C, which is an extension of the node <B2-1> and it is in a different branch resulting in <$A^1B^2C^3$-1C^4-1-1-1>. Unlike the sequential mining, due to the existence of branches in trees, there occurs the sibling extension event in a different branch from <A^1B^2-1C^3-1-1>. Hence, it requires the closure checking across several branches.

In order to efficiently check for closure for backward extension events across several branches, a technique called "maintain-and-test" is deployed to check for closure. A naïve approach to check for closure is to check for all the backward extension events having the same support. However, it is an expensive operation and hence to reduce the number of checks, a parameter, which is the sum of the tree ids, is included to check for closure. To apply this technique, we first check whether for a given subtree *T'*, there exists a backward extension to edge E resulting in *T''* having the same support and sum of tree ids as *T'*. If it exists then they are checked for closure.

Figures 3 and 4 outline the algorithm PCITMiner and the subroutine ***Fre*** respectively. PCITMiner starts with the scan of the database and identifies the *1-Length* frequent subtrees *b*. After finding the *1-Length* frequent subtrees, it employs the backScan property by checking the support of the predecessor of each *b* and the support of each *b*. If they are same, then the projection for *b* could be pruned, as the predecessor for *b* will include *b* in its output. Otherwise, using the recursive subroutine ***Fre*** outlined in Figure 4, recursively identifies all the occurrences of *b* in the dataset *D* to construct <*b-1*> projected database by collecting all the corresponding project-instances in *D*. The subroutine ***Fre*** checks for the forward extension event and the backward extension event against the projected database. This subroutine is recursively called until there are no more frequent GEs to form the projected dataset.

Algorithm PCITMiner

Input: A tree dataset *D*, minimum support threshold (*min_supp*)
Output: All closed induced frequent subtrees

Methods:
1. Scan *D* and find all *1-length* frequent label *b*.
2. For each frequent label *b*
   2.1 If the supp (predecessor of *b*) = = supp(*b*) then Do not project the dataset.
   2.2 else,
   2.2.1 Find all occurrences of *b* in dataset *D*, and construct <*b*−*1*>-projected dataset (i.e. *ProDB(D,<b − 1 >))* through collecting all corresponding Project-Instances in *D*.
   2.2.2 Call ***Fre*** *(< b − 1 >, 1, ProDB(D,< b − 1 >), min sup, supp(b))* to mine the projected dataset and obtain frequent induced subtrees until no more subtrees that could be found.

Figure 3: Algorithm PCITMiner

Function *Fre* (T_p, n, *ProDB(D,* T_p*), min_supp,prepat_supp)*

Input: A prefix-tree T_p, the length of T_p(n), <T_p>-projected dataset(*ProDB(D,*T_p*)*), the minimum support threshold (*min_supp*), the support of the previous pattern used to generate this projected dataset (*prepat_supp*)
Output: *C*: Closed frequent induced subtrees

Methods:
1. Scan *ProDB(D,* T_p *)* once to find all the *1-length* frequent *GEs*($GE_{0, \dots, k}$) according to Lemma 1.
2. output=true.
3. Count the support of all *GEs*.
4. If supp(GE_0 || GE_1,...,|| GE_k) == supp(T_p) then Do not output the subtree, output = false.
5. For each GE_b
   5.1 if GE_b is frequent then
      5.1.1 Extend T_p with *b* to form a subtree pattern T_p'.
      5.1.2 if (output) then Insert T_p' into *C*.
   5.2 else
      5.2.1 Check T_p' for occurrence of any of its subset with the same support and sum of tree ids in the output *C*. If there exists any subset for T_p' then remove the subset of T_p' and insert T_p' into *C*.
6. Find all occurrences of GE_b in *ProDB(D,* T_p*),* construct the <T_p'>-projected database (i.e. *ProDB(D,* T_p' *)*) through collecting all corresponding Project-Instances in *ProDB(D,* T_p*)*.
7. Call ***Fre***(T_p' , n + 1, *ProDB(D,* T_p'), *min_supp prepat_supp*)

Figure 4: Recursive function *Fre*

5 Experimental evaluation

All the experiments were conducted on the Intel Pentium-4 PC with 2.39GHz processor and 1GB main memory, running Windows XP. Both the algorithms PrefixTreeISpan and PCITMiner were written in C++ with the STL library support and compiled with the Microsoft Visual C++ .Net compiler. The experiments were conducted on the synthetic datasets generated.

The Zaki's treegenerator[1] has been often used to generate the synthetic datasets for benchmarking the tree mining algorithms. Using the Zaki's tree generator there are two datasets generated namely the F5 and D10 datasets with the parameters as indicated in Table 8 where "f" represents the fan out factor, "d" the depth of the tree, "n" the number of unique labels for the trees, "m" the total number of nodes in a parent tree and "t" indicates the number of trees.

Name	Description
F5	-f 5 -d 10 -n 100 -m 100 -t 100000
D10	-f 10 -d 10 –n 100 -m 100 -t 100000

Table 8: Datasets and their parameters

Studies have indicated that the performance of some of the existing closed frequent subtree mining algorithm degrades for datasets having a high branching factor (Termier et al. 2005). To evaluate the performance of PCITMiner with high branched trees, two datasets F5 and D10 with varied branches, fan out factors of 5 and 10 respectively, are generated.

The proposed algorithm, PCITMiner is compared with the prefix-based pattern-growth algorithm PrefixTreeISpan (Zou et al. 2006b) to show the benefit of closure. The output of the PrefixTreeISpan algorithm is frequent induced subtrees. Experimental studies on PrefixTreeISpan (Zou et al. 2006b) with FreqT (the *generate-and-test* based frequent subtree mining method) already has shown that PrefixTreeISpan outperforms FreqT (Zou et al. 2006b). So in this paper, we do not conduct any empirical analysis with the generate-and-test method. As the objective of this study is to apply closure on the pattern growth algorithms hence CMTreeMiner is also not used as a benchmark as the latter algorithm is based on the candidate "*generate-and-test*" approach.

Figures 5 and 6 presents the experimental results on the number of subtrees and the run time in seconds for PCITMiner and PrefixTreeISpan on the F5 dataset. Figure 5 reveals that the PCITMiner reduces the number of subtrees by about three-fold in comparison to PrefixTreeISpan. The benefit is larger for the relatively lower support-threshold (where a large number of subtrees are generated).

[1] http://www.cs.rpi.edu/~zaki/software

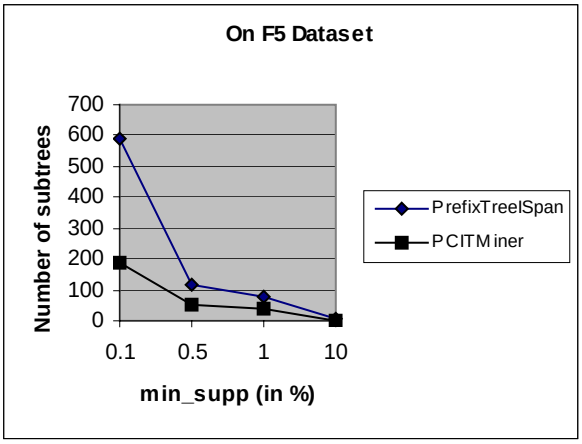


Figure 5: Number of subtrees of PCITMiner and PrefixTreeISpan against various min_supp on F5 dataset

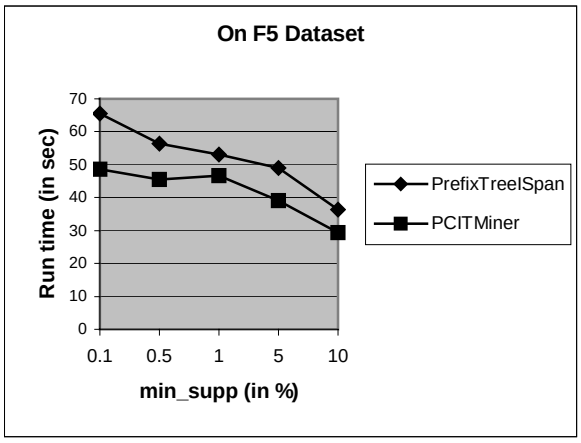


Figure 6: Run times of PCITMiner against PrefixTreeISpan against various min_supp on F5 dataset

Figure 6 reveals that with the reduced number of subtrees, PCITMiner mines the frequent subtrees faster than the base algorithm PrefixTreeISpan. The improvement obtained in F5 dataset can be attributed to the number of back scan pruning. Figure 7 shows the increase in the number of projections pruned with the reduced support threshold.

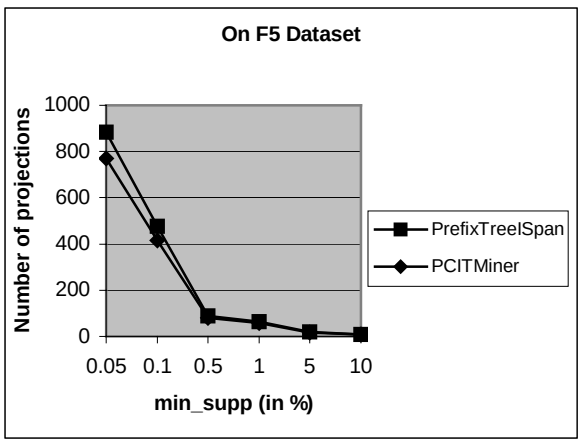


Figure 7: Number of projections of PCITMiner against PrefixTreeISpan against various min_supp on F5 dataset

Figures 8 and 9 presents the experimental results on the run time in seconds and the number of subtrees for PCITMiner and PrefixTreeISpan on D10 dataset respectively. The PCITMiner achieves improved performance over PrefixTreeISpan by reducing the number of subtrees by about seven-fold at lower support values.

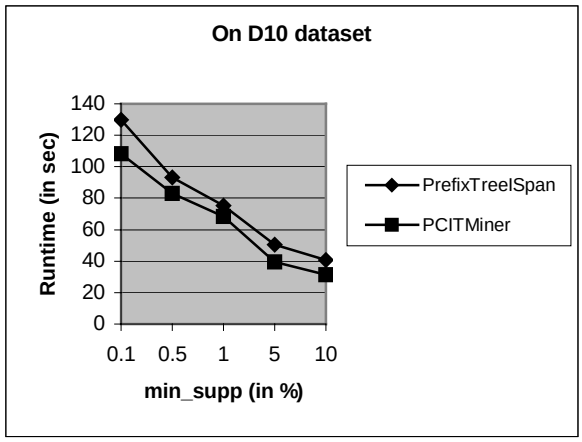


Figure 8: Run times PCITMiner against PrefixTreeISpan against various min_supp on D10 dataset

The comparison of experimental results of the F5 and D10 datasets clearly indicates that PCITMiner remains unaffected with high branched trees (large fan out factor). PCITMiner shows the improved performance in run time as well as reducing the number of subtrees efficiently in both the data sets. Moreover, the saving in terms of the number of output patterns is more apparent with the high-branched trees.

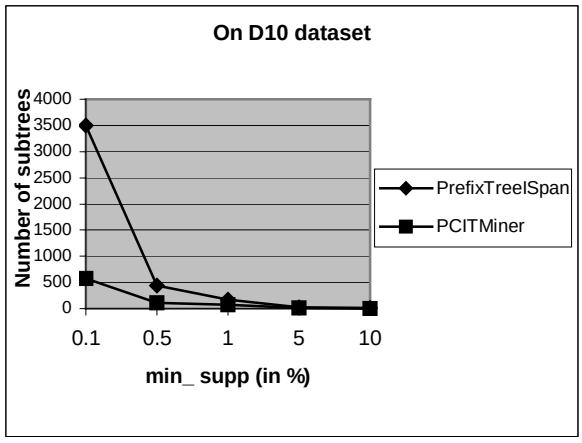


Figure 9: Number of subtrees of PCITMiner and PrefixTreeISpan against various min_supp on D10 dataset

6 Conclusions and Future work

In this paper, we have proposed PCITMiner for generating the closed frequent induced subtrees using the pattern-growth technique. The experimental results clearly indicate that PCITMiner performs faster and produces reduced number of frequent subtrees without any information loss. Contrary to the existing closed frequent subtree mining algorithms the proposed

algorithm PCITMiner performs efficiently for high branched trees. We would like to apply this closure technique for embedded subtrees and to graph-based frequent mining as a future work.

7 Acknowledgement

We would like to thank Lei Zou at HuaZhong University of Science and Technology for kindly providing us the base algorithm PrefixTreeISpan.

8 References

Agrawal, R., H. Mannila, R. Srikant, H. Toivonen and A. I. Verkamo.(1996): Fast discovery of association rules. In *Advances in knowledge discovery and data mining, 1996.,*307-328: American Association for Artificial Intelligence.

Asai, T., K. Abe, S. Kawasoe, H. Arimura, H. Satamoto and S. Arikawa.(2002): Efficient substructure discovery from large semi-structured data. *2nd SIAM International Conference on Data Mining*.

Asai, T., H. Arimura, T. Uno and S. Nakano.(2003): Discovering Frequent Substructures in Large Unordered Trees. *The 6th International Conference on Discovery Science*.

Chi, Y., S. Nijssen, R. R. Muntz and J. N. Kok.(2005):Frequent Subtree Mining- An Overview. *Fundamenta Informaticae*, **66**: 161-198. IOS Press.

Chi, Y., Y. Yang and R. R. Muntz.(2004a): HybridTreeMiner: an efficient algorithm for mining frequent rooted trees and free trees using canonical forms. *Scientific and Statistical Database Management, 2004. Proceedings. 16th International Conference on*, 11-20.

Chi, Y., Y. Yang, Y. Xia and R. R. Muntz.(2004b): CMTreeMiner: Mining both closed and maximal frequent subtrees. In *In The Eighth Pacific Asia Conference on Knowledge Discovery and Data Mining (PAKDD'04)*.

Han, J., J. Pei and X. Yan.(2005): Sequential Pattern Mining by Pattern-Growth: Principles and Extensions. In *Foundations and Advances in Data Mining*.

Pei, J. 2002. *Pattern-growth methods for Frequent pattern mining*, School Of Computing Science, Simon Fraser University, Burnaby, British Columbia, Canada.

Tatikonda, S., S. Parthasarathy and T. M. Kur.(2006): TRIPS and TIDES: new algorithms for tree mining. *Proceedings of the 2006 ACM CIKM International Conference on Information and Knowledge Management*, 455-464. Arlington, Virginia, USA: ACM.

Termier, A., M.-C. Rousset and M. Sebag.(2002): TreeFinder: a first step towards XML data mining. *ICDM 2002. Proceedings. 2002 IEEE International Conference on Data Mining, 2002.*, 450-457.

Termier, A., M.-C. Rousset, M. Sebag, K. Ohara, T. Washio and H. Motoda.(2005): Efficient mining of high branching factor attribute trees. *Proc. Fifth IEEE International Conference on Data Mining*.

Wang, C., M. Hong, J. Pei, H. Zhou, W. Wang and B. Shi.(2004): Efficient Pattern-Growth Methods for Frequent Tree Pattern Mining. In *Advances in Knowledge Discovery and Data Mining*.

Wang, J. and J. Han.(2004): BIDE: Efficient Mining of Frequent Closed Sequences. In *Proc. 20th International Conference on Data Engineering*: IEEE Computer Society.

Xiao, Y. and J.-F. Yao.(2003): Efficient data mining for maximal frequent subtrees. *Proc. Third IEEE International Conference on Data Mining(ICDM03)*, 379-386.

Yan, X., J. Han and R. Afshar.(2003): CloSpan: Mining Closed Sequential Patterns in Large Datasets. *Proc. SIAM International Conference on Data Mining*.

Zaki, M. J.(2002): Efficiently mining frequent trees in a forest. *Proc. Eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 71-80. Edmonton, Alberta, Canada: ACM Press.

Zaki, M. J.(2005):Efficiently mining frequent trees in a forest: algorithms and applications. *IEEE Transactions on Knowledge and Data Engineering*, **17 (8)**: 1021-1035.

Zou, L., Y. Lu, H. Zhang and R. Hu.(2006a): PrefixTreeESpan: A Pattern Growth Algorithm for Mining Embedded Subtrees. In *Web Information Systems , WISE 2006*.

Zou, L., Y. Lu, H. Zhang, R. Hu and C. Zhou.(2006b): Mining Frequent Induced Subtrees by Prefix-Tree-Projected Pattern Growth. *Proc. Seventh International Conference on Web-Age Information Management Workshops, 2006. WAIM '06.*, 18.

News Aware Volatility Forecasting: Is the Content of News Important?

Calum S. Robertson
Information Research Group
Faculty of Information Technology
Queensland University of Technology
2 George Street, Brisbane, QLD, Australia 4000
cs.robertson@qut.edu.au

Shlomo Geva
Information Research Group
Faculty of Information Technology
Queensland University of Technology
2 George Street, Brisbane, QLD, Australia 4000
s.geva@qut.edu.au

Rodney C. Wolff
School of Economics and Finance
Faculty of Business
Queensland University of Technology
2 George Street, Brisbane, QLD, Australia 4000
r.wolff@qut.edu.au

Abstract

The efficient market hypothesis states that the market incorporates all available information to provide an accurate valuation of the asset at any given time. However, most models for forecasting the return or volatility of assets completely disregard the arrival of asset specific news (i.e., news which is directly relevant to the asset). In this paper we propose a simple adaptation to the GARCH model to make the model aware of news. We propose that the content of news is important and therefore describe a methodology to classify asset specific news based on the content. We present evidence from the US, UK and Australian markets which show that this model improves high frequency volatility forecasts. This is most evident for news which has been classified based on the content. We conclude that it is not enough to know when news is released, it is necessary to interpret its content.

Keywords. Stock Market, News, Document Classification, Volatility, Forecast.

1. Introduction

The efficient market hypothesis states that the market incorporates all available information to provide an accurate valuation of the asset at any given time. There is large body of evidence that assets tend to react to public information, most often when the information contains a shock. This evidence includes the reaction to public information in the form of newspaper/magazine/real-time source (e.g. Cutler et al. 1989, Goodhart 1989, Goodhart et al. 1993, Melvin and Yin 2000, Mitchell and Mulherin 1994, Mittermayer 2004), macroeconomic announcements (e.g. Almeida et al. 1998, Ederington and Lee 1993, 1995, 2001, Graham et al. 2003, Kim et al. 2004, Nofsinger and Prucyk 2003), analyst recommendations Hong et al. 2000, Michaely and Womack 1999, (e.g. Womack 1996), and weather reports (e.g. Roll 1984).

Ederington and Lee (1993) found that volatility on Foreign Exchange and Interest Rate Futures markets increases within one minute of a macroeconomic news announcement, and the effect lasts for about 15 minutes. Ederington and Lee (1995) determined that the same markets begin to react within 10 seconds of macroeconomic news announcements, with weak evidence that they tend to overreact to news within the first 40 seconds after news, but settle within 3 minutes. Graham et al. (2003) established that the value of stocks on the S&P 500 index are influenced by scheduled macroeconomic news, however, they did not investigate any intraday effect. Nofsinger and Prucyk (2003) concluded that unexpected bad macroeconomic news is responsible for most abnormal intraday volume trading on the S&P 100 Index option.

Despite strong evidence that the stock market does react to macroeconomic news, there is far more asset specific news, i.e., news which is directly relevant to the asset, than macroeconomic news. Furthermore, unlike macroeconomic news, most asset specific news is not scheduled and therefore investors have not formed their own expectation, or adopted analysts' recommendations about the content of the news. Mittermayer (2004) investigated the effect of Press Announcements on the New York Stock Exchange and the NASDAQ and determined that the content of news can be used to predict, with reasonable accuracy, if the market will exhibit high return within 60 minutes of the announcement. Unfortunately press announcements are only a fraction of asset specific news, so further investigation is required to determine how the stock market reacts, if at all, to this type of news.

The Generalised Autoregressive Conditional Heteroskedasticity (GARCH) model introduced by Bollerslev (1986) has been shown to be a reliable model for forecasting the volatility of an asset. However, like virtually all volatility forecasting models, it completely disregards the impact of public information. Kalev et al. (2004) found that the forecast accuracy of GARCH(1,1) for 30 minute returns can be improved by factoring in the number of asset specific documents released to the market in the previous 30 minutes. Furthermore they found that the forecast accuracy could be further improved by restricting the news based on how the Australian Stock Exchange (ASX) categorised the news (e.g. Progress Report, Dividend Announcement, Mergers and Acquisitions). Whilst the ASX may classify news, it is not safe to assume that every asset specific news

document for assets throughout the world will be classified in the same fashion at the time of their release. Therefore it is advisable to use an automated form of news classification, which can be applied to news from any source.

In this paper we propose a modification to the GARCH model proposed by Bollerslev (1986), to handle the arrival of asset specific news. Furthermore we describe an automated method to classify the news, which can be used to limit the number of documents which this model processes. Finally we demonstrate how this model can improve the volatility forecast accuracy using a large asset base and high frequency data.

2. Data

All data for this research were obtained using the Bloomberg Professional® service. The dataset consists of stocks which comprised the S&P 100, FTSE 100, and ASX 100 indices as at July 2005 and continued to trade through to November 2006, which is a total of 283 stocks. For each stock the Trading Data, and News were collected for the period beginning May 2005 through to and including the October 2006. There are over 500,000 documents (news articles) in this dataset, which we believe to be the largest used for the types of experiments we conduct.

2.1. Trading Data

The set defined in Eq. (1) consists of each distinct minute (z) where trading occurred for the stock (s), within all minutes for the period of data collection (T_A). For each minute ($d_{(s,z)}$) the average price ($p_{(s,z)}$) for trades during that minute are stored.

$$I_{(s)} = \{I_1, I_2, ..., I_m\} \mid I_{(s,z)} = \left(d_{(s,z)}, p_{(s,z)}\right) \wedge z \in T_A \quad (1)$$

However, only business time scale (minutes which occurred during business hours for the market on which the stock trades) is of interest. Furthermore it is necessary to have a homogenous time series (i.e., an entry for every business trading minute for the stock, regardless of whether any trading occurred). Therefore the date ($D_{(s)}$) and price ($P_{(s)}$) time series are produced for all minutes in the business time scale (T_B) with the definitions in Eqs. (2) and (3). The price at time t is defined as the price of the last actual trade for the stock prior to or at the given time. Note that if the stock was suspended from trading for a whole day then the day is excluded from T_B.

$$D_{(s)} = \{D_1, ..., D_n\} \mid D_{(s,t)} > D_{(s,t-1)} \wedge D_{(s,t)} \in T_B \wedge T_B \subseteq T_A \quad (2)$$

$$P_{(s)} = \{P_1, ..., P_n\} \mid P_{(s,t)} = \left(p_{(s,z)} \mid z = \max\left(z \mid d_{(s,z)} \le D_{(s,t)}\right)\right) \quad (3)$$

2.2. News

The news search facility within the Bloomberg Professional® service was used to download all relevant documents for each stock within the dataset. These documents include Press Announcements, Annual Reports, Analyst Recommendations and general news which Bloomberg has sourced from over 200 different news providers.

The set defined in Eq. (4) consists of each distinct news document (λ) for the stock (s) and contains the time ($d_{(s,\lambda)}$) and content ($C_{(s,\lambda)}$) of the document. Note that we allow the market time to react to news by ignoring any document which occurred within the last $\Delta\tau$ minutes of a business day (i.e., $time(d_{(s,\lambda)}) < max(time(T_B)) - \Delta\tau$). Furthermore we ignore the first $\Delta\tau$ minutes of a business day as we expect investors are more focussed on opening their positions for the day rather than reading the latest news (i.e., $min(time(T_B)) + \Delta\tau \le time(d_{(s,\lambda)})$).

$$A_{(s,\Delta\tau)} = \{A_1, A_2, ..., A_p\} \mid A_{(s,\Delta\tau,\lambda)} = \left(d_{(s,\lambda)}, C_{(s,\lambda)}\right) \wedge d_{(s,\lambda)} \in T_B \wedge \min\left(time(T_B)\right) + \Delta\tau \le time\left(d_{(s,\lambda)}\right) < \max\left(time(T_B)\right) - \Delta\tau \quad (4)$$

All documents are pre-processed to remove numbers, URLs, email addresses, meaningless symbols, and formatting. Each term in the content $C_{(s,\lambda)}$ of the document is stemmed using the Porter stemmer algorithm (Porter 1980). The Porter stemmer removes suffixes from words, using strict rules which apply to the English language, such that words with the same stem are considered to be the same word. For example the stems of "finance", "finances", "financed", and "financing" are the same. Stemming is performed to reduce the number of terms which need to be investigated, and to help to find similar documents. The stemmed term index defined in Eq. (5) is created with the stemmed terms which appear in the document ($S_{(s,\lambda,\omega)}$), and the number of times they appear within the document ($SC_{(s,\lambda,\omega)}$), where ω is the stemmed term identifier.

$$C_{(s,\lambda)} = \{T_1, T_2, ..., T_q\} \mid T_{(s,\lambda,\omega)} = \{S_{(s,\lambda,\omega)}, SC_{(s,\lambda,\omega)}\} \wedge SC_{(s,\lambda,\omega)} = \#\{\forall S_{(s,\lambda,\omega)} \in C_{(s,\lambda)}\} \quad (5)$$

3. Methodology

The methodology section is divided into sections titled News Classification, News Aware GARCH, and Measuring Forecast Performance. In the first section we define a classifier which we use to predict whether a document will cause abnormal market behaviour based on its content. In the second section we describe how the classified documents are incorporated into a model to forecast volatility. In the final section we define how we measure the performance of the new model.

3.1. News Classification

In order to classify documents it is first necessary to categorise the documents and determine which documents are of more interest. Building a classifier which predicts whether a document is interesting requires the construction of training and test sets. To ascertain if these documents in the training set have anything in common it is then necessary to analyse the terms contained in the documents, and rank the terms which are most interesting. Subsequently the accuracy of the classifiers must be tested by comparing the predictions of the classifiers with the actual document category. Therefore this section is split into subsections covering document categorisation, training and test sets, term ranking, and classification, and testing.

3.1.1. Document Categorisation

In order to determine the accuracy of a classifier it is necessary to have specific measures of how the market reacts to news. To do so it is necessary to perform time series analysis on the trading data and categorise each document according to how the market behaved shortly after its arrival.

The return time series in Eq. (6) gives the log returns over the period Δt for the stock. The return time series is one of the most interesting to investors as it demonstrates the amount of money which can be made. However, at high frequencies it is impossible to predict returns as the market is far too noisy.

$$R_{(s,\Delta t)} = \{R_1,...,R_m\} \mid R_{(s,\Delta t,t)} = \log\left(P_{(s,t)}\right) - \log\left(P_{(s,t-\Delta t)}\right) \tag{6}$$

Realised volatility given by $\upsilon_{(s,n,\rho,\Delta t)}^{\ 2}$ in Eq. (7) is more commonly used within the finance community to estimate the risk of owning an asset. The variable n defines the number of previous minutes to sum and ρ is the exponent for the return.

$$\upsilon_{(s,n,\rho,\Delta t)} = \{\upsilon_1, \upsilon_2, ..., \upsilon_u\} \mid \upsilon_{(s,n,\rho,\Delta t,t)} = \left[\frac{1}{n}\sum_{j=0}^{n-1} R_{(s,\Delta t,t-j)}^{\ \rho}\right]^{\frac{1}{\rho}} \tag{7}$$

There are many methods used to forecast volatility, though the GARCH (Generalised Autoregressive Conditional Heteroskedasticity) model introduced by Bollerslev (1986) is one of the most common. The GARCH(P,Q) forecast volatility for the stock s, at time t is given by ($\sigma_{(s,\Delta t,P,Q,t)}^{\ 2}$) in Eq. (8). It combines autoregression in the variance with the lagged conditional variance. The variable P is used to define the number of autoregressive components, and Q is used to define the number of lagged conditional variances to include in the forecast. The variable α_0 is a constant, whilst the α's and β's are used to scale the autoregressive and lagged conditional variances respectively.

$$\sigma_{(s,\Delta t,P,Q)} = \{\sigma_1, \sigma_2, ..., \sigma_u\} \tag{8}$$
$$\mid \sigma_{(s,\Delta t,P,Q,t)} = \sqrt{\alpha_0 + \sum_{i=1}^{P} \alpha_i R_{(s,\Delta t,t-i)}^{\ 2} + \sum_{j=1}^{Q} \beta_j \sigma_{(s,\Delta t,P,Q,t-j)}^{\ 2}}$$

The parameters for the model are optimised using the previous month's trading data, to ensure that they are not fitted to the given month's trading conditions. For the calendar month of January parameters which were optimised using all trading data for the stock for the calendar month of December are used. This achieved by maximising the log-likelihood function, given by Eq. (9) for the stock s, where there are n entries in the time series (Dacorogna et al. 2001). The parameters which produce the maximum likelihood function for the given data are chosen.

$$L(\theta) = -\frac{n}{2}\ln(2\pi) - \frac{1}{2}\sum_{t=1}^{n}\left(\ln\left(\sigma_{(s,\Delta t,P,Q,t)}^{\ 2}\right) + \frac{R_{(s,\Delta t,t)}^{\ 2}}{\sigma_{(s,\Delta t,P,Q,t)}^{\ 2}}\right) \tag{9}$$

In our case we are trying to optimise the GARCH model after the arrival of news. Therefore we apply GARCH by calculating the forecast error given by Eq. (10), which is the difference between the forecast (Eq. (8)) and realised volatility (Eq. (7)). This highlights periods where the GARCH model is poor at forecasting the volatility. We use P=Q=3, because we found in previous work that abnormal forecast errors with these parameters have a strong correlation with the arrival of asset specific news.

$$e_{(s,\Delta t,P,Q)} = \{e_1, e_2, ..., e_u\} \mid e_{(s,\Delta t,P,Q,t)} = \sigma_{(s,\Delta t,P,Q,t)}^{\ 2} - \upsilon_{(s,1,2,\Delta t,t)}^{\ 2} \tag{10}$$

We want to categorise documents whose incidence correlates with abnormal forecast errors as interesting. To do so it is necessary to calculate the mean and standard deviation of the forecast error over a given period. The variable M in Eq. (11) defines the average number of trading minutes per month by using the average number of trading minutes per business day for the relevant country, and multiplying by the average number of trading days per month (20).

$$M = 20 \times m \mid \{m_{US} = 390, m_{UK} = 510, m_{AU} = 360\} \tag{11}$$

In Eq. (12) the mean ($\mu_{(s,\Delta t,t)}$) for time t in the forecast error time series $e_{(s,\Delta t,P,Q)}$ is defined by taking the mean value for the M trading minutes which preceded the start of the current trading day. In Eq. (13) the standard deviation ($std_{(s,\Delta t,t)}$) for time t in the forecast error time series $e_{(s,\Delta t,P,Q)}$ is defined by again using the M trading minutes which preceded the start of the current trading day. Note that if a stock was suspended from trading during the last 20 trading days for the stock exchange, only the last 20 days which the stock traded on are used.

$$\mu_{(s,\Delta t,P,Q,t)} = \frac{\sum_{j=t_0-M}^{t_0-1} e_{(s,\Delta t,P,Q,j)}}{M} \tag{12}$$
$$\mid t_0 = \min\left(\left\{\forall T_{(B,i)} \mid time\left(T_{(B,i)}\right) = \min\left(time\left(T_B\right)\right) \wedge T_{(B,i)} \le t\right\}\right)$$

$$std_{(s,\Delta t,P,Q,t)} = \sqrt{\frac{\sum_{j=t_0-M}^{t_0-1}\left(e_{(s,\Delta t,P,Q,j)} - \mu_{(s,\Delta t,j)}\right)^2}{M}} \tag{13}$$
$$\mid t_0 = \min\left(\left\{\forall T_{(B,i)} \mid time\left(T_{(B,i)}\right) = \min\left(time\left(T_B\right)\right) \wedge T_{(B,i)} \le t\right\}\right)$$

The category Ψ of each document in Eq. (4) is calculated using the definition in Eq. (14). If the forecast error within $\Delta\tau$ minutes equals or exceeds δ standard deviations from the mean function value then the document is categorised as interesting (i.e., 1), for same δ. Otherwise it is categorised as uninteresting (i.e., 0).

$$\Psi_{(s,\Delta t,\Delta\tau,P,Q,\delta)} = \{\Psi_1, \Psi_2, ..., \Psi_p\} \mid \Psi_{(s,\Delta t,P,Q,\Delta\tau,\delta,\lambda)} = \tag{14}$$
$$\left(\begin{array}{l} \exists t \mid d_{(s,\lambda)} < t \le d_\lambda + \Delta\tau \\ \wedge\left(\begin{array}{l} e_{(s,\Delta t,P,Q,t)} \ge \mu_{(s,\Delta t,P,Q,t)} + \delta \times std_{(s,\Delta t,P,Q,t)} \\ \vee\, e_{(s,\Delta t,P,Q,t)} \le \mu_{(s,\Delta t,P,Q,t)} - \delta \times std_{(s,\Delta t,P,Q,t)} \end{array}\right) ? 1 : 0 \end{array}\right)$$

3.1.2. Training and Test Sets

The stocks for each country c are grouped together using Eq. (15) to form a large dataset of related stocks. Each document for each stock within each country is then categorised using the forecast error time series with the chosen parameters. Training sets are created by taking N documents, of which R are categorised as interesting (i.e., those which correlated to abnormal behaviour), and the rest are not. The test set is a subset of the documents not included in the training set.

$$G_{(c)} = \{G_1, G_2, ..., G_v\} \quad (15)$$

3.1.3. Term Ranking

A dictionary is created using Eq. (16) for each term which appears in at least one document for a stock in the training set. The term count (d_j), document count (df_j), and interesting document count (r) are stored for each term. The term count d_j is the total number of times the given term appears in all documents in the training set. The document count df_j is the total number of documents which contain the given term. The interesting document count r is the total number of documents which are categorised as interesting in the training set which contain the given term. The subscript η refers to a distinct document within the training set.

A sub-dictionary is formed by taking the top φ terms based on a given term ranking algorithm. For this research we chose three term ranking methods which we will subsequently define.

$$\begin{aligned} &X_{(c,\Delta t,P,Q,\Delta \tau,\delta)} = \{X_1, X_2, ..., X_w\} \mid X_{(c,\Delta t,\Delta \tau,\delta,\eta)} = \\ &\{S_{(c,\Delta t,\Delta \tau,\delta,\eta)}, d_{j(\eta)}, df_{j(\eta)}, r_{(\eta)}\} \\ &\wedge d_{j(\eta)} = \sum SC_{(s,\lambda,\eta)} \mid s \in G_{(c)} \\ &\wedge df_{j(\eta)} = \#\{\forall C_{(s,\lambda)} \mid SC_{(s,\lambda,\eta)} > 0 \wedge s \in G_{(c)}\} \\ &\wedge r_{(\eta)} = \#\{\forall C_{(s,\lambda)} \mid SC_{(s,\lambda,\eta)} > 0 \wedge \Psi_{(s,\Delta t,P,Q,\Delta \tau,\delta,\lambda)} = 1 \wedge s \in G_{(c)}\} \end{aligned} \quad (16)$$

Firstly, we choose the term frequency inverse document frequency (TFIDF) method given by Eq. (17). Note the N in Eq. (17) is the number of documents in the training set. The inverse document frequency helps to bias against terms which occur in every document. The term frequency helps to favour terms which occur frequently. Note that, typically, TFIDF is used to measure the effect of a term within a single document, whilst here it is used to measure the effect of the term within the training set.

$$TFIDF = d_j \times \log_{10}\left(\frac{N}{df_j}\right) \quad (17)$$

Secondly, the binary version of the gain ratio introduced by Quinlan (1993), given by Eq. (19) was chosen. This method selects terms which provide the most information, i.e., splits the data between the classes most effectively. In Eq. (19) $E(R, N)$ is the entropy value (Eq. (18)) for the ratio of interesting documents (R) to documents (N) in the training set. The next part calculates the entropy value for the ratio of interesting documents to documents which contain the term, scaled by the ratio of documents which contain the term. This helps to select terms which occur frequently in interesting documents. The last part of the equation calculates the entropy value for the ratio of uninteresting documents to documents which contain the term, scaled by the ratio of documents which do not contain the term. This helps to select terms which do not occur in interesting documents, i.e., documents which do not contain the term are interesting.

$$E(n,m) = -\left(\frac{n}{m}\log_2\left(\frac{n}{m}\right) + \left(1-\frac{n}{m}\right)\log_2\left(1-\frac{n}{m}\right)\right) \mid n \le m \quad (18)$$

$$GAIN = E(R,N) - \frac{df_j}{N} \times E(r, df_j) - \frac{N - df_j}{N} \times E(df_j - r, df_j) \quad (19)$$

Finally, the BM25 algorithm (Best Match) introduced by Robertson and Spärck Jones (2006) was adapted to get the Average Document BM25 value (ADBM25). This is given by Eq. (20), where k_1 and b are constants, $dl_{(i)}$ is the length of the document i, and $avdl$ is the average document length for documents in the training set. The ADBM25 algorithm is the same as the BM25 algorithm if N were equal to 1, or in other words if there was only one document. The first part of the equation normalises the term frequency by taking into account the length of the document which contains the term and the average document length. This ensures that, if a term occurs frequently in a very long document, it is not given unwarranted significance. The log part of the equation normalises results by factoring in the number of interesting documents which contain the term (r), the number of documents which contain the term (df_j) and the total number of interesting documents (R) and documents (N). This favours terms which provide more information, i.e., splits the two classes most efficiently.

$$ADBM25 = \frac{1}{N}\sum_{i=1}^{N} \frac{(k_1+1)\times d_j}{\left(k_1 \times \left((1-b) + b \times \frac{dl_{(i)}}{avdl}\right)\right) + d_j} \times \log\left(\frac{(r+0.5)(N - df_j - R + r + 0.5)}{(df_j + 0.5)(R - r + 0.5)}\right) \quad (20)$$

3.1.4. Classification

A binary vector is created for each document in the training and test sets where each entry specifies whether the given term (which is a member of the sub-dictionary) occurred in the document. These vectors are used to train and test the C4.5 decision tree introduced by Quinlan (1993), and the support vector machine (SVM) introduced by Vapnik (1999) using the SVM Light Classifier released by Joachims (2007).

The C4.5 decision tree introduced by Quinlan (1993) classifies documents by building a tree where the root node is the term which produces the highest Gain value (Eq. (18)). The root node contains two leaf nodes, the first is for all documents which contain the term and the second is for all documents which exclude the term. The tree is grown by recursively repeating the process at each node on the documents which contain/exclude each term contained in the path directly from the root node to the current node. However, only terms which are contained in the remaining documents are included in the search for the next term.

The support vector machine (SVM) introduced by Vapnik (1999) projects the terms and their values into higher dimensional space (e.g. one dimension per term). It produces a classifier by identifying the hyperplane which most effectively separates the two classes.

3.1.5. Testing

To compare the performance of different classifiers there are several statistical measures which are commonly used. The most important of these is the classification accuracy, given by Eq. (21), which is the ratio between the number of vectors correctly classified (#*TP* is the

number of true positives, and #*TN* is the number of true negatives, and *N* is the number of documents).

$$Accuracy = \frac{\#TP + \#TN}{N} \tag{21}$$

The True Positive Rate also known as Sensitivity, given by Eq. (22) is the percentage of documents whose incidence correlated with abnormal behaviour which were correctly classified.

$$True\ Positive\ Rate = Sensitivity = \frac{\#TP}{\#TP + \#FN} \tag{22}$$

The False Positive Rate which is equivalent to 1 subtract the Specificity, given by Eq. (23), is the percentage of documents whose incidence did not correlate with abnormal behaviour which were incorrectly classified.

$$False\ Positive\ Rate = 1 - Specificity = \frac{\#FP}{\#TN + \#FP} \tag{23}$$

It is common practice when demonstrating the performance of a classifier to plot a Receiver Operating Characteristic (ROC) Curve. This has the True Positive Rate on the Y axis and the False Positive Rate on the X axis.

3.2. News Aware GARCH

In this section we define a variation of the GARCH model which is aware of the arrival of news. In Eq. (4) we defined the set of each distinct news document for the stock. For the purpose of forecasting the reaction to news we are more concerned whether news occurred at the given time for the stock. Therefore we produce the news time series defined in Eq. (24) such that each trading minute for the stock contains the count of the documents forecast to cause a shock. Note that $\Gamma(A_{(s,\Delta\tau,\lambda)},\delta)$ denotes the outcome of the classifiers defined in 3.1 where δ is the given threshold. Note also that when we refer to δ=0, we simply mean that no classification was used so every document at the given time is included.

$$N_{(s,\Delta\tau,\delta)} = \{N_1, N_2, ..., N_q\} \mid N_{(s,\Delta\tau,\delta,t)} = \\ \#\{\forall A_{(s,\Delta\tau,\lambda)} \mid D_{(t-1)} < d_{(s,\lambda)} \le D_{(t)} \wedge \Gamma(A_{(s,\Delta\tau,\lambda)},\delta) = 1\} \tag{24}$$

3.2.1. NAGARCH-S

Let us assume that the GARCH model is effective at forecasting future volatility when news has not been released to the market. Furthermore let us assume that when news is released to the market investors process this information and their behaviour makes it difficult for GARCH to forecast volatility. Therefore the state of the GARCH model must change in order to take advantage of the knowledge that news has been released.

The Baseline GARCH model for predicting Δt minutes into the future for the stock *s*, at time *t* is given by $(\sigma_{B(s,\Delta t,P,Q,t)}{}^2)$ in Eq. (25) where α_{B0}, α_{Bi}, and β_{Bj} are constants.

$$\sigma_{B(s,\Delta t,P,Q,t)} = \sqrt{\alpha_{B0} + \sum_{i=1}^{P} \alpha_{Bi} R_{(s,\Delta t,t-i\times\Delta t)}{}^2 + \sum_{j=1}^{Q} \beta_{Bj} \sigma_{B(s,\Delta t,P,Q,t-j\times\Delta t)}{}^2} \tag{25}$$

We define the News Aware GARCH Switching model (NAGARCH-S) for predicting Δt minutes into the future for the stock *s*, at time *t* is given by $(\sigma_{S(s,\Delta t,P,Q,\delta,t)}{}^2)$ in Eq. (25), where α_{S0}, α_{Si}, and β_{Sj} are constants. Furthermore $N_{(s,\Delta t,\delta,t-k)}$ is the number of articles at time *t-k* classified using the threshold δ to correlate with abnormal market behaviour. Note that the conditional variance (i.e., the forecast volatility) of the Baseline GARCH model is used within NAGARCH-S. This is to ensure that forecasts are unaffected by a period when news occurs frequently, which is a concern for parameter optimisation.

$$\sigma_{S(s,\Delta t,P,Q,\delta,t)} = \left(\left(\sum_{k=1}^{\Delta t} N_{(s,\Delta t,\delta,t-k)}\right) = 0 \ ?\ \sigma_{B(s,\Delta t,P,Q,t)} : \right. \\ \left. \sqrt{\alpha_{S0} + \sum_{i=1}^{P} \alpha_{Si} R_{(s,n,\Delta t,t-i\times\Delta t)}{}^2 + \sum_{j=1}^{Q} \beta_{Sj} \sigma_{B(s,\Delta t,P,Q,t-j\times\Delta t)}{}^2}\right) \tag{26}$$

Parameters which are evaluated in a given test month are optimised to maximise the Log Likelihood function defined in Eq. (9) for the models during a training set for each stock. The training set comprises of a limited number of months which occurred prior to the test month. The classifier used to classify documents during the training period is trained during a period which excludes both the training and test months. This is to ensure that the classifier does not use prior knowledge to determine how the market will react after the news is released.

Parameters for the Baseline GARCH model are optimised over the entire time series in the training set for the stock. Parameters for the NAGARCH-S model are optimised during the Δt minutes after the release of a document classified to correlate with abnormal market behaviour using the threshold δ in the training set. This is because it is the only time when the model produces a different forecast from the Baseline GARCH model.

In the event that parameters could not be found to improve the NAGARCH-S model over the Baseline GARCH model, parameters from a previous month for the stock are chosen.

3.3. Measuring Forecast Performance

In order to evaluate whether the NAGARCH-S model is any better than the GARCH model it is necessary to measure the difference in forecast accuracy. In this section we define several measures which we use to compare the models.

The benchmark signal (*b*), given by Eq. (27), is the error between the forecast and realised volatility for the stock *s* at time *t* using the GARCH model. We define the realised volatility at time *t* using the volatility definition in Eq. (7) using n=1 and ρ=2.

$$b_{(s,\Delta t,P,Q,t)} = \sigma_{B(s,\Delta t,P,Q,t)}{}^2 - \upsilon_{(s,1,2,\Delta t,t)}{}^2 \tag{27}$$

The forecast signal (*f*) is the error between the NAGARCH-S forecast and the realised volatility for the stock *s* at time *t*. We define the realised volatility at time *t* using the volatility definition in Eq. (7) using n=1 and ρ=2.

$$f_{(s,\Delta t,P,Q,\delta,t)} = \sigma_{S(s,\Delta t,P,Q,\delta,t)}{}^2 - \upsilon_{(s,1,2,\Delta t,t)}{}^2 \tag{28}$$

We want to evaluate the performance of the model across multiple stocks from the same country so we group the stocks as defined in Eq. (29). This is useful for calculating the average performance improvement for the model for each country.

$$S = \{S_1, S_2, ..., S_h\} | h \geq 1 \quad (29)$$

3.3.1. Unscaled Forecast Quality (Q_u)

The unscaled forecast quality (Q_u), given by Eq. (30), measures the improved performance of the model over the benchmark by comparing the sum of the absolute errors for all stocks in the set. Note that term "unscaled" is used as typically the forecast quality factors in the change in the realised volatility (Dacorogna et al. 2001). Note also that $t \in T_{B(s)}$ means that the minute t is a member of business time T_B for the stock s. In other words the minute occurred during a business day when the stock was not suspended from trading.

$$Q_{u(S,\Delta t,P,Q,\delta)} = 1 - \frac{\sum_{\forall s \in S} \sum_{\forall t} \left\{ \left| f_{(s,\Delta t,P,Q,\delta,t)} \right| \mid t \in T_{B(s)} \right\}}{\sum_{\forall s \in S} \sum_{\forall t} \left\{ \left| b_{(s,\Delta t,P,Q,t)} \right| \mid t \in T_{B(s)} \right\}} \quad (30)$$

3.3.2. Superior Quality (Q_s)

The superior quality (Q_s), given by Eq. (31), finds the percentage of times that the forecast signal is better than the benchmark signal. If the value is 0 then it is not worth using the model as the forecast is never better than the benchmark. Note that $t \in T_{B(s)}$ means that the minute t is a member of business time T_B for the stock s. In other words the minute occurred during a business day when the stock was not suspended from trading.

$$Q_{s(S,\Delta t,P,Q,\delta)} = \frac{\sum_{\forall s \in S} \# \left\{ \forall t \mid \left| f_{(s,\Delta t,P,Q,\delta,t)} \right| < \left| b_{(s,\Delta t,P,Q,t)} \right| \mid t \in T_{B(s)} \right\}}{\sum_{\forall s \in S} \# \left\{ \forall t \mid t \in T_{B(s)} \right\}} \quad (31)$$

4. Results

We have separated the results in sections titled News Classification and Model Performance. The first section describes how the best classifiers were chosen for each country. The second section describes how the NAGARCH-S model performed using the classified news.

4.1. News Classification

We have divided this section into two subsections. The first addresses the issue of the size of the time window for finding abnormal market behaviour. The second addresses the problem of how much historical knowledge is necessary to produce the best classifiers.

4.1.1. Choice of Time Window

In order to choose an effective news classifier it is first necessary to determine an effective time window for measuring abnormal behaviour. For this purpose the documents are categorised using various time window sizes (Δt=$\Delta \tau$) and δ=6 standard deviations for the forecast error time series with P=Q=3 (approximately the 99.7th percentile for 30 minute returns) (Robertson et al. 2007).

There were 10 training sets created by selecting N=1,000 documents and R=500 documents at random which correlated to abnormal market behaviour from the entire collection of news documents for the country. The test set for the respective training sets contained all the documents for the country which were not included in the training set. An equal allocation of documents which correlated to abnormal behaviour and those that did not was chosen so as not to bias the classifier. Tests were run using both the SVM and the C4.5 classifiers and each term ranking algorithm with varying φ values (100, 200, 500, 1,000, 2,000, and 5000 terms).

In Fig. 1 the effect of increasing the time window size (Δt=$\Delta \tau$) is investigated on the mean accuracy of the classifiers using every variation of φ terms (Note that the mean is calculated over the 10 test sets). The most accurate term ranking algorithm and classifier combination are displayed.

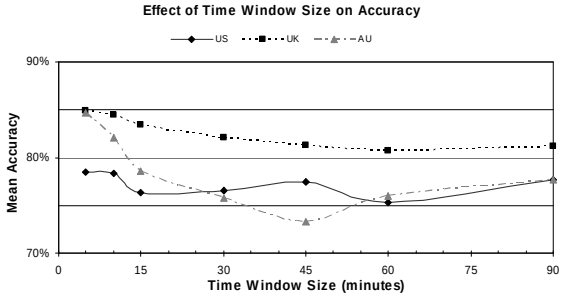


Fig. 1. Effect of Time Window Size on Accuracy.

The most accurate results (i.e., those with the highest mean accuracy) for every country are achieved within 5 minutes. As the time window size is increased there is a slight reduction in the accuracy in the UK, though a substantial reduction in Australia. The US is a little more stable than Australia but not as efficient as the UK. Therefore it appears that investors in all countries react quickly and decisively to news. This indicates that investors in all countries are rational. Increasing the time window size reduces the accuracy as $\Delta \tau$ increases the number of documents which spuriously correlate to abnormal market behaviour. Increasing the value of Δt however could yield better results as there is too much noise in the market at extremely high frequencies.

4.1.2. Choice of Historical Time Window

The tests in the previous section were useful for highlighting the time window size (Δt=$\Delta \tau$) to use for classifying news. However, it is not practical to use a classifier trained on a sample of all documents. This is because it is possible that priori information is used to classify the document. Therefore it is necessary to produce classifiers for each month which have no knowledge of the immediate future.

The training sets were created using the past Ω months. Documents are categorised using the forecast error time series with P=Q=3, Δt=$\Delta \tau$=5 minutes, and δ=6 standard deviations. Each document categorised as interesting during this period (R) is included in the training set. To

avoid biasing the classifier we use N=2R and therefore chose R uninteresting documents at random in the same period. In the event that there are not Ω months prior to the given month then extra months from the end of the dataset are used. It is unlikely that an event which caused a major shock will be referred to in a document released a long time afterwards.

A training set is created for each month and each Ω value (3, 6, 9 and 12 months) using both the SVM and the C4.5 classifiers and each term ranking algorithm with varying φ values (100, 200, 500, 1,000, 2,000, and 5000 terms).

The results in Fig. 2 show the mean accuracy of the best classifier for each Ω value (Note that the mean is calculated over the test set for each month). It is clear that more historical knowledge is advantageous as it provides the classifier with a wide selection of different types of documents which correlated to shocks. If the classifier were only trained on documents which were released during annual reporting season, it is likely that there would be a strong bias towards words such as "earnings", "profit", and "loss". These words are less likely to cause a shock throughout the rest of the year, unless the document reports an unexpected large profit or loss. Note that the mean and standard deviation of classifiers which are trained on only immediate history are very similar to those which also use months from the end of the dataset.

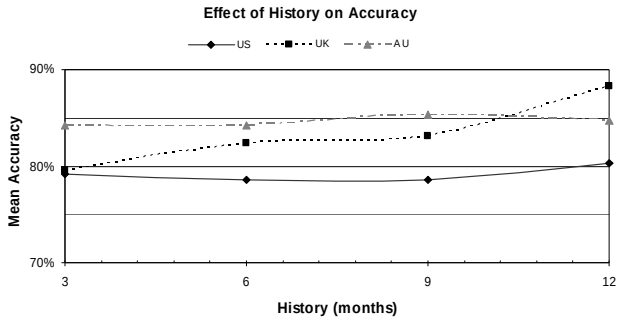


Fig. 2. Effect of History on Accuracy.

The results in Table 1 and Table 2 show the classification details for the best classifier for each country. The mean true and false positive rates are provided in the TPR and FPR columns respectively of Table 2. The Ω=12 value yielded the best results for the US and UK, whilst the Ω=9 value produced the best results for Australia.

Table 1. Characteristics of Best Classifiers.

					Documents	
δ	Country	Classifier	Term Ranking	Terms (φ)	Total	Positive
4	US	SVM	GAIN	1,000	133,019	28,742
	UK	C4.5	GAIN	100	81,522	6,907
	AU	SVM	ADBM25	5,000	33,098	5,187
6	US	SVM	ADBM25	100	133,019	25,944
	UK	C4.5	GAIN	100	81,522	8,995
	AU	SVM	ADBM25	2,000	33,098	4,753

The true positive rate (TPR) value in Table 2 shows that despite the UK having a high accuracy there is a low percentage of documents which actually correlated to a shock which were correctly classified. However, the accuracy rates for all tests are promising as Mittermayer (2004) only achieved 58%. It should be noted though the Mittermayer was attempting predict the direction of return, which is harder to do.

Table 2. Accuracy of Best Classifiers.

δ	Country	Accuracy	TPR	FPR
4	US	77.73%	34.36%	78.67%
	UK	90.19%	19.44%	91.77%
	AU	83.77%	39.13%	84.94%
6	US	80.31%	42.26%	80.77%
	UK	88.25%	25.60%	89.18%
	AU	85.19%	37.07%	86.04%

4.2. Model Performance

In this section we evaluate the performance of the NAGARCH-S model using several thresholds for the news time series. Initially we investigate the unscaled forecast quality for each country to determine if the NAGARCH-S model improves on the Baseline GARCH model. This includes tests to determine whether the results are statistically significant. Then we evaluate the superior quality for each country to determine how frequently the NAGARCH-S model provides a better forecast than the Baseline GARCH model.

For all tests we used the previous 3 months of trading data for each stock to optimise parameters. Furthermore the news time series were assembled using the classified documents for the same period for the stock. Specifically this means that 3 separate classifiers were used for each test as the classifiers were each produced to predict one month ahead. We did so because there are not enough samples for the δ=6 news time series with only one month of data.

We forecast the volatility of returns for every minute in the time series for every stock in each country, using P=Q=1 to limit the cost of parameter optimisation. We make no attempt to predict the delay between news arrival and market reaction, but simply use regression to optimise the parameters for the Δt minutes after news. Therefore if the market tends to take 3 minutes to react to news then the forecast volatility of the NAGARCH-S for the first 3 after news will probably be worse that the Baseline GARCH model. Note that as we classified documents with $\Delta\tau$=5 minutes, articles which occurred within the first or last 5 minutes of the trading day were excluded.

4.2.1. Unscaled Forecast Quality

In Fig. 3 - Fig. 5 the unscaled forecast quality (Q_u) for the US, UK, and Australia respectively is evaluated for all time windows (Δt). Note that Q_u is calculated for the Δt minutes after news as the models are the same without news. In each figure the legends STD0, STD4, and STD6 correspond to the model using news classifiers with the δ=0, δ=4, and δ=6 thresholds respectively. Note that the δ=0 threshold means that all news is processed by the NAGARCH-S model.

In Fig. 3 it is shown that the Q_u of the models using the δ=4 and δ=6 thresholds in the US are consistently better than for the δ=0 threshold. This suggests that the content of the news is important, and the classifiers are performing well. The δ=4 and δ=6 thresholds provide very similar values until after the 15 minute time window. This implies that news which is classified using the δ=6 is not significantly different from that classified using the δ=4 threshold. However, for time windows larger than 15 minutes the δ=4 threshold yields higher Q_u values. This is most likely because there are less documents are classified to correlate with abnormal volatility forecast errors using the δ=6 threshold. Therefore it is difficult to optimise parameters for this threshold as there are fewer periods around news, and therefore regression tends to overfit parameters.

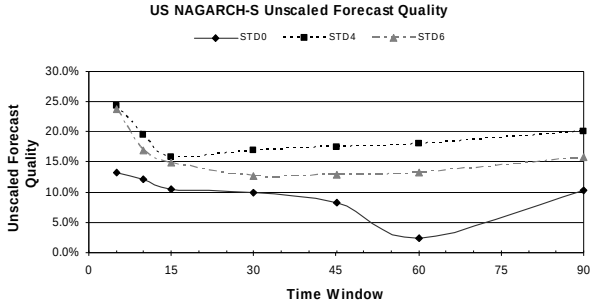


Fig. 3. Unscaled Forecast Quality in the US.

The results in Fig. 4 show that until the 15 minute window the Q_u of the model using the δ=4, and δ=6 thresholds in the UK are higher than for the δ=0 threshold. It is also clear that the δ=6 threshold yields better results than δ=4 during this period. However, for larger time windows the Q_u values tend to be negative. This is because the forecast accuracy of the Baseline GARCH model with these time windows is substantially lower than for smaller time windows. Therefore as NAGARCH-S attempts to improve on the Baseline GARCH model it overfits parameters to the training set which leads to significantly worse performance in the test set.

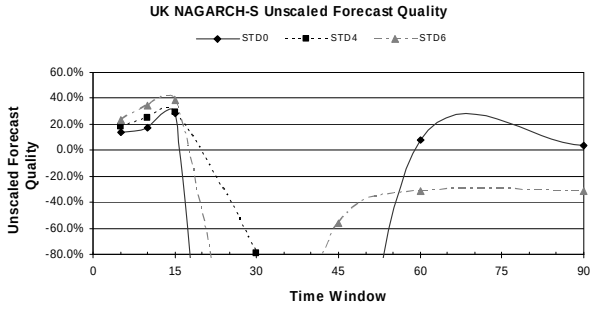


Fig. 4. Unscaled Forecast Quality in the UK.

For the 60 and 90 minute time windows in the UK, as shown in Fig. 4, the δ=0 threshold provides positive Q_u values. This indicates that the Baseline GARCH performance begins to improve and the large number of documents aids parameter optimisation. Therefore it appears that the content of news is important in the UK and the classifiers are performing well. However, it is difficult to forecast volatility a long time into the future. This suggests that the volatility does not persist for long after the release of news.

The 5 minute time window in Fig. 5 reveals that the δ=0 threshold provides higher Q_u values than the δ=4, and δ=6 thresholds in Australia. This is possibly because there is the potential for a large improvement over the Baseline GARCH model during this period, and the other thresholds do not have sufficient documents to optimise parameters effectively. However, for all other time windows the δ=4 and δ=6 thresholds yield higher Q_u values. This suggests that the content of news is important in Australia and that the classifiers are performing well.

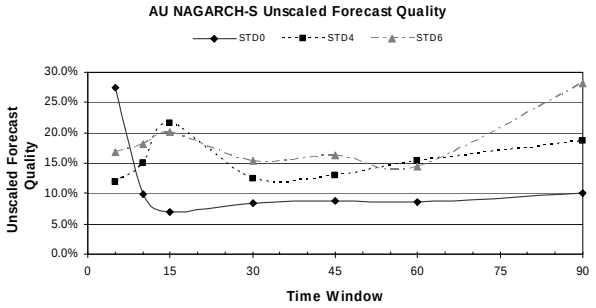


Fig. 5. Unscaled Forecast Quality in Australia.

We test the null hypothesis that the NAGARCH-S model produces the same forecasts as the Baseline GARCH model using an F-Test. This compares the average forecast error for each model for each month and each stock in the given country for the 5 minute time window. The p values of these tests are shown in Table 3. They reveal that, apart for the δ=0 threshold in the US, the NAGARCH-S model produces statistically significant different forecasts than the Baseline GARCH model.

Table 3. Significance of Forecasts for the 5 minute window.

	Threshold (δ)		
Country	**0**	**4**	**6**
US	69.79%	0.00%	6.63%
UK	0.00%	0.00%	0.00%
AU	0.00%	0.42%	0.00%

The results in this section indicate that documents classified to correlate with abnormal volatility forecast errors improve the NAGARCH-S model more than all documents. This implies that the content of the news is important and investors do not tend to react to all news.

4.2.2. Superior Quality

In Fig. 6 - Fig. 8 the superior quality (Q_s) for the US, UK, and Australia respectively is evaluated for all time windows (Δt). Note that Q_s is calculated for the Δt minutes after news as the models are the same without news. In each figure the legends STD0, STD4, and STD6 correspond to the model using news classifiers with the δ=0, δ=4, and δ=6 thresholds respectively. Note that the δ=0 threshold means that all news is processed by the NAGARCH-S model.

The results in Fig. 6 - Fig. 8 reveal that the δ=4 and δ=6 thresholds provide better forecasts than the δ=0 threshold for all time windows. Note that it is difficult to tell for the 5 minute time window in Australia, though it is the case.

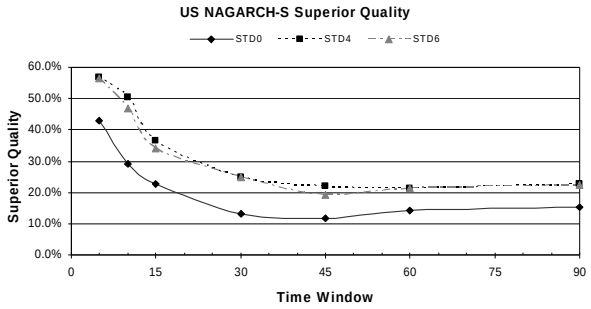


Fig. 6. Superior Quality in the US.

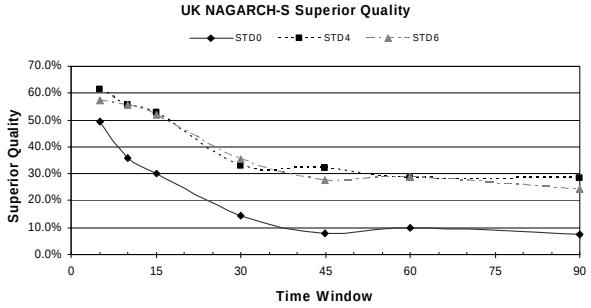


Fig. 7. Superior Quality in the UK.

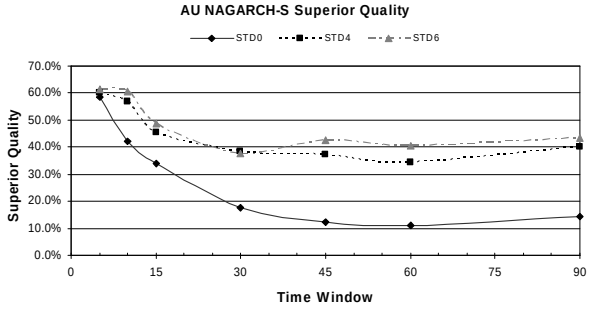


Fig. 8. Superior Quality in Australia.

The results in Fig. 6 - Fig. 8 also demonstrate that as the models attempt to forecast volatility further into the future there is less chance of producing better forecasts than the Baseline GARCH model. However, this does not mean that the models tend to be worse than the Baseline GARCH model. They actually have forecast accuracies greater than or equal to the Baseline GARCH model over 70% of the time for all time windows. Despite forecasts being worse for up to 30% of the time the results in the previous section reveal that the models tend to be better than the Baseline GARCH model. This suggests that when the models provide worse forecasts they are not large compared to the periods of better forecasts.

4.2.3. Summary

These results demonstrate that substantially greater forecasts can be achieved when considering news. However, the unscaled forecast quality results in the UK demonstrate that this model is not universally effective. Therefore it is necessary to perform comprehensive tests on a large dataset before determining what conditions are best for applying this model.

5. Conclusions

We have introduced a variation of the GARCH model which is aware of the arrival of news. We have shown that it is very effective at improving the forecast accuracy around news for the US and Australia for forecasts up to 90 minutes into the future. However, in the UK it is best not to forecast more than 15 minutes into the future as the model tends to be worse than the Baseline GARCH model.

We have demonstrated that classifying news based on the content improves the performance of this model more than by using all news. To our knowledge we have achieved higher classification accuracy rates for forecasting the market reaction to news than any previously reported by other authors.

Furthermore we have provided evidence that these models are statistically better than GARCH except in the US when using all news. Therefore it is clear that knowledge of news arrival is not enough, and it is very important to interpret the content of the news before forecasting how the market will react.

In future research we plan to investigate ways to improve the forecasts.

6. References

Almeida, A., Goodhart, C. A. E. and Payne, R. (1998): The Effects of Macroeconomic News on High Frequency Exchange Rate Behavior. Journal of Financial & Quantitative Analysis, 33(3):383-408.

Bollerslev, T. (1986): Generalized Autoregressive Conditional Heteroskedasticity. Journal of Econometrics, 31(3):307-27.

Cutler, D. M., Poterba, J. M. and Summers, L. H. (1989): What Moves Stock Prices? Journal of Portfolio Management, 15(3):4-12.

Dacorogna, M. M., Gencay, R., Müller, U., Olsen, R. B. and Pictet, O. V. (2001) An Introduction to High-Frequency Finance, Academic Press, London.

Ederington, L. H. and Lee, J. H. (1993): How markets process information: News releases and volatility. Journal of Finance, 48(4):1161-1191.

Ederington, L. H. and Lee, J. H. (1995): The short-run dynamics of the price adjustment to new information. Journal of Financial & Quantitative Analysis, 30(1):117-134.

Ederington, L. H. and Lee, J. H. (2001): Intraday Volatility in Interest-Rate and Foreign-Exchange Markets: ARCH, Announcement, and Seasonality Effects. Journal of Futures Markets, 21(6):517-552.

Goodhart, C. A. E. (1989): News and the foreign exchange market. Proc. Manchester Statistical Society, 1-79.

Goodhart, C. A. E., Hall, S. G., Henry, S. G. B. and Pesaran, B. (1993): News Effects in a High-Frequency Model of the Sterling-Dollar Exchange Rate. Journal of Applied Econometrics, 8:1-13.

Graham, M., Nikkinen, J. and Sahlstrom, P. (2003): Relative Importance of Scheduled Macroeconomic News for Stock Market Investors. Journal of Economics and Finance, 27(2):153-165.

Hong, H., Lim, T. and Stein, J. C. (2000): Bad news travels slowly: Size, analyst coverage, and the profitability of momentum strategies. Journal of Finance, 55(1):265-95.

Joachims, T., SVM Light Classifier(2007). Available: http://svmlight.joachims.org/.

Kalev, P. S., Liu, W.-M., Pham, P. K. and Jarnecic, E. (2004): Public Information Arrival and Volatility of Intraday Stock Returns. Journal of Banking and Finance, 28(6):1441-1467.

Kim, S.-J., McKenzie, M. D. and Faff, R. W. (2004): Macroeconomic News Announcements and the Role of Expectations: Evidence for US Bond, Stock and Foreign Exchange Markets. Journal of Multinational Financial Management, 14(3):217-232.

Melvin, M. and Yin, X. (2000): Public Information Arrival, Exchange Rate Volatility, and Quote Frequency. Economic Journal, 110(465):644-661.

Michaely, R. and Womack, K. L. (1999): Conflict of Interest and the Credibility of Underwriter Analyst Recommendations. Review of Financial Studies, 12(4):653-86.

Mitchell, M. L. and Mulherin, J. H. (1994): The Impact of Public Information on the Stock Market. Journal of Finance, 49(3):923-50.

Mittermayer, M.-A. (2004): Forecasting Intraday Stock Price Trends with Text Mining Techniques. Proc. 37th Annual Hawaii International Conference on System Sciences (HICSS'04), Big Island, Hawaii, 30064b.

Nofsinger, J. R. and Prucyk, B. (2003): Option volume and volatility response to scheduled economic news releases. Journal of Futures Markets, 23(4):315-345.

Porter, M. F. (1980): An Algorithm for Suffix Striping. Automated Library and Information Systems, 14(3):130-137.

Quinlan, J. R. (1993) C4.5: Programs for Machine Learning, Morgan Kaufmann.

Robertson, C. S., Geva, S. and Wolff, R. C. (2007): The Intraday Effect of Public Information: Empirical Evidence of Market Reaction to Asset Specific News from the US, UK, and Australia. SSRN Working Paper Series: http://ssrn.com/abstract=970884.

Robertson, S. and Spärck Jones, K. (2006): Simple, Proven Approaches to Text Retrieval. University of Cambridge Computer Laboratory Technical Report no. 356.

Roll, R. (1984): Orange Juice and Weather. American Economic Review, 74(5):861-80.

Vapnik, V. (1999) The Nature of Statistical Learning Theory, Springer-Verlag.

Womack, K. L. (1996): Do Brokerage Analysts' Recommendations Have Investment Value? Journal of Finance, 51(3):137-67.

Effectiveness of Using Quantified Intermarket Influence for Predicting Trading Signals of Stock Markets

Chandima D. Tilakaratne[1,2] **Musa A. Mammadov**[1] **Sidney A. Morris**[1]

[1] Center for Informatics and Applied Optimization
School of Information Technology and Mathematical Sciences
University of Ballarat, PO Box 663
Ballarat, Victoria 3353, Australia,
Email: {ctilakaratne@students., m.mammadov@, s.morris@}ballarat.edu.au

[2] Department of Statistics
University of Colombo,
PO Box 1490, Colombo 3, Sri Lanka.

Abstract

This paper investigates the use of influence from foreign stock markets (intermarket influence) to predict the trading signals, buy, hold and sell, of the of a given stock market. Australian All Ordinary Index was selected as the stock market whose trading signals to be predicted. Influence is taken into account as a set of input variables for prediction. Two types of input variables were considered: quantified (weighted) input variables and their un-quantified counterparts. Two criteria was applied to determine the trading signals: one is based on the relative returns while the other uses the conditional probability that a given relative return is greater than or equals zero. The prediction of trading signals was done by Feedforward neural networks, Probabilistic neural networks and so called probabilistic approach which was proposed in past studies. Results suggested that using quantified intermarket influence as input variables to predict trading signals, is more effective than using their un-quantified counterparts.

Keywords: Forecasting, Stock market, Intermarket Influence, Neural networks, Optimization

1 Introduction

Profitability of stock market trading is directly related to the prediction of trading signals. The majority of the past studies (Chenoweth et al., 1996; Fernando et al., 2000; Vanstone, 2006; Wood & Dasgupta, 1996; Yao et al., 1999) focused on classification of future values into two categories (up or down) which are considered to be buy and sell signals. Timely decisions must be made which result in buy signals when the market is low and sell signals when the market is high (Chapman, 1994). However, it is worth holding shares if there is no significant rise or drop in the price index. Therefore, from the practical point of view, it is important to consider the 'hold' category.

The literature (Bhattacharyya & Banerjee, 2004; Eun & Shim, 1989; Taylor & Tonks, 1989; Wu & Su, 1998; Yang et al., 2003) confirms that the world's major stock markets are integrated. Also some studies (Becker et al., 1990; Eun & Shim, 1989; Wu & Su, 1998) provide evidence that US stock markets have strong influence on the other major global markets. These studies confirm the existence of intermarket influence[1] among the global stock markets. Hence, one stock market can be considered as a part of a single global system (Tilakaratne et al., 2006). The influence from one stock market on a dependent market may include the influence from one or more stock markets on the former. This matter indicates that the intermarket influence (from a set on influential markets on a dependent market) needs to be quantified in order to use them effectively in applications such as prediction.

Although, some evidence found in the literature (Olson & Mossaman, 2001; Pan et al., 2005) for the possibility of improving the prediction accuracy by incorporating intermarket influence, none of these studies either aimed at predicting trading signals or incorporated quantified intermarket influence for predictions.

The aim of this paper is to investigate the effectiveness of applying quantified intermarket influence for predicting trading signals, buy, hold and sell, of stock markets. We chose the Australian All Ordinary Index (AORD) as the stock market to be studied. Following Yao & Tan (2000) this study also assumed the major blue chips in the stock basket are bought or sold, and the aggregate price of the major blue chips is the same as the index.

The remainder of this paper is presented as follows: The next section discusses the related work. The third section explains the technique used for quantifying intermarket influence on the AORD together with the corresponding optimization problem. This section also presents the quantified intermarket influence on the AORD. The forth section defines the trading signals. The fifth section discusses the techniques (algorithms) applied for predicting trading signals and how these algorithms were trained. The input features used for these algorithms are also discussed in this section. The sixth section described how the prediction results were evaluated. The next section presents the results together with interpretations. Final section concludes the paper.

2 Related Work

In the last few decades, there has been a growing number of studies attempting to predict the trading

[1] Intermarket Influence Analysis is defined as the study of relationships between the current price (or a derivative of price) of a dependent market with lagged price (or a derivative thereof) of one or more influential markets (Tilakaratne, 2006; Tilakaratne et al., 2006)

signals of financial market indices. Many past studies (for example, Chenoweth et al. (1996); Fernando et al. (2000); Vanstone (2006); Wood & Dasgupta (1996); Yao et al. (1999)) considered only two trading signals: buy and sell. Although not very common, some studies (for example, Chen et al. (2003); Chenoweth et al. (1996); Kohara et al. (1997); Kuo (1998); Leung et al. (2000); Mizuno et al. (1998)) considered a third signal:hold.

Feedforward neural networks (FNNs) and Probabilistic neural networks (PNNs) seem to be the most commonly used techniques in the literature (Chen et al., 2003; Fernando et al., 2000; Kohara et al., 1997; Leung et al., 2000; Mizuno et al., 1998; Wood & Dasgupta, 1996; Yao et al., 1999) to forecast the trading signals. Some studies (Chen et al., 2003; Leung et al., 2000) showed that the PNNs outperform alternative linear as well as non-linear models in terms of profitability. The literature (Kohara et al., 1997; Yao et al., 1999) reveals that the FNNs outperform the alternative linear models. Furthermore, Leung et al. (2000) found that the PNNs outperformed the FNNs in terms of profitability and predictability.

A PNN directly outputs the trading signals while a FNN outputs the value of the stock market index of interest or its derivative such as relative return. The predicted value is classified as a trading signal according to a certain criterion.

Different studies used different criteria for defining trading signals. Fernando et al. (2000); Kohara et al. (1997); Leung et al. (2000); Mizuno et al. (1998); Vanstone (2006) and Yao et al. (1999) determined the trading signals based on the value of index level or relative return. Chen et al. (2003) and Leung et al. (2000) used a criterion based on the probability to define the trading signals. Studies (such as, Fernando et al. (2000); Vanstone (2006); Yao et al. (1999)), which concern only two signals (buy and sell), considered only one threshold. On the other hand, the studies (for instance Kohara et al. (1997); Kuo (1998); Mizuno et al. (1998)), which considered three trading signals used two threshold criteria. Unlike others, Chen et al. (2003) applied a single threshold criterion as well as a two threshold criterion to determine the trading signals.

Vanstone (2006) suggested that the fundamental variables may be suitable as the input features, if the intention is to do long term forecasts. On the other hand, if the intention is to do short term predictions, technical variables may be more suitable.

Some studies, for example, (Chen et al., 2003; Kohara et al., 1997; Kuo, 1998; Leung et al., 2000; Vanstone, 2006), relied on both fundamental and technical variables for forecasting. Many published research (for instance, Chen et al. (2003); Chenoweth et al. (1996); Fernando et al. (2000); Kohara et al. (1997); Kuo (1998); Leung et al. (2000); Mizuno et al. (1998)) used technical indicators to predict trading signals. Some of these studies (Chenoweth et al., 1996; Fernando et al., 2000; Mizuno et al., 1998) relied only on technical indicators. The use of lagged price or derivatives of the price of the stock market whose trading signals to be predicted seems to be a common feature in many fast studies (Chen et al., 2003; Chenoweth et al., 1996; Fernando et al., 2000; Kohara et al., 1997; Leung et al., 2000; Mizuno et al., 1998).

Nowadays, experts argue that stock markets are influenced by many interrelated factors including the effects of economic, political and even psychological factors. These factors interact with each other in a complex fashion, and it is therefore, very difficult to find an exact set of factors which determine the behaviour of stock markets (Tilakaratne, 2004). The effect of these factors may reflect on the price index on global markets. Hence, it may be able to capture the combine effect of these factors by using lagged price index to predict future price index of the same markets as well as the other global market.

It is noteworthy that the use of lagged prices or the derivatives of the prices of foreign stock markets to predict the trading signals of a selected market is very rare. The use of such information to predict trading signals may improve the predictability and profitability of the prediction.

3 Quantification of Intermarket Influences

This study selected the AORD as the stock market index whose trading signals to be predicted. In order to investigate the effectiveness of applying quantified intermarket influence for predicting trading signals of the AORD, the intermarket influence on the AORD needs to be quantified. This study adopts the quantification technique developed by Tilakaratne et al. (2007).

This technique quantifies the intermarket influences on a dependent market by finding the coefficients, ξ_i, i=1, 2, ...(see Section 3.1), which maximise the median rank correlation between the relative return of the Close price of day t of the dependent market and the sum of ξ_i multiplied by the lagged relative returns of the Close prices of a combination of influential markets over a number of small non-overlapping windows of a fixed size. ξ_i measures the contribution from the ith influential market to the combined influence which equals to the optimal correlation.

There is a possibility that the maximum value leads to a conclusion about a relationship which does not exist in reality. In contrast, the median is more conservative in this respect. Therefore, instead of selecting the maximum of the optimal rank correlation, the median was considered.

Spearmans Rank Correlation coefficient was used as the rank correlation measure. For two variables X and Y, Spearmans Rank Correlation Coefficient, r_s, can be defined as:

$$r_s = \frac{n(n^2-1) - 6\sum d_i^2 - (T_x + T_y)/2}{\sqrt{(n(n^2-1) - T_x)(n(n^2-1) - T_y)}} \quad (1)$$

where n is the total number of bivariate observations of x and y, d_i is the difference between rank of x and rank of y in the ith observation, T_x and T_y are the number of tied observations of X and Y, respectively.

Since, influential patterns between markets may vary with time (Tilakaratne, 2006), the whole study period was divided into a number of moving windows of a fixed length. The correlation structure between stock markets also changes with time (Wu & Su, 1998). Therefore, each moving window was further divided into a number of small windows of length 22 days. 22 days of a stock market time series represent a trading month. Spearman's rank correlation coefficients (see (1)) were calculated for these smaller windows within each moving window.

The absolute value of the correlation coefficient was considered when finding the median optimal correlation. This is appropriate as the main concern is the strength rather than the direction of the correlation (that is either positively or negatively correlated).

The objective function to be maximised (Section 3.1 described below) is defined by Spearmans correlation coefficient. Spearman's correlation coefficient is a piece-wise constant function as it depends on the rank of the elements of the vectors used for the calculation. Solving this type of optimization

problems is extremely difficult. The majority of algorithms need smoothness or at least semi-smoothness of the objective functions to be minimised. Only a few algorithms, that can be used to solve optimization problems with discontinuous objective functions, are available .

In this study, the global optimization algorithm developed by Mammadov (2004) and Mammadov et al. (2005) was used. This algorithm uses a line search mechanism where the descent direction is obtained via a dynamical system approach. The performance of this algorithm has been demonstrated in solving different optimization problems with discontinuous objective functions (for example Koubor et al. (2006)).

3.1 Optimization Problem

Let $Y(t)$ be the relative return of the Close price of a selected dependent market at time t and $X_j(t)$ be the relative return of the Close price of the jth influential market at time t. Define $X_\xi(t-i)$ as:

$$X_\xi(t-i) = \sum_j \xi_j X_j(t-i) \tag{2}$$

where the coefficient $\xi_j \geq 0$, $j = 1, 2, ..., m$, measures the strength of influence from each influential market X_j. We named these coefficients *quantification coefficients*. m is the total number of influential markets and i represents the time lag.

The aim is to find the optimal values of the quantification coefficients, $\xi = (\xi_1, ..., \xi_m)$, which maximise the rank correlation $Y(t)$ and $X_\xi(t-i)$ for a given window and time lag i. In the calculations, $i = 0, 1, 2, 3, 4$, which represent influence within a week, were considered. $i = 0$ gives the same day correlation between the Close price of the dependent market and a selected combination of the Closes prices of influential markets. $i = 1$ gives the correlation between the Close price of day t of the dependent market and the Close prices of day $(t-1)$ of a combination of influential markets and this correlation is referred as the previous day's combined influence from the influential markets on the dependent markets. Other time lags can be defined in a similar manner.

The correlation can be calculated for a window of a given size. This window can be defined as;

$$T(t^0, l) = \{t^0, t^0 + 1, ..., t^0 + (l-1)\} \tag{3}$$

where t^0 is the starting date of the window and l is its size (in days).

The correlation between the variables $Y(t), X_\xi(t-i),\ t \in T(t^0, l)$, defined on the window $T(t^0, l)$, will be denoted as;

$$Corr(Y(t), X_\xi(t-i) \parallel T(t^0, l)) \tag{4}$$

For a period of several years, the optimal correlation changes according to the starting point of the window. To define optimal weights for a long period, the following method is applied. Let $[1, T] = 1, 2, ..., T$ be a given period (for instance a large window). This period is divided into n windows of size l. This study set l=22 days.

$$T(t_k, l), \quad k = 1, 2, 3, ..., n \tag{5}$$

so that,

$$T(t_k, l) \cap T(t_{k'}, l) = \phi \quad \text{for} \quad \forall \ \ k \neq k' \tag{6}$$

$$\bigcup_{k=1}^{n} T(t_k, l) = [1, T] \tag{7}$$

For given i, the correlation coefficient on a window $T(t_k, l)$ is denoted as;

$$C_k^i(\xi) = Corr(Y(t), X_\xi(t-i) \parallel T(t_k, l)), \ \ k = 1, ..., n. \tag{8}$$

To define an objective function over the period $[1, T]$, the median of the vector, $(C_1^i(\xi), ..., C_n^i(\xi))$ is used. Now, the main optimization problem can be defined as:

$$\text{Maximise} \quad f(\xi) = \text{Median} \ \ (C_1(\xi), ..., C_n(\xi)); \tag{9}$$

$$\text{s.t.} \quad \sum_j \xi_j = 1, \quad \xi_j \geq 0 \quad j = 1, 2, ..., m. \tag{10}$$

The solution to (9) (10) is a vector, $\xi = (\xi_1, ..., \xi_m)$, where ξ_j , $j = 1, 2, ..., m$, denotes the strength of the influence from the jth influential market.

In this paper, the quantity, $\xi_j X_j$ is called the quantified relative return corresponding the jth influential market.

3.2 Quantification of Intermarket Influence on the AORD

Tilakaratne et al. (2006) revealed that the Close prices of the US S&P 500 Index (GSPC), the UK FTSE 100 Index (FTSE), French CAC 40 Index (FCHI), German DAX Index (GDAXI) as well as that of the AORD itself showed an impact on the direction of the next day's Close price of the AORD. Therefore, this study quantified the intermarket influence from the following two combinations of stock market indices: (i) the GSPC, FTSE, FCHI and the GDAXI; and, (ii) the GSPC, FTSE, FCHI, GDAXI and the AORD. Also Tilakaratne et al. (2007) found that only the Close prices of day $(t-1)$ of these market significantly [2] influence the Close price of day t of the AORD. Hence it is sufficient to consider i=1 in (2). In other words, the relative returns of the Close prices of day $(t-1)$ of the above mentioned market combinations were considered for the quantification.

For this study, we consider the time series data corresponding to the relative returns of Close prices of the above mentioned five markets, from 2nd July 1997 to 30th December 2005. Since different stock markets are closed on different holidays, the regular time series data sets considered have missing values. If no trading took place on a particular day, the rate of change of price should be zero. Therefore, the missing values of the Close price were replaced by the corresponding Close price of the last trading day.

Relative Returns RR of the daily Close price of the stock market indices were used for the analysis.

$$RR(t) = \frac{P(t) - P(t-1)}{P(t-1)} \tag{11}$$

where $RR(t)$ and $P(t)$ are the relative return and the Close price of a selected index on day t, respectively. Returns are preferred to price, since returns for different stocks are comparable on equal basis.

It is worth noting that the opening and closing times for many of the various markets do not coincide. For example, the Australian, Asian, French and German markets have all closed by the time the US markets open.

[2] Optimal median rank correlation is significant at the 5% level

The whole study period was divided into six moving windows of three trading years (for stock market time series, 256 days is considered as a trading year). Each time the window was shifted forward by one trading year in order to get the starting point of the next window. For each window, the quantification coefficients, which maximise the median Spearman's rank correlation between the relative return of the Close price of day t of the AORD and the sum of the quantification coefficient multiplied by the relative returns of the Close prices day $(t-1)$ of the potential influential markets, were derived.

Table 1 and 2 presents the quantification coefficient associated with each market for each window with the corresponding combined influence (optimal median Spearman's correlation) for market combinations (1) and (2), respectively.

Table 1: Optimal values of quantification coefficients (ξ) and the optimal median Spearman's correlations corresponding to market combination (1) for different moving windows

Mov. win. no.	Optimal values of ξ				Opt. median Spear. corr.
	GSPC	FTSE	FCHI	GDAXI	
1	0.57	0.29	0.12	0.02	0.578
2	0.61	0.18	0.08	0.13	0.548
3	0.77	0.09	0.13	0.01	0.568
4	0.79	0.06	0.15	0.00	0.579
5	0.56	0.17	0.03	0.24	0.590
6	0.66	0.06	0.08	0.20	0.5359

Table 2: Optimal values of quantification coefficients (ξ) and the optimal median Spearman's correlations corresponding to market combination (2) for different moving windows

Mov. win. no.	Optimal values of ξ					Opt. median Spear. corr.
	GSPC	FTSE	FCHI	GDAXI	AORD	
1	0.56	0.29	0.10	0.03	0.02	0.580
2	0.58	0.11	0.12	0.17	0.02	0.550
3	0.74	0.00	0.17	0.02	0.07	0.570
4	0.79	0.07	0.14	0.00	0.00	0.580
5	0.56	0.17	0.04	0.23	0.00	0.590
6	0.66	0.04	0.09	0.20	0.01	0.537

Optimal median correlations are significant at 5% level irrespective of the window number and the market combination (Table 1 to 2). The GSPC seems to be the most influential market on the AORD.

The quantification coefficients (ξ) presented in the above two tables (Table 1 to 2) were used when predicting the trading signals of the AORD (Section 5.1).

4 Defining Trading Signals

Most of the past studies (Fernando et al., 2000; Vanstone, 2006; Wood & Dasgupta, 1996; Yao et al., 1999) classified the future values into buy or sell signals based on the direction of the trend (upward or downward) of the future values. The studies (Chen et al., 2003; Chenoweth et al., 1996; Kohara et al., 1997; Kuo, 1998; Leung et al., 2000; Mizuno et al., 1998) aimed at predicting three trading signals (buy, hold and sell) applied two threshold criteria. Since, this study also consider three signals, the following criterion, which uses two thresholds, was introduced to determine the trading signals.

Criterion A

$$\begin{array}{ll} buy & \text{if} \quad Y(t) \geq l_u \\ hold & \text{if} \quad l_l < Y(t) < l_u \\ sell & \text{if} \quad Y(t) \leq l_l \end{array}$$

where $Y(t)$ is the relative return of the Close price of day t of the AORD while l_u and l_l are two thresholds.

The values of l_u and l_l depend on the traders' choice. There is no standard criterion found in the literature how to decide the values of l_u and l_l and theses values may vary from one stock index to another. A traders may decide the values for these threshold according to his/her knowledge and experience.

We tested a range of values for l_u and l_l. The selection of suitable pair of values was done on basis of the profitability. Detailed description is found in Section 5.2.

The other way to identify the trading signals is to consider the probability of the predicted return is in upward (or downward) trend (Chen et al., 2003; Leung et al., 2000). Chen et al. (2003) considered the corresponding trading signal is a buy signal if this probability is above 0.7 and a sell signal if its value is below 0.5. Otherwise, the corresponding trading signal was considered as a hold signal. However, these limits associate with the probability of the predicted return is in upward trend may vary according to the stock index. Following Leung et al. (2000), this study also employed a criterion based on probability (Criterion B) to identify the trading signals.

Criterion B

$$\begin{array}{ll} buy & \text{if} \quad P \geq p_2 \\ hold & \text{if} \quad p_1 < P < p_2 \\ sell & \text{if} \quad P \leq p_1 \end{array}$$

where P is the conditional probability that a given relative return of the Close price of day t of the AORD ≥ 0. The choice of p_1 and p_2 is based on the profitability of trading and described in Section 5.4.1.

5 Predicting Trading Signals of the AORD

Since stock market time series are non-linear systems, the linear classification techniques (such as, linear regression, vector autoregressive models, linear discriminant analysis and ARIMA models) are not suitable for our prediction purpose. The literature (Section 2) shows that the non-linear classification techniques such as FNN, PNN and probabilistic approached (Section 5.4) proposed by Leung et al. (2000) performed well in predicting the trading signals of stock market movements. Therefore, this study also adopted these three techniques (algorithms) to predict the trading signals of the AORD.

5.1 Data Set Generation for Prediction Experiments

Two types of inputs sets were used as input features to the prediction algorithms (FNN, PNN and probabilistic approach); one set consists of the quantified relative returns while the other set contains the unquantified relative returns. The aim was to examine the effectiveness of applying quantified intermarket influence for the prediction of interest.

As previously mentioned in Section 3.2, the Close price on day t of the AORD is affected by those prices

on day $(t-1)$ of the GSPC, FTSE, FCHI, GDAXI as well as the AORD itself, th two combinations (mentioned in Section 3.2) of stock markets were considered when forming the input sets. Therefore, the input sets used for algorithms are:

1. Four input features of the relative returns of the Close prices on day $(t-1)$ of the market combination (1)
   - (GSPC$(t-1)$, FTSE$(t-1)$, FCHI$(t-1)$, GDAXI$(t-1)$); denoted as GFFG;
2. Four input features of the quantified relative returns of the Close prices on day $(t-1)$ of the market combination (1)
   - (ξ_1GSPC$(t-1)$, ξ_2FTSE$(t-1)$, ξ_3FCHI$(t-1)$, ξ_4GDAXI$(t-1)$); denoted as GFFG-q;
3. Five input features of the relative returns of the Close prices of on day $(t-1)$ the market combination (2)
   - (GSPC$(t-1)$, FTSE$(t-1)$, FCHI$(t-1)$, GDAXI$(t-1)$, AORD$(t-1)$); denoted as GFFGA;
4. Five input features of the quantified relative returns of the Close prices on day $(t-1)$ of the market combination (2)
   - (ξ_1^AGSPC$(t-1)$, ξ_2^AFTSE$(t-1)$, ξ_3^AFCHI$(t-1)$, ξ_4^AGDAXI$(t-1)$, ξ_5AORD$(t-1)$); denoted as GFFGA-q;

(ξ_1, ξ_2, ξ_3, ξ_4) and ($\xi_1^A, \xi_2^A, \xi_3^A, \xi_4^A, \xi_5^A$) are the solutions to (9) (10) (in Section 3.1) corresponding to the two market combinations: the GSPC, FTSE, FCHI and the GDAXI, and the GSPC, FTSE, FCHI, GDAXI and the AORD, respectively. We note that it may be $\xi_i \neq \xi_i^A$, for i=1, 2, 3, 4.

As mentioned previously in Section 3, the influential patterns between markets may vary with time. Hence, to capture these varying patterns, the algorithms were trained for several moving windows. The same six moving windows employed for quantified intermarket influence on the AORD (Section 3.2), was used for training the algorithms. Hence, the respective optimal values of quantification coefficient presented in Table 1 to 2 can be used as the corresponding values of ξ_i, i=1, 2, 3, 4 and ξ_i^A, i=1, 2, 3, 4, 5, respectively.

Each moving window consists of 768 samples (relative returns of three trading years). The most recent 10% of data (76 samples) of each window was allocated for testing while the remaining 90% (692 samples) was allocated for training. The training set was further divided into two sets; the most recent 22.2% of data of each training set (20% of the full data set) was allocated for validation while the remaining 77.8% (70% of the full data set) was used for training.

5.2 Training PNNs

The six moving windows, which was mentioned in Section 5.1, were used for the experiments with PNN. The above mentioned four input sets (Section 5.1) were considered for network training. Networks output the class (buy, hold, or sell) of AORD according to Criterion A (Section 4). In Criterion A, different pairs of values of l_l and l_u were tested; l_l= 0.003, 0.0035, 0.004, 0.0045, $\cdots$, 0.007 and l_u= 0.003, 0.0035, 0.004, 0.0045, $\cdots$, 0.007. The aim was to find suitable pair of values for l_l and l_u, which yield higher profits. Section 5.2.1 below describes how these values were determined.

The lost incurred by misclassification, for each class was assumed to be equal. The joint distribution of the input variables was assumed to be Gaussian. The parameters of the distribution were estimated by using the training data. When there were multiple inputs, the average standard deviation of the individual input variables was considered as the standard deviation of the joint distribution.

5.2.1 Choosing the Values for l_u and l_l

The validation set was used to determine the appropriate values for l_u and l_l in Criterion A. By varying the values of l_u and l_l, the corresponding trading signals for the validation was obtained. Trading simulations (described in Section 6.1) were performed on the trading signals corresponding to each pair of values considered. The pairs of values of l_u and l_l which gives the highest rate of return in each window for each input set are shown in Table 3.

Table 3: ($-l_l$, l_u) which gives the maximum rate of return (relevant to validation sets) for the four input sets for different windows (Some windows have more than one pair of values for ($-l_l$, l_u) and NA shows the cases where no trading (either buy or sell) took place

Input set	Window number	($-l_l$, l_u)
GFFG	1	(-0.0030, 0.0030)
	2	(-0.0040, 0.0030)
	3	(-0.0030, 0.0030)
	4	(-0.0055, 0.0030)
	5	NA
	6	NA
GFFG-q	1	(-0.0035, 0.0030)
	2	(-0.0035, 0.0030)
	3	(-0.0030, 0.0030)
	4	(-0.0040, 0.0030)
		(-0.0045, 0.0030)
	5	NA
	6	NA
GFFGA	1	(-0.0030, 0.0030)
	2	(-0.0035, 0.0035)
	3	(-0.0040, 0.0035)
	4	(-0.0045, 0.0035)
	5	(-0.0030, 0.0030)
	6	(-0.0035, 0.0030)
		(-0.0040, 0.0030)
		(-0.0045, 0.0030)
		(-0.0050, 0.0030)
		(-0.0055, 0.0030)
		(-0.0060, 0.0030)
		(-0.0065, 0.0030)
		(-0.0070, 0.0030)
GFFGA-q	1	(-0.0050, 0.0035)
	2	(-0.0035, 0.0035)
	3	(-0.0030, 0.0035)
	4	(-0.0030, 0.0030)
	5	(-0.0030, 0.0030)
	6	(-0.0030, 0.0035)
		(-0.0035, 0.0035)
		(-0.0030, 0.0040)
		(-0.0035, 0.0040)

The value of l_l which gives the highest rate of return varies from 0.0030 to 0.0070 while that for l_u takes values from 0.0030 to 0.0040 (Table 3). Therefore, the middle values of the ranges, [0.0030, 0.0070] and [0.0030, 0.0040] (that is 0.0050 and 0.0035), were chose as the appropriate values for l_l and l_u, respectively.

To evaluate the prediction results, the trading simulation was performed on the trading signals obtained from the test results. Criterion A with l_l=0.0050 and l_u=0.0035 was applied to determined these trading signals.

5.3 Training FNNs

The same six moving windows, that considered for PNN experiments (Section 5.2), were used for experiments with FNN. Three-layered FNNs with one hidden layer were trained for each one of the six moving windows considered. In each window, FNN was trained for 500 times.

The same sets of inputs (which were used as inputs for PNN) were considered as the inputs to FNN. These networks output the relative return of the Close price of the AORD. The average value of the prediction (over 500) for each day was calculated and this average value subsequently classified into the three classes of interest according to Criterion A (Section 4).

For this study we chose the same values, which used for the PNN experiments (Section 5.2.1), as the corresponding limits of Criterion A. In other words, 0.0050 and 0.0035 were taken as l_l and l_u, respectively.

A tan-sigmoid function was used as the transfer function between the input layer and the hidden layer while the linear transformation function was employed between the hidden and the output layers. The slope of a sigmoid function approaches zero as the input gets large and therefore the gradient can have a very small magnitude. If the steepest descent algorithm is used, this causes small changes in the weights and biases, even though the weights and biases are far from their optimal values (Demuth et al., 2006). Resilient backpropagation training algorithm (Rprop) (Riedmiller & Braun, 1989) eliminates these harmful effects of the magnitudes of the partial derivatives. It uses the sign of the derivative to determine the direction of the weight update; the magnitude of the derivative has no effect on the weight update. Therefore, the networks were trained with the resilient backpropagation training algorithm.

Different number of neurons for the hidden layer and different values for learning rate as well as the momentum coefficient were tested. FNNs gave the best results when there were three neurons in the hidden layer and the learning rate and the momentum coefficient were 0.003 and 0.01, respectively.

5.4 Probability Based Approach for Forecasting Trading Signals

Let $Y(t)$ be the relative return of the Close price of day t of the AORD (the target variable). The data is classified into two classes using $Y(t)$ as below:

$$\text{Upward Trend if } \; Y(t) \geq 0 \tag{12}$$

$$\text{Downward Trend if } \; Y(t) < 0 \tag{13}$$

Suppose that C_i is the target class corresponding to the i-th observation of a selected set of input features. Also let:

$$C_i = 1 \;\text{ if }\; Y(t) \geq 0 \tag{14}$$

$$C_i = 0 \;\text{ if }\; Y(t) < 0 \tag{15}$$

Then the conditional probability (P) that a given observation X_i belongs to the upward trend class is;

$$\begin{aligned} P &= \Pr(C_i = 1 | X = X_i) \\ &= \frac{\Pr(X = X_i | C_i = 1) \times \Pr(C_i = 1)}{\sum \Pr(X = X_i | C_i = j) \times \Pr(C_i = j)} \end{aligned} \tag{16}$$

$\Pr(X = X_i|C_i = j)$, j=0,1, can be calculated assuming a Gaussian distribution.

$$\Pr(X = X_i | C_i = j) = \frac{1}{2\Pi^{I/2}\sigma^I} \sum_{i=1}^{n_j} \exp \frac{-(X_i - j)'(X_i - j)}{2\sigma^2 n_j} \tag{17}$$

where I is the number of input features included in the input set and n_j is the number of training observations the class in which $C_i = j$, j=0,1.

The probability corresponding to each class (upward trend or downward trend) can be calculated as below:

$$\Pr(C_i = j) = \frac{n_j}{N_T} \tag{18}$$

where N_T is the total number of observations in the training set.

5.4.1 Parameter Estimation

The same six moving windows, which were used for training FNNs, PNNs were used for these experiments. Unlike Leung et al. (2000), this study considers a validation set in addition to the training and the test sets. The above mentioned four input sets (Section 5.1) were considered as the input variables.

As described by Leung et al. (2000), the parameters of the Gaussian distribution was estimated by using the training data set. This study assumes that the average standard deviations of the input variables (of the training sample) as the value of σ of the Gaussian distribution (see (17)). $\Pr(C_i = j)$, j=0,1 (see 18), was also estimated by using the training data.

Using the estimated Gaussian distribution and $\Pr(C_i = j)$, the conditional probability that a given observation X_i in the validation set belongs to the upward trend class, was derived. This probability associated with to each observation in the validation set was found.

Applying Criterion B (described in Section 4) on these probabilities relevant to the validation set, the corresponding trading signals (for the validation set) were determined. Different values for p_1 and p_2 were considered; p_1=0.20, 0.25, 0.30, 0.35, $\cdots$, 0.50 and p_2=0.50, 0.55, 0.60, 0.65, $\cdots$, 0.80. Practically, the conditional probability that a given observation X_i on an upward trend, P is below 0.5, the corresponding signal can not be considered as a buy signal. In contrast, if this probability is above 0.5, then the corresponding trading signal will not be a sell signal. Therefore, the upper limit for p_1 as well as the lower limit for p_2 should be 0.5. Leung et al. (2000) fixed the lower limit of p_1 at 0.254 and the upper limit of p_2 at 0.746. Therefore, we also chose closer values for the lower limit of p_1 and the upper limit of p_2.

By varying the values of p_1 and p_2, we aimed to find a suitable pair of values (for p_1 and p_2) which gives higher profits. Trading simulations (described in Section 6.1) were performed on the trading signals obtained by substituting different values of (p_1, p_2) in Criterion B. Values of (p_1, p_2) which yields highest rate of return (profit) for each window for each input set are shown in Table 4. These rates of returns were obtained from the trading simulations performed on the trading signals obtained from the validation set of each window.

According to trading simulations performed on the validation sets, the value of p_1 which yield the highest rate of return from the trading simulations varied between 0.4 and 0.5 (Table 4). The corresponding range for p_2 was [0.50, 0.70]. Therefore, the median value of the ranges of were taken as the values of p_1 and p_2. In other words, it was assumed p_1=0.45 and p_2=0.60.

Table 4: (p_1, p_2) which gives the maximum rate of return (relevant to validation sets) for the four input sets for different windows (Some windows have more than one pair of p_1 and p_2)

Input set	Window number	(p_1, p_2)
GFFG	1	(0.50, 0.50)
	2	(0.50, 0.60)
	3	(0.50, 0.65)
	4	(0.40, 0.60)
	5	(0.50, 0.60)
	6	(0.50, 0.70)
GFFG-q	1	(0.50, 0.60)
	2	(0.40, 0.60), (0.45, 0.60)
	3	(0.45, 0.55), (0.50, 0.55)
	4	(0.45, 0.50), (0.50, 0.50)
	5	(0.50, 0.60)
	6	(0.40, 0.70)
GFFGA	1	(0.50, 0.50)
	2	(0.50, 0.65)
	3	(0.50, 0.60)
	4	(0.40, 0.60)
	5	(0.45, 0.60)
	6	(0.50, 0.70)
GFFGA-q	1	(0.50, 0.55)
	2	(0.50, 0.50)
	3	(0.50, 0.60)
	4	(0.40, 0.50)
	5	(0.50, 0.60)
	6	(0.40, 0.70)

These probability levels are different from the corresponding probability levels used in Chen et al. (2003) (Section 4).

The conditional probability that a given test observation X_i belongs to the upward trend class was found. Finally, Criterion B with p_1=0.45 and p_2=0.60 was applied to determine the trading signals of each test set.

6 Evaluation of Predictions

The prediction results were evaluated in terms of profitability. Profitability was measured by the rate of return obtained by performing trading simulations.

The rate of return is a measure that provides the net gain in assets as a percentage of the initial investment. Profit depends not only on the accuracy of the forecasts but also on the trading strategy.

Different past studies employed different trading strategies to asses the profitability of the forecasts (Thawornwong & Enke, 2004). This study adopted buy and sell strategy to form the trading simulation. As mentioned in Section 1, this study assumed the major blue chips in the stock basket of the Australian stock exchange are bought or sold, and the aggregate price of the major blue chips is the same as the AORD.

The speciality of the trading simulation proposed in this study is that it search for the proportion of money that a trader needs to invest and the proportion of shares that he/she needs to sell in order to maximise the profit. In this sense, the proposed simulation is very close to the reality.

6.1 Trading Simulations

This study assumes that at the beginning of each period, the trader has some amount of money as well as a number of shares. Furthermore, it is assumed that the value of money in hand and the value of shares in hand are equal. Two types of trading simulations were used: (1) response to the predicted trading signals which might be a buy, hold or a sell signal; (2) do not participate in trading, and hold the initial shares and the money in hand until the end of the period. The second simulation was used as a benchmark.

6.1.1 First Trading Simulation (The Proposed Trading Simulation)

Let the value of the initial money in hand be M^0 and the number of shares at the beginning of the period be S^0. $S^0 = M^0/P_0$, where P_0 is the Close price of the AORD on the day before the starting day of the trading period.

Also let M_t, S_t, P_t, VS_t be the money in hand, number of shares, Close price of the AORD, value of shares holding on the day t (t=1, 2, ..., T), respectively. This simulation assumes that always a fixed amount of money is used in trading regardless of the trading signal is buy or sell. Let this fixed amount be denoted as F^0 and be equal to M^0/L, $L > 0$. In the calculations L = 1, 2, ..., 10 is considered. When L = 1, F^0 equals to M^0, when L = 2, F^0 equals to 50% of M^0 and so on. Let Δ_t^b and Δ_t^s be the number of shares buy and the number of shares sell at day t, respectively.

Suppose the trading signal at the beginning of the day t is a buy signal. Then the trader spends F =min$\{F^0, M_{t-1}\}$ amount of money to buy a number of shares at a rate of the previous day's Close price.

$$M_t = M_{t-1} - F, \qquad F = \min\{F^0, M_{t-1}\} \tag{19}$$

$$\Delta_t^b = \frac{F}{P_{t-1}} \tag{20}$$

$$S_t = S_{t-1} + \Delta_t^b \tag{21}$$

$$VS_t = S_t \times P_t \tag{22}$$

Suppose the trading signal is a hold signal, then:

$$M_t = M_{t-1} \tag{23}$$

$$S_t = S_{t-1} \tag{24}$$

$$VS_t = S_t \times P_t \tag{25}$$

Let the trading signal at the beginning of the day t is a sell signal. Then the trader sells S'=min$\{(F^0/P_{t-1}), S_{t-1}\}$ amount of shares.

$$\Delta_t^s = S', \qquad S' = \min\{(F^0/P_{t-1}), S_{t-1}\} \tag{26}$$

$$M_t = M_{t-1} + S' \times P_{t-1} \tag{27}$$

$$S_t = S_{t-1} - \Delta_t^s \tag{28}$$

$$VS_t = S_t \times P_t \tag{29}$$

It should be noted that a buy signal that immediately follows another buy signal will be treated as a hold signal. Also, if all shares have been sold, a sell signal is ignored.

6.1.2 Second Trading Simulation (The Benchmark Trading Simulation)

In this case the trader does not participate in trading. Therefore, $M_t = M^0$ and $S_t = S^0$ for all t=1, 2, ..., T. However, the value of the shares changes with the time and therefore, the value of shares at day t, $VS_t = S^0 \times P_t$.

6.2 Rate of Return

At the end of the period (day T) the total value of money and shares in hand:

- for the first trading simulation

$$TC = M_T + S_T \times P_T \tag{30}$$

- for the second trading simulation

$$TC = M^0 + S^0 \times P_T \tag{31}$$

The rate of return ($R\%$) at the end of a trading period is calculated as below:

$$R\% = \frac{TC - 2M_0}{2M_0} \times 100 \tag{32}$$

7 Results and Interpretations

This sections presents the rates of returns corresponding to FNN, PNN and the probabilistic approach, with the interpretations.

The trading simulations showed that the highest rate of return was obtained when the full amount of money in hand is invested and the full amount of shares in hand is sold. This matter was true for all the input sets as well as all the windows used.

Table 5 shows the average rates of return obtained by performing the proposed trading simulation (described in Section 6.1) on the prediction results (corresponding to the test set) obtained by FNN, PNN and the probabilistic approach, for different input sets.

Table 5: Average (over the six windows) rates of return relating to three algorithms trained with the four input sets (**The *annual average rate of return* relating to the benchmark simulation = 9.57%**)

Algorithm	Input set	Rate of return for test period	Annual rate of return
PNN	GFFG	3.65%	12.31%
	GFFG-q	3.99%	13.44%
	GFFGA	6.43%	21.66%
	GFFGA-q	6.91%	23.29%
FNN	GFFG	8.23%	27.72%
	GFFG-q	7.61%	25.63%
	GFFGA	8.21%	27.65%
	GFFGA-q	8.50%	28.63%
Prob. approach	GFFG	5.92%	19.94%
	GFFG-q	9.69%	32.64%
	GFFGA	8.91%	30.01%
	GFFGA-q	9.25%	31.16%

Table 5 evidences that, irrespective of the input set used, a trader can gain higher profits by responding to the trading signals produced by any algorithm considered. The average rate of return, obtained from the probabilistic approach, is higher when the predictions are based on the quantified intermarket influence (that is, input sets GFFG-q and GFFGA-q) than when the predictions are based on un-quantified intermarket influence. The highest rate of return was obtained when the predictions are based quantified intermarket influence from the GSPC, the three European markets and the AORD (input set GFFG-q).

The rates of return relevant to PNN also suggests that quantified intermarket influence produced more profitable trading signals, than their un-quantified counterparts. The highest rate of return was obtained when the quantified intermarket influence from the GSPC, the three European markets as well as the AORD itself was used as the input variables to predict the trading signals.

Results relating to FNN indicates that higher profits can be obtained when the quantified intermarket influence from the GSPC, the three European markets and the AORD were used as the input variables than using their un-quantified counterparts. However, the results relevant to the market combination of the GSPC and the three European markets suggests the opposite.

This exceptional behaviour of FNN may be due the inappropriateness of the Ordinary least squares (OLS) error function (used in the standard FNNs) for a classification problem. FNNs output the value of the prediction, but not the predicted class. OLS error function minimises the difference between the actual and predicted values irrespective of whether the predicted value in is the correct class or not.

8 Conclusions and Further Research

Probabilistic approached described in Section 5.4 seems to be a better technique to predict the trading signals of the AORD, than PNN and FNN. The criterion applied to determine the trading signals may also contributed to the effectiveness of this approach. This criterion uses the conditional probability that a relative return is in upward trend.

In general, the prediction results were better when the quantified intermarket influence on the AORD used as the input variables, than when their un-quantified counterparts were used as input variables. The exceptional behaviour of the FNN may be due to the inappropriateness of its error function for a classification problem.

Designing new neural network algorithm with appropriate error function, for predicting trading signals may be a good direction for future research. Such error function can be proposed by introducing a penalty to the Ordinary least squares error function, to deal with incorrectly predicted trading signals.

References

Becker, K. G., Finnerty, J. E. and Gupta, M. (2003), 'The Intertemporal Relation Between the U. S. and Japanese Stock markets', *The Journal of Finance* **XLV**(4), 1297-1306.

Bhattacharyya M. and Banerjee, A. (2004), 'Integration of Global Capital Markets: An Emphirical Exploration', *International Journal of Theoretical and Applied Finance* **7**(4), 385-405.

Cao, L. and Tay, F. E. H. (2001), 'Financial forecasting using support vector machines', *Neural Computing and Applications* **10**, 184-192.

Chapman, A. J. (1994), 'Stock market trading systems through neural networks: developing a model.', *International Journal of Applied Expert Systems* **2**(2), 88-100.

Chen, A. S., Leung, M. K. and Daouk, H. (2003), 'Application of neural networks to an emerging financial market: forecasting and trading the Taiwan Stock Index', *Computers and Operations Research* **30**, 901-923.

Chenoweth, T., Obradovic Z. and Stephenlee, S. (1996), 'Embedding Technical Analysis into Neural

Network Based Trading Systems', *Applied Artificial Intelligence* **10**, 523-541.

Demuth, H., Beale, M. and Hagan, M. (2006), *Neural Network Toolbox User's Guide*, The Math Works Inc..

Eun C. S. and Shim, S. (1989), 'International Transmission of Stock Market Movements', *The Journal of Financial and Quantitative Analysis* **24**(2), 241-256.

Fernando, F. R, Christian, G. M. and Simon, S. R. (2000), 'On the profitability of technical trading rules based on artificial neural networks: Evidence from the Madrid stock market', *Economics Letters* **69**, 89-94.

Kaastra, I. and Boyd, M. (1996), 'Designing a neural network for forecasting financial and economic time series', *Neurocomputing* **10**, 215-236.

Kim, S. H. and Chun S. H. (1998), 'Graded forecasting using an array of bipolar predictions: Application of probabilistic neural networks to a stock market index', *International Journal of Forecasting* **14**, 323-337.

Koubor, S., Ugon, J., Mammadov, M. A., Rubinov, A. M. and Kruger, A. (2006), Coverage in WLAN: Optimization Model and Algorithm, *in* 'First IEEE International Conference on Wireless Broadband and Ultra Wideband Communications', Sydney, Australia.

Kohara, K., Ishikawa, T., Fukuhara, Y. and Nakamura, Y. (1997), 'Price Prediction Using Prior Knowledge and Neural Networks', *Intelligent Systems in Accounting, Finance and Management* **6**, 11-22.

Kuo, R. J. (1998), 'A Decision Support System for the Stock Market Through Integration of Fuzzy Neural Networks and Fuzzy Delphi', *Applied Artificial Intelligence* **12**, 501-520.

Leung, M. T., Daouk, H. and Chen, A. S. (2000), 'Forecasting stock indices: a comparison of classification and level estimation models', *International Journal of Forecasting* **16**, 173-190.

Mammadov, M. A.,(2004), A new global optimization algorithm based on dynamical systems approach, *in* A. Rubinov and M. Sniedovich eds, 'The Sixth International Conference on Optimization: Techniques and Applications (ICOTA6)', Ballarat, Australia.

Mammadov, M. A., Rubinov A. M. and Yearwood, J. (2005), Dynamical systems described by relational elasticities with applications to global optimization, *in* V. Jeyakumar and A. Rubinov, eds, 'Continuous Optimisation: Current Trends and Applications', Springer, pp. 365-387.

Mizuno, H., Kosaka, M. and Yajima, H. (1998), 'Application of Neural Network to Technical Analysis of Stock Market Prediction', *Studies in Information and Control* **7**, 111-120.

Nguyen, D. and Widrow, B. (1990), 'Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights', *Proceedings of the Internationsl Joint Conference on Neural Networks (IJCNN)*, **3**, 21-26.

Olson D. and Mossaman, C. (2001), 'Cross-correlation and Predictability of Stock Returns', *Journal of Forecasting* **20**, 145-160.

Pan, H., Tilakaratne, C., and Yearwood, J. (2005), 'Predicting the Australian Stock Market Index Using Neural networks Exploiting Dynamical Swings and Intermarket Influences', *SJournal of Research and Practice in Information Technology* **37**, 43-55.

Riedmiller, M. and Braun, H. (1993), A direct adaptive method for faster backpropagation learning: The RPROP algorithm, *in* 'The IEEE International Conference on Neural Networks', California, USA.

Taylor, M. P. and Tonks, I. (1989), 'The Internationalisation of Stock Markets and the Abolition of U.K. Exchange Control', *Review of Economics and Statistics* **71**, 332-336.

Thawornwong, S. and Enke, D. (2004), Forecasting stock returns with artificial neural networks, *in* G. P. Zhang eds, *Neural Networks in Business Forecasting* , chapter 3, Idea Group Publishing.

Tilakaratne, C. D (2004), A Neural Network Approach for Predicting the Direction of the Australian Stock Market Index, MIT (by research), University of Ballarat, Australia.

Tilakaratne, C. D. (2006), A Study of Intermarket Influence on the Australian All Ordinary Index at Different Time Periods, *in* 'Second Australian Business and Behavioural Seiences Association (ABBSA) International Conference', Adeliade, Australia.

Tilakaratne, C. D., Mammadov M. A. and Hurst, C. P. (2006), Quantification of Intermarket Influence Based on the Global Optimization and Its Application for Stock Market Prediction, *in* K-L. Ong, K. Smith-Miles, V. Lee & W-K. Ng eds, 'International Workshop on Integrating AI and Data Mining (AIDM'06)', IEEE Computer Society, California, USA, pp. 42–49.

Tilakaratne, C. D., Morris S. A., Mammadov M. A. and Hurst, C. P. (2007), Quantification of Intermarket Influence on the Australian All Ordinary Index Based on Optimization Techniques, *in* W. Read eds, 'Proceedings of the 13th Biennial Computational Techniques and Applications Conference (CTAC'06)', *ANZIAM Journal*, **48**, C104-C118.

Vanstone, B. (2006), Trading in the Australian stockmarket using artificial neural networks, Ph.D., Bond University, Australia.

Wood, D. and Dasgupta, B. (1996), 'Classifying Trend Movements in the MSCI U.S.A. Capital Market Index - A Comparision of Regression, ARIMA and Neural Network Methods', *Computers & Operations Research* **23**(6), 611-622.

Wu, C. and Su, Y. (1998), 'Dynamic Relations among International Stock Markets', *International Review of Eeconomic and Finance* **7**(1), 63-84.

Yang, J., Khan M. M. and Pointer, L. (2003), 'Increasing Integration Between the United States and Other International Stock Markets?; A Recursive Cointegration Analysis', *Emerging Markets Finance and Trade* **39**(6), 39-53.

Yao, J., Tan, C. H. and Poh, H. L. (1999), 'Neural Networks for Technical Analysis: A Study on KLCI', *International Journal of Theoretical and Applied Finance* **2**(2), 221-241.

Yao, J. and Tan, C. H. (2000), Time Dependent Directional Profit Model for Financial Time Series Forecasting, *in* 'IEEE-INNS-ENNS Internationsl Joint Conference on Neural Networks (IJCNN'00)', Como, Italy.

Adaptive Spike Detection for Resilient Data Stream Mining

Clifton Phua[1] **Kate Smith-Miles**[2] **Vincent Lee**[1] **Ross Gayler**[3]

[1] Clayton School of Information Technology
Monash University,
Clayton, Victoria, Australia 3800,
Email: clifton.phua@infotech.monash.edu.au, vincent.lee@infotech.monash.edu.au

[2] School of Engineering and Information Technology
Deakin University,
Burwood, Victoria, Australia 3125,
Email: katesm@deakin.edu

[3] Veda Advantage
Level 12, 628 Bourke Street
Melbourne, Victoria, Australia 3000,
Email: ross.gayler@vedaadvantage.com

"A system's resilience is the single most important security property it has."

- Bruce Schneier, 2003, "Beyond Fear: Thinking Sensibly about Security in an Uncertain World"

Abstract

Automated adversarial detection systems can fail when under attack by adversaries. As part of a resilient data stream mining system to reduce the possibility of such failure, adaptive spike detection is attribute ranking and selection without class-labels. The first part of adaptive spike detection requires weighing all attributes for spiky-ness to rank them. The second part involves filtering some attributes with extreme weights to choose the best ones for computing each example's suspicion score. Within an identity crime detection domain, adaptive spike detection is validated on a few million real credit applications with adversarial activity. The results are F-measure curves on eleven experiments and relative weights discussion on the best experiment. The results reinforce adaptive spike detection's effectiveness for class-label-free attribute ranking and selection.

Keywords: adaptive spike detection, resilient data mining, data stream mining, class-label-free attributes ranking and selection

1 Introduction

Adversarial detection systems are fraud and crime detection, and other security systems. Our main concern here is when adversaries focus their attack on certain attributes (also known as fields, variables, and features), the weights (importance) of attributes can change quickly.

Data stream mining (Kleinberg 2005) involves detecting real-time patterns to produce accurate suspicion scores (which are indicative of anomalies). At the same time, the detection system has to handle continuous and rapid examples (also known as records, tuples, and instances) where the recent examples have no class-labels.

The work here is motivated by identity crime detection, or more specifically, credit application fraud detection (Phua et al. 2005) (also known as white-collar crime) . When adversaries manipulate real-time data, these detection systems can fail badly, if not completely. First, this is caused by too many new and successful attacks which are detected too late. Second, this is due to time delays in manual intervention by trusted people when new attacks are detected and underway. Resilient data stream mining is necessary to prevent failure of detection systems. It is the security systems' ability to degrade gracefully, or to adjust to changing circumstances when under attack (Schneier 2003).

Resilient data stream mining requires a series of multiple, independent, and sequential layers in a system. This is termed "defence-in-depth". These layers interact with each other to deal with the new and deliberate attacks, and make it much harder for persistent adversaries to circumvent the security system (Schneier 2003).

For example, there is a need to protect the personal identity databases of financial institutions. They contain individual applicants' details from real identity theft and synthetic identity fraud. The former refers to innocent peoples' identity details being used illegitimately by adversaries without their permission. The latter refers to non-existent peoples' identity details being created by adversaries to cheat assessment procedures. Three of our proposed identity crime detection procedures are:

- Known fraud matching (Phua et al. 2005) as **first-layer defence** - it is effective for repetitive frauds and real identity theft. However, there is a long delay between time that the identity is stolen and time the identity is actually reported stolen. This allows adversaries to use any stolen identity quickly before being discovered.
- Communal detection (Phua et al. 2006b) as **second-layer defence** - it utilises an example-based approach (similar to graph theory and record linkage) by working on a fixed set of attributes. It reduces the significant time delay and false alarms by filtering normal human relationships with whitelists to save significant money. In addition, it is good for new, duplicative frauds and synthetic identity fraud. But communal de-

tection is domain-specific, inflexible, and computationally slow.

- Spike detection (Phua et al. 2006a) as **third-layer defence** - it uses an attribute-oriented approach (similar to time series analysis) by working on a variable-size set of attributes. It reduces significant time delay by searching for recent duplicates. In comparison to communal detection without blocking (Baxter et al. 2003), spike detection with string similarity matching is computationally faster.

There is a fundamental security flaw with our communal detection framework (Phua et al. 2006b). It captures a substantial amount of frauds by filtering innocent relationships and data errors. However, if three or more values of each current identity are exact or similar to a window of previously collected identities, then a numeric suspicion score is produced. However, this encourages adversaries to quickly duplicate one and/or two important values which have been successful in their previous attempts. Spike detection can overcome this weakness by monitoring on one or two of the most important attributes.

Spike detection, with exact matching on a few of the most important attributes is much faster than communal detection. In hindsight, these important attributes have few missing values, are not de-identified, and string similarity matching can be performed (anonymised from identity strings to unidentifiable numbers). These attributes are appropriate ("not-too-dense and not-too-sparse") so that once they become dense, they become suspicious. They are also easiest to investigate (contacting the person or verifying against other databases). In contrast, communal detection scans many attributes of each identity as they arrive continuously and rapidly, so the system can become too slow with increased volume.

Both spike and communal detection have their own unique advantages. Spike detection can be computed in parallel on multiple workstations, each on an attribute. Also, the spike detection's relative weights can be applied to the communal detection's attributes. Communal detection is more accurate as it filters real communal relationships and data errors from other more anomalous exact and approximate duplicates. In contrast, spike detection can only remove duplicates based on each attribute.

At the highest level, resilient data stream mining also protects each security layer individually in the system. This is introduced in this paper as "layer reinforcement". This is to reduce the effects of attacks on security layers by persistent adversaries. For example, each of our identity crime detection procedure is defended by:

- Personal name analysis (Phua et al. 2006) as the **second-layer reinforcement** - it focuses on verifying and extracting information from personal names to improve known fraud matching.
- Adaptive communal detection (Phua et al. 2007) as the **third-layer reinforcement** - it has been proposed to prevent tampering of whitelists in communal detection.
- Adaptive spike detection is proposed in this paper as the **fourth-layer reinforcement** - static spike detection will be probed by persistent adversaries and the exact attributes monitored will eventually be known and circumvented with more duplicated values of the other attributes instead. To dynamically adapt to these adversaries' counter-measures, this paper formulates two related and consecutive challenges in adaptive spike detection.

The first challenge is to rank all attributes with class-labels. Although classification algorithms provide accurate attribute ranking with the benefit and clarity of hindsight, three main problems exist with using class-label attribute ranking for security detection systems:

- **Untimely:** There are time delays in labeling examples as positive because it takes time for fraud/crime to be revealed. This provides a window of opportunity for adversaries. Class-label attribute ranking is also computationally slow in an operational event-driven environment which requires efficient processing and rapid decision making (Phua et al. 2005).
- **Incomplete:** The positive class can be significantly understated. The case where the system labels current and prior positives but not future ones is common (Phua et al. 2006c). It is also possible that some of the data sources do not contribute positive class-labels.
- **Un-reliable:** The class-labeling is highly dependent on people. Each example has to be manually labeled and this has the potential of breaching privacy particularly if the data contains identity information. In addition, human factors can result in class-labels being incorrect, expensive, and difficult to obtain (Phua et al. 2005).

The second challenge is to use some attributes to detect attacks. The argument here is that two extreme (can also be known as redundant) types of attributes do not provide any symptoms of attacks:

- **Densest attributes:** They are attributes with highest weights (also known as densest; most spiky, duplicative, repetitive, and highest regularity and frequency attributes). They are usually numerical and have a finite number of values and therefore occur more frequently (for example, street numbers and postcodes).

  Since all their values are already highly duplicative, the system cannot find significantly denser values which are highly indicative of suspicious activity.
- **Sparsest attributes:** They are attributes with the smallest weights. They usually consist of string occurrences and identifiers with an enormous number of possible values which can occur at widely spaced intervals (for example, personal names and telephone numbers).

  Since their values do not re-occur or occur so rarely, the system cannot detect many attacks.

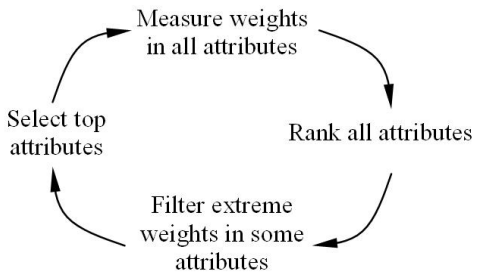


Figure 1: Attribute selection cycle

Figure 1 gives a visual account of the general iterative steps to ensure good class-label-free attribute ranking and selection. There are two research questions for adaptive spike detection for security systems:

Question 1 - How does the system empirically measure the weights of all attributes and rank them without the class-label attribute?

Question 2 - How does the system systematically filter redundant weights and select some appropriate attributes to calculate the suspicion score?

Section 2 outlines related work. Section 3 introduces the re-designed spike detection framework; and explains its resilient version in Section 4. Section 5 describes the identity crime detection domain, electronic credit application data, experimental design, and the performance metric; followed by results and discussion in Section 6. Section 7 concludes the paper.

2 Related Work

Spike detection is inspired by Stanford Stream Data Manager (STREAM) (Balakrishnan et al. 2004) and AURORA (Arasu et al. 2003) which deal with a rapid and continuous stream with structured examples by executing continuous queries. STREAM uses the SQL extension - Continuous Query Language (CQL) - in the extraction of example-based (or time-based) sliding windows, optional elimination of exact duplicates, and enforcement of increasing time-stamp order in queues. AURORA has been applied to financial services, highway toll billing, battalion and environmental monitoring.

Analysing sparse attributes exists in document and bio-surveillance streams. Kleinberg (2005, 2002) surveys threshold-, state-, and trend-based stream mining techniques used in topic detection and tracking. Goldenberg et al. (2002) use time series analysis to track early symptoms of synthetic anthrax outbreaks from daily sales of retail medication (throat, cough, and nasal) and some grocery items (facial tissues, orange juice, and soup).

Adaptive spike detection is easily extensible. In addition to identity crime detection, they are useful to other well-known security domains which profile suspicious behaviour or activity. These domains also aim to detect early irregularities in temporal data with unsupervised techniques. They include, but not limited to:

- **Bio-terrorism (also known as syndromic surveillance, aberration, and outbreak) detection:** Control-chart-based statistics, exponential weighted moving averages, and generalised linear models were tested on the same benchmark data and fixed alert rate. The conclusion was that these techniques perform well only when there are rapid and large increases in duplicates relative to the baseline level (Jackson et al. 2007). Bayesian networks were applied to uncover simulated anthrax attacks from real emergency department data (Wong et al. 2003).
- **Credit transactional account fraud detection:** Peer Group Analysis is recommended to monitor inter-account behaviour over time and also suggest Break Point Analysis to monitor intra-account behaviour over time (Bolton & Hand 2001).
- **Spam detection:** The use of document space density (class-labels are avoided) to find large volumes of similar emails is encoded as hash-based text (Yoshida et al. 2004).

3 Spike Detection Framework

The spike detection framework (Phua et al. 2006a) monitors streams to find recent and denser values in attributes (which can be highly indicative of identity crime). In other words, spiky-ness of each attribute is more than the number of exact and approximate values - it also factors in the recency of the duplicates. The more current the duplicates, the more interesting or suspicious they become. The framework basically uses exponential smoothing to find single attribute value spikes, and integrates the multiple attribute value spikes to score each example.

Input	Process	Output
Current example d_n Window W Window steps k Exponential smoothing α Similarity threshold r Time filter e	Process each current value against previous values within an example-based window Step 1: Calculate scaled counts of current value by comparison to previous examples window[1] Step 2: Calculate smoothed score on current value[2] Step 3: Calculate suspicion score on current example[3]	Score

Table 1: Parameters, spike detection, and suspicion score

Table 1 gives an overview of the input parameters, spike detection process/algorithm, and output suspicion score.

3.1 Step 1: Scaled Counts in Single Step

Let Y represent one continuous stream with an ordered set of $\{\ldots, y_{j-2}, y_{j-1}, y_j, y_{j+1}, y_{j+2}, \ldots\}$ discrete streams. Let y_j represent a current discrete data stream with an ordered set of $\{d_{j,n}, d_{j,n+1}, d_{j,n+2}, \ldots, d_{j,n+p}\}$ examples to be processed in real time. For simplicity, the subscript j is omitted. Each current example d_n (to be scored) contains M chosen attributes with a set of $\{a_{n,1}, a_{n,2}, \ldots, a_{n,M}\}$ values. Let W represent the window size of number of previous examples to match against.

Let r represent the string similarity (also known as fuzzy matching) threshold between values where $0 < r \leq 1$. 0 is a complete mismatch and 1 is an exact match. The fast string similarity metric Jaro-Winkler (Winkler 2006) is used. Let e represent the time difference filter (for example, seconds, minutes, and hours) between values to improve data quality where $0 \leq e <\leq$ inf. 0 means no filter and inf means all previous values are filtered.

In addition to a larger number of attributes M or window size W, lower string similarity threshold r or time difference filter e might also produce higher suspicion score for an example $S(d_n)$. Each current value a_{n-1} is being searched for its exact or approximate values in $\{a_{n-1,i}, a_{n-2,i}, \ldots, a_{n-W,i}\}$ where the matches' string similarity and time difference are larger than r and e.

$$s_x(a_{n,i}) = count_x(a_{n,i})/k, \text{ for } x = 1, 2, \ldots, t \quad (1)$$

In reference to Phua et al. (2006a): [1]At step 1, no form of random sampling is used except to filter out six dummy and three hashed-addresses subscribers. [2]At step 2, "smoothing level", "spiking alpha", and four other optional parameters are removed to simplify work. [3]At step 3, explicit normalisation of scores to 0 and 1 is not necessary.

Equation 1 is the scaled counts for each step $s_x(a_{n,i})$ (a window is made up of many steps) to remove volume effects where $0 \leq s_x(a_{n,i}) \leq 1$. Each W is divided into t steps (number of blocks of consecutive values) of k step size (maximum number of values in each block). t is also the most recent time step.

3.2 Step 2: Spike Detection of Single Value

$$S(a_{n,i}) = \sum_{x=1}^{t}[(1-\alpha) \times s_x(a_{n,i}) + \alpha \times s_{x-1}(a_{n,i})] \quad (2)$$

Equation 2 is the exponential smoothing of each value (all steps) to determine spike score $S(a_{n,i})$ for weighing current examples more heavily or previous examples more lightly (Cortes et al. 2003) where $0 \leq S(a_{n,i}) \leq 1$. α is exponential smoothing factor to gradually discount the effects of previous older steps and $0 \leq \alpha \leq 1$.

3.3 Step 3: Suspicion Score from Multiple Values

$$S(d_n) = \sum_{i=1}^{M} S(a_{n,i}) \quad (3)$$

Equation 3 sums up all the spike scores to derive a suspicion score for each example $S(d_n)$ where $0 \leq S(d_n) \leq M$.

3.4 A Simple Spike Detection Illustration

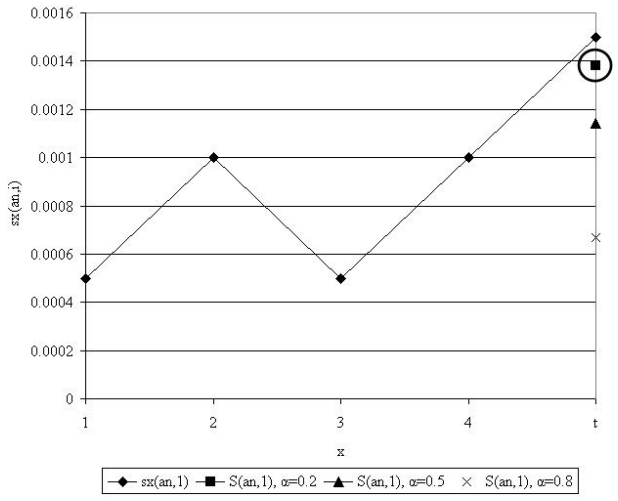


Figure 2: Scaled Counts and Spike Detection

Figure 2 demonstrates steps 1 and 2 of the spike detection framework. The y-axis represents the scaled counts; and the x-axis represents the steps. In the illustration above, given that the parameters' values are $W = 10,000, t = 5, k = W/t = 2,000$, therefore $count_{1,2,...,5}(a_{n,1}) = 1, 2, 1, 2, 3$.

Step 1: Scaled Counts $count_{1,2,...,5}(a_{n,1}) = 0.0005, 0.001, 0.0005, 0.001, 0.0015$ is represented by the line.

Step 2: Spike Detection $S(a_{n,1}) = 0.0013$ is the smoothed spike score (circled point) with the lowest weight on the previous examples ($\alpha = 0.2$).

Figure 3 shows step 3 of the spike detection framework. The y-axis represents the cumulative suspicion score; and the x-axis represents the number of attributes. For example, given that $S(a_{n,1,2,...,5}) = 0.0013, 0.0129, 0.00543, 0.0511, 0.0732$.

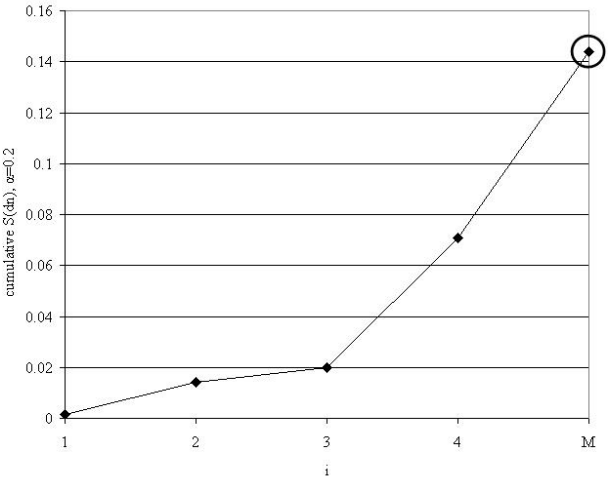


Figure 3: Suspicion score

Step 3: Suspicion Score $S(d_n) = 0.0013 + 0.0129 + 0.00543 + 0.0511 + 0.0732 = 0.144$ (circled point).

4 Adaptive Spike Detection

With adversarial activity in mind, the weight of each attribute is measured regularly at either fixed time or fixed example intervals (for example, after each month or after every ten thousand examples) to re-weigh and re-rank all attributes. To be more specific, each attribute is measured by relative weights at every interval.

From all attributes' weights, attributes with highest weights (densest) and lowest weights (sparsest) are filtered. Therefore, suspicion scores are computed from "not-too-dense and not-too-sparse" attributes with fewer missing values (so that the scores are more accurate).

In this way, when these appropriate attributes' weights suddenly become larger, it creates a spike in the time series, making them more interesting or suspicious. To be precise, attributes with relative weights which exceed one standard deviation or below half of average will have their weights set to zero. In this way, only some attributes are re-chosen and have their corresponding non-zero weights factored into the suspicion score.

4.1 Initialisation of Weights

$$\bar{w}_i = 1/M \quad (4)$$

When there are no prior weights, Equation 4 uses average/equal weights for all attributes.

4.2 Application of Weights

$$S(d_n) = \sum_{i=1}^{M}[\hat{w}_i \times S(a_{n,i})] \quad (5)$$

When processing each example, Equation 5 which is an extension from Equation 3, applies relative weights to all corresponding attribute values of each example.

4.3 Evolution of Weights

$$w_i = \sum_{n=1}^{p} S(a_{n,i})]/n \quad (6)$$

When updating weights at the end of each interval, Equation 6 represents the absolute/total weights per example.

$$\bar{w}_i = w_i / \sum_{i=1}^{M} w_i \tag{7}$$

Modified from Equation 6, Equation 7 represents the relative weights.

$$\hat{w}_i = \begin{cases} \bar{w}_i & \text{if average weight}/2 \leq \bar{w}_i \\ & \leq \text{average weight} + \text{standard deviation} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

Modified from Equation 7, Equation 8 symbolises the filtered relative weights which are actually applied to the attributes in Equation 5. Equation 8 retains only the relative weights of attributes which remain within the lowerbounds (average weight is usually low) and upperbounds (to exclude only the densest attributes), and removes attributes with extreme relative weights by setting zero weights.

4.4 A Simple Weights Illustration

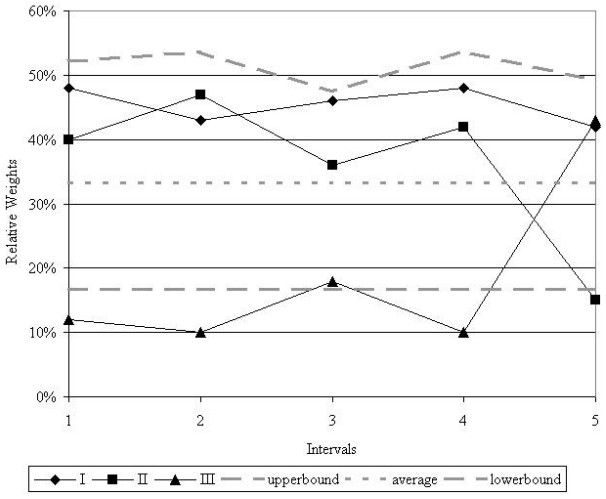


Figure 4: Evolution of relative weights

Figure 4 explains the concept of attributes' relative weights and ranks changing over time. The y-axis represents relative weights - the density/spiky-ness of three attributes - I II, and III - with respect to one another; and the x-axis represents the time interval (for example, hours, days, and months).

In the first four intervals, attributes I and II are ranked higher than III. As they are also within the upper and lower boundaries, they are chosen to calculate the suspicion score. However, in the last interval, attribute II loses its density at the expense of III and falls below the lower bound limit. As attribute III suddenly becomes significantly denser (deemed as anomalous and quite possibly a new attack), therefore it is used together with attribute I to calculate subsequent suspicion scores.

5 Identity Crime Detection

Adaptivity is about helping the system adjust and function well within a changing environment. A data streaming environment is a rapidly changing one and the weights (importance) of attributes do not remain static. Therefore, spike detection is necessary to measure an attribute value's regularity within its attribute's recent times and represent them as weights relative to all other weights.

In addition, adversaries gather information about previous parameters of the spike detection framework to choose attributes that attempt to force the worst-case result. Those attributes with the appropriate amount of regularity for detecting suspicious behaviour do change in a principled fashion. Knowing them in time are additional defences against adversaries.

Therefore, adaptive spike detection is a specific case of resilient data stream mining. It deals with our adversaries who have a tendency to re-use identity values of certain attributes in a bursty manner. In addition, it also copes well when adversaries change their focus to other attributes. In this way, the adversaries are more likely to get relatively higher suspicion scores which are directly correlated with the risk of identity crime (see Figure 6).

5.1 Data and Evaluation Metrics

There are five main technical challenges posed by data used in the following experiments:

- **Large scale:** Thirteen months of several million real credit applications (only the last seven months are used in the experiments described here because they have the most complete class labels). Every month is made up of a few hundred thousand applications and every day has more than ten thousand applications. The data is recent, consecutive, and time-stamped to the milliseconds.
- **Dense and sparse attributes:** About thirty raw attributes such as personal names, addresses, telephone numbers, driver licence numbers (or social security numbers), date-of-births, and other personal identifiers. Some of these personally identifying attributes were encrypted prior to this study to preserve privacy. Encrypted attributes can be exactly matched, but in a real application unencrypted attributes would be used to allow approximate matching. For confidentiality reasons, we cannot specify the best attributes found in this study for credit application fraud detection.
- **Extreme class imbalance:** Less than one percent of these are known to be fraudulent in binary class-labeled (as "fraud" or "legal") data. Also, the earliest and latest months' known fraud rate is significantly understated as not all known frauds were provided for this research.
- **Diverse data sources:** A few dozen financial institutions are providers of examples. Each provider has varying arrival rates, has sudden behavioural changes, and contributes their own number and type of attributes, and adds data quality problems.
- **Few significant fraud patterns:** For the period under analysis, relational (links between examples), temporal (for example, hourly, daily, and monthly), spatial (for example, suburb, country, and state), and provider-related fraud behaviour are hard to differentiate from legitimate behaviour.

For evaluation metrics, precision-recall curves are avoided as they will divulge the sensitive nature of the true positive tp, false positive fp, and false negative fn rates.

Also, metrics which use true negatives tn such as accuracy and receiver operating characteristic curves are avoided since fp rates are likely to be understated (Christen & Goiser 2007).

$$F\text{-measure curve} = \frac{2 \times precision_X \times recall_X}{precision_X + recall_X} \quad (9)$$

The F-measure curve in Equation 9 consists of multiple values under X different thresholds. Each value depicts a trade-off between $precision_X = \frac{tp}{tp+fp}$ and $recall_X = \frac{tp}{tp+fn}$.

5.2 Experimental Design

In the following spike detection experiments, we are particularly interested in finding out which are the best attributes, out of a total of 19, for detecting identity crime. To do so, the parameters, applied to each attribute, for all the following experiments remain unchanged:

- Window $W = w/100$, Window Steps $k = 10$
- Exponential Smoothing $\alpha = 0.5$
- Similarity Threshold $r = 0.8$, Time Filter $e = 1$ hour

Some of their values, such as w, $\alpha = 0.5$, and $r = 0.8$, are based on our previous experimental experience and current practical domain knowledge. However, in comparison with our previous experiments in communal and spike detection, parameter values of W and k here are much smaller - W is 100 times smaller, k is 10 times smaller - to compare fewer examples. String similarity matching can be performed on 10 attributes; but cannot be applied to the other 9 because they seem to be too dense and/or are de-identified. Also, e is larger to filter more examples.

The use of small W and k values are due to the very high computational cost in applying string similarity to all comparisons for the 10 attributes. In addition, the experiments are meant to illustrate the concepts of adaptive spike detection. Also, for confidentiality reasons, these experiments do not reveal the results of a realistic W and k.

t Exp.	Attribute(s)
t1	I
t2	IV
t3	V
t4	XIV
t5	XVII
t6	XVIII
t7	XIX
t8	All-static
t9	2-static (XIV & XVIII)
t10	All-monthly
t11	2-monthly

Table 2: Static (t1 to t9) and adaptive (t10 and t11) spike detection experiments to test predictability of attributes

Table 2 show that experiments t1 to t9 are static where the relative weights are not used. t1 to t7 uses individual attributes regarded as useful from domain knowledge. t8 uses all 19 attributes. t9 uses only top 2 attributes (XIV and XVIII as advised by domain experts) throughout all the data.

Experiments t10 and t11 are adaptive where the relative weights change at a monthly interval. t10 answers Question 1 by measuring weights and ranking all attributes monthly without the class-label attribute. t11 answers both Questions 1 and 2 by filtering extreme ranked weights and then choosing the top 2 attributes (either XII, XIV, XIX, or III according to the highest unfiltered relative weights) monthly to calculate the suspicion score.

6 Results and Discussion

With F-measure over different thresholds, results are presented from seven important attributes, and justifies importance of the adaptive spike detection framework to measure and rank attributes. With relative weights, the useful role of adaptive spike detection to filter and choose attributes are verified.

6.1 Spike Detection Results

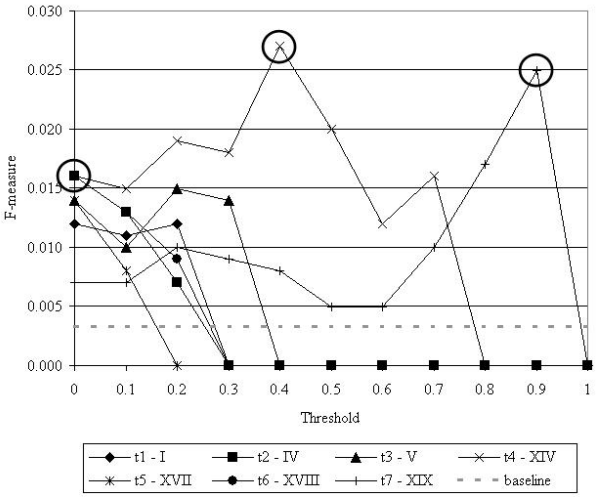


Figure 5: F-measure across 11 thresholds, of spike detection experiments t1 to t7 of individual attributes

Figure 5 illustrates the F-measure results over different thresholds of seven valuable single attributes. The most predictive attribute by spike detection is t4 - XIV with F-measure above 0.025 at threshold 0.4. This fact is also acknowledged by domain experts.

There is a need to automatically filter out most attributes for calculation of the suspicion score. Although all the other individual attributes are better than the baseline (random) at threshold 0, most are much poorer attributes compared to attribute XIV across most thresholds.

Spike detection is practical for our security domain. Two other predictive attributes revealed by spike detection include t7 - XIX and t6 - XIV. At the higher thresholds, the attribute XIX yield better results than all other attributes. Attribute XIX is another predictive attribute acknowledged by domain experts.

Figure 6 illustrates the F-measure results over big and small sets of attributes, static and adaptive. From observation, the most predictive set of attributes is t11 - 2-monthly with F-measure above 0.025 at threshold 1.

Finding the right set of appropriate attributes is crucial. This is illustrated by two facts from the F-measure results: First, the use of 2 attributes (static or adaptive) performs better than using all attributes. Second, 2-monthly is superior to using just the most predictive attribute XIV with the former's F-measure above 0.02 for most thresholds.

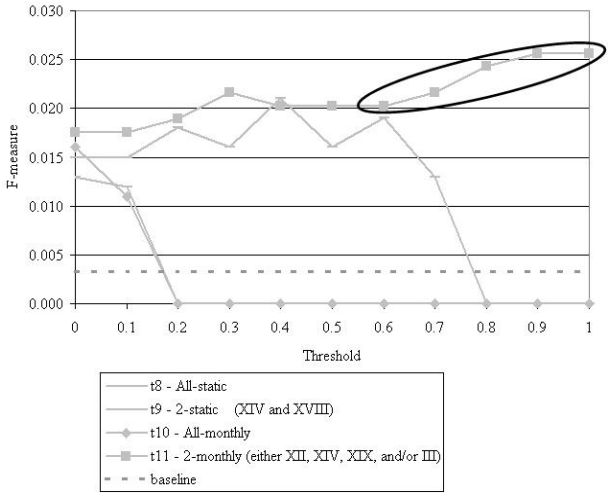


Figure 6: F-measure across 11 thresholds, of static (t8 and t9) and adaptive (t10 and t11) spike detection experiments of multiple attributes

Adaptivity with changing relative weights for attributes provide better results than static attributes with no weights. This is substantiated by t11 - 2-monthly which outperforms the second-best result from t9 - 2-static by a large margin.

In reality, F-measure results will be higher. The current F-measure results are underestimated because many of the class-labels were still not known at the time when this data set was constructed. Hence, the F-measure performance metric evaluated predictions based on significantly smaller numbers of positive class-labels (Phua et al. 2007).

6.2 Relative Weights Discussion

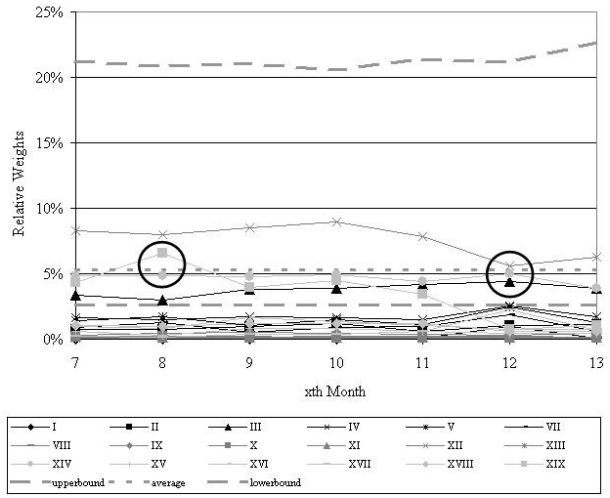


Figure 7: Relative weights across 7 months, of all attributes (except attribute VI) from experiment t11 - 2-monthly

Figure 7 highlights the relative weights which are within acceptable boundaries across seven months (the reason that a smaller set of attributes, surprisingly, performs better). Use of relative weights find the most appropriate attributes. Only a maximum of four attributes stay within the lower and upper-bounds for any given month. The two best attributes XIV and XIX which are tested in Figure 5 have the highest acceptable relative weights.

Relative weights change over time and so do appropriate attributes: attribute XIX overtook XIV during the 8th month as the top 2 attributes and it dropped out of the acceptable boundaries at the 12th month.

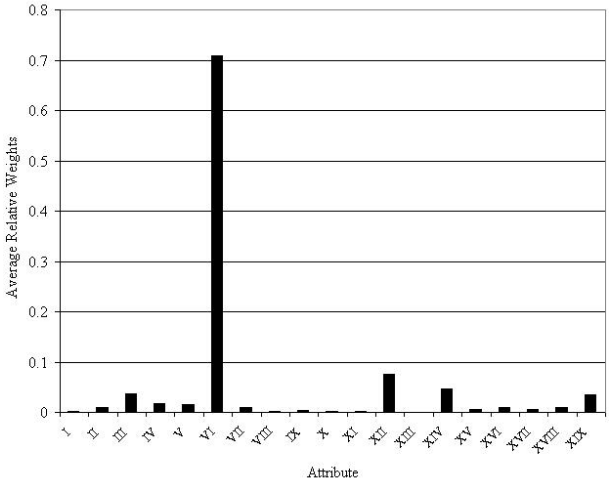


Figure 8: Average relative weights of all 7 months, of all attributes (inclusive of attribute VI) from experiment t11 - 2-monthly

Figure 8 focuses attention on the average relative weights of all seven months. The densest and sparsest attributes are not predictive. Attribute VI overspikes (highest weight). The other attributes do not spike enough (smallest weights). Yet, the densest and sparsest attributes cannot be discarded permanently because they can still become useful in the future.

There is a strong relationship between spiky-ness of certain attributes (represented by acceptable relative weights) and risk of identity crime (represented by class-labels). This is evident from the attributes XII, XIV, III, and XIX which are both spiky and predictive of fraud/crime. Therefore, the interpretation of results on substantial amount of historical data, modeled as data streams, justify that adaptive spike detection is significantly better than the static version (see Figure 6). As this idea is novel, attackers cannot apply what they have studied previously from elsewhere.

7 Conclusion

The overall goal is to propose resilient data stream mining for all data mining-based security systems with adaptive spike detection for attribute ranking and selection. The spike detection framework's parameters and suspicion score functions were significantly updated and made more resilient with the evolution of weights. In our identity crime domain, the challenges in our real data and the rationale for the evaluation measure were given. In the spike detection results, adaptive spike detection's attribute ranking and selection gave the best outcome. A deeper analysis of the relative weights showed that the most appropriate attributes were found.

8 Acknowledgements

The first author is financially supported by ARC under Linkage Grant Number LP0454077 and previously by DEST under an Endeavour Research Fellowship.

References

Arasu, A., Babcock, B., Babu, S., Datar, M., Ito, K., Nishizawa, I., Rosenstein, J. & Widom, J. (2003), STREAM: the stanford stream data manager demonstration description, *in* 'SIGMOD03'.

Balakrishnan, H., Balazinska, M., Carney, D., Cetintemel, U., Cherniack, M., Convey, C., Galvez, E., Salz, J., Stonebraker, M., Tatbul, N., Tibbetts, R. & Zdonik, S. (2004), 'Retrospective on aurora', *VLDB Journal* **13**(4), pp. 370–383.

Baxter, R., Christen, P. & Churches, T. (2003), A comparison of fast blocking methods for record linkage, *in* 'ACM SIGKDD03 Workshop on Data Cleaning, Record Linkage and Object Consolidation'.

Bolton, R. & Hand, D. (2001), Unsupervised profiling methods for fraud detection, *in* 'Credit Scoring and Credit Control VII'.

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.

Cortes, E., Pregibon, D. & Volinsky, C. (2003), 'Computational methods for dynamic graphs', *Journal of Computational and Graphical Statistics* **12**, pp. 950–970.

Goldenberg, A., Shmueli, G. & Caruana, R. (2002), 'Using grocery sales data for the detection of bioterrorist attacks', *Statistical Medicine*.

Jackson, M., Baer, A., Painter, I. & Duchin, J. (2007), 'A simulation study comparing aberration detection algorithms for syndromic surveillance', *BMC Medical Informatics and Decision Making* **7**(6).

Kleinberg, J. (2005), Temporal dynamics of on-line information streams, *in* M. Garofalakis, J. Gehrke & R. Rastogi, eds, 'Data Stream Management: Processing High-Speed Data Streams', Springer.

Kleinberg, J. (2002), Bursty and hierarchical structure in streams, *in* 'SIGKDD02'.

Phua, C., Lee, V., Smith-Miles, K. & Gayler, R. (2005), 'A comprehensive survey of data mining-based fraud detection research', *Artificial Intelligence Review*.

Phua, C., Lee, V., & Smith-Miles, K. (2006), 'The personal name problem and a recommended data mining solution', *Encyclopedia of Data Warehousing and Mining (2nd Edition)*.

Phua, C., Lee, V., Gayler, R., & Smith-Miles, K. (2006a), Temporal representation in spike detection of sparse personal identity streams, *in* 'PAKDD06 Workshop on Intelligence and Security Informatics'.

Phua, C., Gayler, R., Smith-Miles, K. & Lee, V. (2006b), Communal detection of implicit personal identity streams, *in* 'IEEE ICDM06 Workshop on Mining Evolving and Streaming Data'.

Phua, C., Gayler, R., Lee, V. & Smith-Miles, K. (2006c), 'On the communal analysis suspicion scoring for identity crime in streaming credit applications', *European Journal of Operational Research*.

Phua, C., Lee, V., Smith-Miles, K. & Gayler, R. (2007), Adaptive communal detection in search of adversarial identity crime, *in* 'ACM SIGKDD07 Workshop on Domain-Driven Data Mining'.

Schneier, B. (2003), *Beyond fear: thinking sensibly about security in an uncertain world*, Copernicus, New York.

Winkler, W. (2006), 'Overview of record linkage and current research directions', Statistical Research Division, U.S. Census Bureau Publication, RR 2006-2.

Wong, W., Moore, A., Cooper, G. & Wagner, M. (2003), Bayesian network anomaly pattern detection for detecting disease outbreaks, *in* 'ICML03', pp. 217–223.

Yoshida, K., Adachi, F., Washio, T., Motoda, H., Homma, T., Nakashima, A., Fujikawa, H., & Yamazaki, K. (2004), Density-based spam detector, *in* 'SIGKDD04', pp. 486–493.

Mining for offender group detection and story of a police operation

Fatih Ozgul [1] Julian Bondy [2] Hakan Aksoy [3]

[1] School of Computing and Technology,
St. Peter's Campus, University of Sunderland, SR6 0DD, UK
Email: fatih.ozgul@sund.ac.uk

[2] School of Global Studies, Social Science & Planning,
RMIT University, Melbourne, Australia
Email: bondy@rmit.edu.au

[3] Information Processing Unit, Bursa Police Department, Bursa, Turkey
Email:aksoy975@yahoo.com

Abstract

Since discovery of an underlying organisational structure from crime data leads the investigation to terrorist cells or organised crime groups, detecting covert networks are important to crime investigation. As shown in application of Offender Group Detection Model (OGDM), which is developed and tested on a theft network in Bursa, Turkey, use of effective data mining methods can reveal offender groups. OGDM detected seven ruling members of twenty network members. Based on initial findings of OGDM; thirty-four offenders are considered to be in a single offender group where seven of them were ruling members. After *Operation Cash* was launched, the police arrested the seven detected ruling members, and confirmed that the real crime network was consisting of 20 members of which 3 whom had never been previously identified or arrested. The police arrested 17 people, recovered worth U.S. $ 200,000 of stolen goods, and cash worth U.S. $ 180,000.

Keywords: crime data mining, group detection, social network analysis.

1 Introduction

Link analysis and group detection is a newly emerging research area which is at the intersection of link analysis, hypertext and web mining, graph mining (Cook and Holder, 2000) and social network analysis (Scott, 2004). Graph mining and social network analysis in particular attracted attention from a wide audience in police investigation and intelligence (Getoor et al., 2004). As a result of this attention, the police and intelligence agencies realized the knowledge about offender networks and detecting covert networks are important to crime investigation (Senator, 2005). Group detection refers to the discovery of underlying organisational structure that relates selected individuals with each other, in broader context; it refers to the discovery of underlying structure relating instances of any type of entity among themselves (Marcus et al., 2007). Since discovery of an underlying organisational structure from crime data leads the investigation to terrorist cells or organised crime groups, detecting covert networks are important to crime investigation. Detecting an offender group or even a part of group (subgroup) is also important and valuable. A subgroup can be extended with other members with the help of domain experts. An experienced police officer usually knows the friends of well-known offenders, so he can decide which subgroups should be united to constitute the whole group. Another outcome of offender group detection is considered to be pre-emptive strike or crime prevention. For example a drug dealing network prepares all required vehicles and people for transaction where all members are in the process of getting prepared. Such cases can be prevented with offender group detection before it happens. A further advantage of group detection is acting in a group of offenders to commit a crime is regarded as an aggravating factor for a heavier punishment in many country's laws. For instance, Turkish Crime Code extends six years imprisonment for group leader and one year imprisonment for group members plus the punishment.

Specific software like Analyst Notebook (2007), and Sentient (2007) provide some visual spatio-temporal representations of offender groups in graphs, but they lack automated group detection functionality.

In this paper, we make the following contributions for offender group detection (OGD);

- We identify and discuss converting arrest data to graph format where there is no standardised way of doing this. We suggest the choice of representation for edges and nodes should follow the rules in SNA where mostly one-mode social network representation which is now standard (section 4).

- We explain precisely how to use police arrest data to look for possible offender groups (section 5). Surprisingly this has not been explained precisely before.
- We show how we can apply filters to graph data in order to adhere to countries' criminal law requirements (section 7).
- We show that ruling members, not new recruits, are likely to be detected, but "big brother" of network is unlikely to be detected (section 8).

2 Group Detection

Group detection task is defined and different methods applied in data mining, in social network analysis, and in graph theory. For example, Getoor and Diehl (2005) state group detection aims clustering of object nodes in a graph into groups that share common characteristics. But to some extent, subgraph discovery does the same job for finding interesting or common patterns in a graph. On the other hand social network analysis tries to detect cohesive subgroups among which there are relatively strong, direct, intense, frequent, or positive ties (Wasserman and Faust, 1994). Graph matching (Cook and Holder, 2007) methods are also recommended for group detection tasks. There are also many specific group detection models. Adibi et al. (2004, 2005) propose KOJAK group finder which firstly positioning possible groups, expanding these groups using knowledge-based reasoning techniques and then adding more candidates relying on observed interactions that shows possible associations. Kubica et al. (2002, 2003) first proposes a generative model for multi-type link generation, called collaborative graph model (cGraph) and introduce a scalable group discovery algorithm called k-groups, which is similar to k-means algorithm.

3 OGD

When we focus on offender group detection, the most remarkable works are CrimeNet Explorer, which is developed by Xu et al. (2005), and Terrorist Modus Operandi Detection System (TMODS), which is developed by 21st Century Technologies (Moy, 2005).

3.1 CrimeNet Explorer

Xu et al. (2005) defined a framework for automated network analysis and visualization. Using COPLINK connect and COPLINK detect (Chen et al., 2002) structure to obtain link data from text, CrimeNet Explorer used an Reciprocal Nearest Neighbour (RNN) based clustering algorithm to find out links between offenders, as well as discovery of previously unknown groups. CrimeNet Explorer framework includes four stages: network creation, network partition, structural analysis, and network visualization. CrimeNet Explorer uses concept space approach for network creation, RNN-based hierarchical clustering algorithm for group detection; social network analysis based structural analysis, and multi dimensional scaling for network visualisation. CrimeNet Explorer is the first model to solve offender group discovery problem and its success comes from the powerful functionality of overall COPLINK structure. On the other hand, since CrimeNet Explorer was evaluated by university students for its visualization, structural analysis capabilities, and its group detection functionality, the operationally actionable outputs of CrimeNet Explorer has not been proved on real-time police investigations.

3.2 Terrorist Modus Operandi Detection System (TMODS)

TMODS, which is developed by 21st Century Technologies (Marcus et al., 2007), automates the tasks of searching for and analysing instances of particular threatening activity patterns. With TMODS, the analyst can define an attributed relational graph to represent the pattern of threatening activity he or she is looking for. TMODS then automates the search for that threat pattern through an input graph representing the large volume of observed data. TMODS pinpoints the subset of data that match the threat pattern defined by the analyst thereby transforming a manual search into an efficient automated graph matching tool. User defined threatening activity or pattern graph can be produced with possible terrorist network ontology and this can be matched against observed activity graph. At the end, human analyst views matches that are highlighted against the input graph. TMODS is mature and powerful distributed java software that has been under development since October 2001 (Marcus et al., 2007). But it needs a pattern graph and an analyst to run the system. Like a supervised learning algorithm, TMODS tries to tailor the results according to pre-defined threatening activity. Another possible drawback is graphs used in TMODS are multi-mode and can be disadvantageous for further analysis. Multi-mode graph means that nodes in multi-mode graphs are more than two types of entities. A person, a building, an event, a vehicle are all represented as nodes; when for instance we want to detect key players in multi-mode graph, a building can be detected as key player, not a person. This can be a cause of confusion. To overcome this confusion the definition of a one-mode (friendship) social network should be used rather than representing all entities as nodes.

4 Offender Group Representation

Wasserman and Faust (1994) pp.35 states that the modes of a network as the number of sets of entities on which structural variables are measured. One-mode (friendship) networks, the predominate type of network, study just a single set of actors while two-mode (affiliation) networks focus on two sets of actors, or one set of actors and one set of events. One could ever consider (three and higher) mode networks but rarely have social network methods has been designed for such complicated data structures. According to these definitions it is better to represent actors (offenders) as nodes and rest of the relations as edges in one-mode (friendship) social networks. This can produce many link types such as "co-defendant link", "spatial link", "same weapon link", and "same modus operandi link". Thereby many graph theoretical and SNA

solutions can be used on one-mode (friendship) networks effectively such as friendship identification, finding key actors.

5 Police Arrest Data

We recommend looking for common characteristics of offenders in police arrest data. Do they commit the same crime somewhere sometime together, and then any of these offenders has also committed another crime with another offender? This information can be obtained from a relational database table, text-based arrest report, or CCTV footage.

In *Operation Cash* we obtained this information from Bursa Police Arrest Data where the table included the fields for: P_ID (person id), C_ID (crime reference number), BRANCH (police branch that deals with), CRT_ID (Crime type it belongs to), CR (Name of the offence), MOT_ID (Modus Operandi it belongs to), MO (name of the modus operandi), D (date stamp), DIS (district), NG (neighbourhood), and NG_ID (neighbourhood number).

6 Offender Group Detection Model (OGDM)

OGDM is mainly developed for detecting gangs and theft networks. As exhibited in Figure 1. the source of link information is gathered from police arrest records where a link table; consisting of From (From Offender), To (To offender), and W (how many times this offender pair caught together by the police) is produced with an inner join SQL query.

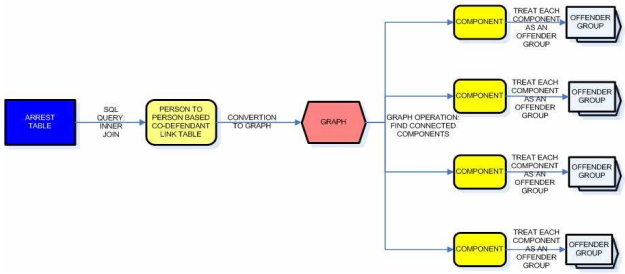


Figure 1

Inner join query result, which we call co-defendant link table, then converted to graph where nodes represent offenders, edges represent crimes committed together using offender group representation exhibited in section 4. Number of times caught together is counted to be used for edge weight (W). At this point a subgraph detection operation is needed; various social network analysis algorithms such as k-clique, k-core (Wasserman et al., 1994) can be used for this purpose. We used strongly connected components (SCC) algorithm in *Operation Cash* because it is scalable and gives concrete results. SCC algorithm is defined as (Cormen et al., 2001); a directed graph is called strongly connected if for every pair of vertices U and V in a graph there is a path from U to V and a path from V to U. The strongly connected components of a directed graph are its maximal strongly connected subgraphs.

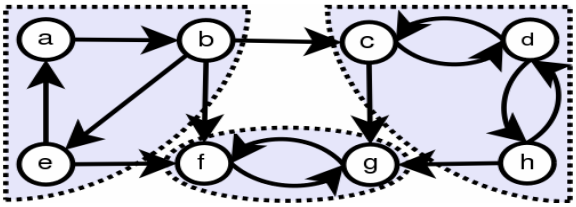


Figure 2. This figure shows graph with its strongly connected components are marked

In a graph generated from an arrest table where there are at least couple of hundred thousand of crimes (edges) and thousands of offender (nodes) makes scalability and performance issue very important. At last, every component represents a unique offender group because one offender can only belong to one group thereby concrete a result of group membership is obtained.

7 Filtering for Legal Requirements

Turkish Crime Code requires that an criminal organisation (offender group) must consist at least of three members, and two members in an offender group must have been convicted together for committing the same crime at least two times (Turkish Crime Code, Article Number:261). According to this definition, where edge weight is W and number of members is N;

$$W_{group} >= 2, N_{group} >= 3$$

is the threshold to constitute a criminal organisation. This requirement can be different in different countries but it is essential to create a filter for a legally accepted criminal organisation.

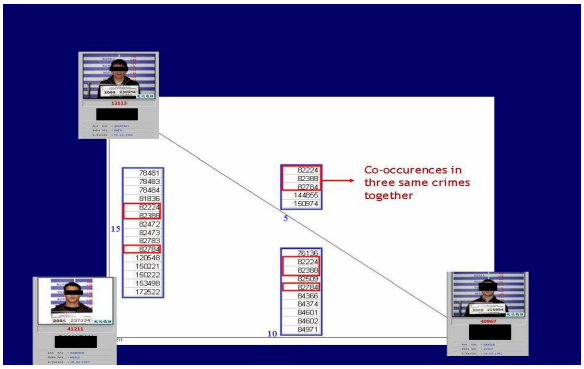


Figure 3. This triad of thieves committed various crimes together. The person in the top left has committed 15 crimes together (W=15) with the person in the left bottom and 5 crimes together (W=5) with the person in the right bottom. The person on the left bottom has also committed 10 crimes together (W=10) with the person on the right bottom. Overall, these three persons have committed 3 crimes together as a group which is shown in crime reference numbers 82224, 82388, and 80784 highlighted in red boxes.

8 Operation Cash

Offender group detection action is started with preparation of Bursa Police arrest data. Initial data pre-processing and data cleaning are done in cooperation with Bursa Police Department on more than 300000 crimes and 6000 offenders. Starting from 1994 to 2007, arrest

data included all offenders with a unique person-id number. This uniqueness allowed us to track all offenders' activities. We had opportunity to find out an offender's history over time with all his/her crimes had committed. We produced first the link table, and then converted it to a massive graph; at the end all components in the graph are obtained with SCC. Accepting that even two offenders caught by the police is enough to be a component, total number of components were 33004 (199728 crimes; with an average of 6.05 crimes per component). When W_{group} threshold is put to 2, number of components is dropped to 4488 (15482 crimes; with an average of 3.45 crimes per group). When N_{group} threshold is put to 3, number of offender groups, which is adherent to Turkish Criminal Law definition, is dropped to 1416. Reminding the fact that these groups included many offenders committed various types of crimes from theft to violence, from gangs to terrorists; we only focused on active theft groups who committed crimes in the last five years. As a result, 63 theft groups are detected and these findings were introduced to the police experts for further examination. According to police experts, our findings were very valuable but not enough. There was a consensus to search group members, gather enough evidence for arrest and prepare the case for a sentence. Besides, in parallel, the effectiveness of our method was also a question for the police so just one random theft group out of 63 is focused, a judge verdict is obtained for electronic surveillance and telephone conversations of all members of selected group are eavesdropped for ten weeks. Our findings for this theft group are exhibited in figure 4 as offenders by person-id numbers, and with degrees of members in brackets. Degree is a metric in social network analysis which is count of incoming and outgoing links for an actor (Wasserman et al., 1994). High degree value for an actor suggests that actor is likely to be a key player in the network.

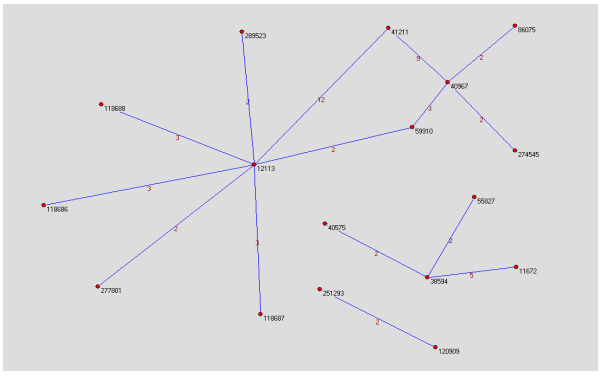


Figure 4. As filtering is applied in order to meet Turkish Criminal Code requirement, the theft group, consisting of 17 offenders with degrees in brackets:12113**(54)**, 41211**(42)**,40967**(32)**, 38594**(18)**, 11672**(10)**,59910**(10)**, 118686**(6)**,118687**(6)**,118688**(6)**,40575**(4)**,55827**(4)**,86075**(4)**, 120909**(4)**,251293**(4)**,274545**(4)**,277801**(4)**, 289523**(4)**

After this electronic surveillance, verification of who is who in the network and gathering enough convincing evidence, *Operation Cash* is launched. The police arrested 17 people, recovered worth US $ 200000 stolen jewelleries, PCs, laptops, mobile phones, and some cash worth US $ 180000.

Obtained evidences and interrogations showed that ruling members were detected using OGDM. It has been proved that the real network was consisting of 21 members and 3 of them (AB, MRK, and SE) have never been arrested by the police so their names were not available in the database. We managed to get only 4 ruling members (12113, 38594, 41211, and 277801). Four leaders were basically the chief of gun-jewellers thieves (12113), the skilled expert thief specialized in electronic goods (277801), chief of electronic goods thieves (38594), chief of car and gadget supplier for the network (41211). Interestingly, "big brother" of the network (220868) has only two records in police database. His leader position is identified after interrogations and cross examination of members' statements.

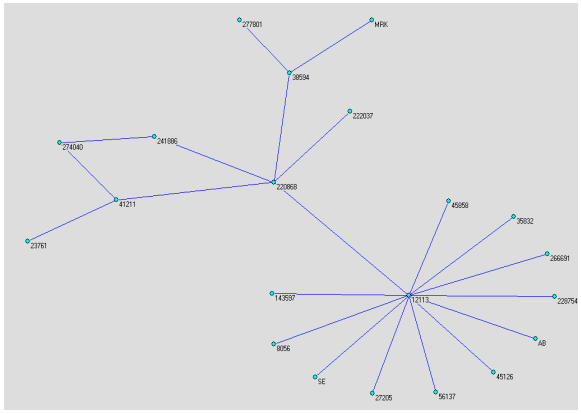


Figure 5. Theft network after verification of evidences. 12113**(24)**,38594**(6)**,41211**(6)**,241886**(4)**,274040**(4)**, 8056**(2)**, 23761**(2)**, 27205**(2)**, 35832**(2)**,45126**(2)**,45858**(2)**,56137**(2)**, 143597**(2)**,220868**(2)**, 222037**(2)**, 228754**(2)**, 266691**(2)**, 277801**(2)**, AB**(2)**, MRK**(2)**, SE**(2)**

Operation Cash has attracted wide attention and positive feedback in local and national newspapers (Zaman, Olay, PolisHaber, 2006). The police commissioner of Bursa city stated that *Operation Cash* was the most successful operation among all operations by Bursa Police in 2006.

9 Conclusion

It has been shown that co-defendant information in police arrest data is beneficial for the police to detect ruling members of offender groups. It has been also shown that detecting an underlying criminal network is possible with link mining and group detection techniques.

OGDM has been successful for partly detection of offender groups. But it is clear that domain expertise is still needed for complete detection of groups. This shows the necessity of semi-supervised models for OGD.

The result achieved depends on the details of the OGDM come from offender group representation success (see section 5). By representing actors as nodes and rest of the relations as edges in one-mode (friendship) social networks can produce many link types such as "co-defendant link", "spatial link", "same weapon link", "same modus operandi link". This helped many graph theoretical and SNA solutions can be used in *Operation Cash*.

Additional criminological conclusions reached after discussions with domain experts are;

- Group members likely to come from the same family (e.g. small-aged pickpocketing group).
- Group members likely to cooperate and come together for required skills to commit crimes.(e.g. theft from offices group, theft from residences group, fraud group, violence group).
- Group members are high likely coming from the same age group and peer group.
- Group members' origins are high likely coming from the same home cities and towns.
- Group members are likely to live in the same areas.
- Group members are likely to operate in the same areas.
- Group members are likely to work in the same industries(e.g. Scrap Dealer Auto theft Group).

References

Adderley, R. (2004), The use of data mining techniques in operational crime fighting *in* 2nd Symposium on Intelligence and Security Informatics, ISI- 2004. Tucson, AZ, USA. 3073, pp. 418–425.

Adibi, J., P. Pantel, et al. (2005), 'Report Link Discovery: Issues, Approaches and Applications', (KDD-2005 Workshop - LinkKDD-2005), *SIGKDD Explorations* **7**(2), pp. 123-125.

Adibi, J. & Chalupsky, H. (2004), The KOJAK Group Finder: Connecting the dots via integrated knowledge based and statistical reasoning, *in* IAAI.

Analyst Notebook (2007), 'i2 Analyst Notebook', i2 Ltd, <http://www.i2.co.uk/> Viewed at 31 July 2007.

Chen, H., Chung, W., et al. (2004), Crime data mining: a general framework and some examples, *in* Computer **37**(4), pp. 50-56.

Chen, H., J. Schroeder, et al. (2002), 'COPLINK Connect: information and knowledge management for law Enforcement', *Decision Support Systems* **34**, pp. 271-285.

Cook, D.J. & Holder, L.B. (2000), 'Graph-Based data mining', *IEEE Intelligent Systems* **15**(2), pp. 32-41

Cook, D.J. & Holder, L.B. (2007), *Graph Mining*, Wiley-Interscience, John Wiley Sons, Hoboken, New Jersey.

Cormen, T. H., Leiserson, C. E., Rivest, R. L. & Stein, C. (2001), *Introduction to Algorithms*. Second Edition. MIT Press and McGraw-Hill

Getoor, L. & Diehl, C.P. (2005), 'Link Mining: A Survey', *SIGKDD Explorations* **7**(2), pp. 3–12

Getoor, L. et al. (2004), 'Link Mining: a new data mining challenge', *SIGKDD Explorations* **5**(1), pp. 84-89.

Guest, S. D., Moody, J., Kelly, L., Rulison, K.L., (2007), 'Density or Distinction? The Roles of Data Structure and Group Detection Methods in Describing Adolescent Peer Groups', *Journal of Social Structure*, **8**(1), Viewed at 28 July 2007,< http://www.cmu.edu/joss/content/articles/volindex.html >.

Kubica, J., Moore, A., et al. (2003), cGraph: A fast graph-based method for link analysis and queries, *in* IJCAI 2003 Text Mining and Link Analysis Workshop.

Kubica, J., Moore, A., et al. (2002), Stochastic Link and Group Detection, *in* 18th National Conference on Artificial Intelligence, AAAI Press/ MIT Press

Marcus, S.M., Moy, M. & Coffman, T. (2007), Social Network Analysis, *in* Diane J.Cook and Lawrence B. Holder, 'Mining Graph Data', John Wiley & Sons.

Moy, M. (2005), 'Using TMODS to run the best friends group detection algorithm', 21st Century Technologies Internal Publication.

Olay (2006), 'Technological tracking to criminal groups', Bursa Olay Local Newspaper, 19th of December 2006 , Viewed at 31 July 2007, <http://www2.olay.com.tr/blocks/haberoku.php?id=5990&cins=Spot%20Bursa>.

PolisHaber (2006), 'Operation 'Cash' By Police', Turkish Police News Portal, Viewed at 31 July 2007, <http://www.polis.web.tr/article_view.php?aid=3666>.

Scott, J. (2004), *Social Network Analysis: A Handbook*, SAGE Publications, London, UK.

Senator, T.E. (2005), 'Link Mining Applications: Progress and Challenges', *SIGKDD Explorations*, **7**(2), pp. 76–83.

Sentient (2007), 'Sentient Data Detective', Sentient Information Systems, Viewed at 31 July 2007, <http://www.www.sentient.nl/>.

Taipale, K. A. (2003), 'Data Mining and Domestic Security: Connecting the Dots to Make Sense of Data', Columbia Science and Technology Law Review 5.

Wasserman, S. & Faust, K. (1994), *Social Network Analysis Methods and Applications.* Structural Analysis in the Social Sciences, Cambridge University Press.

Xu, J. J. & Chen, H. (2005), 'CrimeNet Explorer: A Framework for Criminal Network Knowledge Discovery', *ACM Transactions on Information Systems* **23**(2), pp. 201-226.

Zaman (2006), 'Police tracked down 63 crime groups with new technology help', Zaman National Newspaper, 9th of January 2007, Viewed at 31 July 2007, <http://www.zaman.com.tr/webapp-tr/haber.do?haberno=437444>.

Preference Networks: Probabilistic Models for Recommendation Systems

Tran The Truyen †, **Dinh Q. Phung** ‡ **and Svetha Venkatesh** ‡

Department of Computing, Curtin University of Technology
GPO Box U 1987, Perth, WA, Australia
† thetruyen.tran@postgrad.curtin.edu.au, ‡ {d.phung,s.venkatesh}@curtin.edu.au

Abstract

Recommender systems are important to help users select relevant and personalised information over massive amounts of data available. We propose an unified framework called Preference Network (PN) that jointly models various types of domain knowledge for the task of recommendation. The PN is a probabilistic model that systematically combines both content-based filtering and collaborative filtering into a single conditional Markov random field. Once estimated, it serves as a probabilistic database that supports various useful queries such as rating prediction and top-N recommendation. To handle the challenging problem of learning large networks of users and items, we employ a simple but effective pseudo-likelihood with regularisation. Experiments on the movie rating data demonstrate the merits of the PN.

Keywords: Hybrid Recommender Systems, Collaborative Filtering, Preference Networks, Conditional Markov Networks, Movie Rating.

1 Introduction

With the explosive growth of the Internet, users are currently overloaded by massive amount of media, data and services. Thus selective delivery that matches personal needs is very critical. Automated recommender systems have been designed for this purpose, and they are deployed in major online stores such as Amazon [http://www.amazon.com], Netflix [http://www.netfix.com] and new services such as Google News [http://news.google.com].

Two most common tasks in recommender systems are predicting the score the user might give for a product (the *rating prediction task*), and recommending a ranked list of most relevant items (the *top-N recommendation task*). The recommendations are made on the basis of the content of products and services (*content-based*), or based on collective preferences of the crowd (*collaborative filtering*), or both (*hybrid methods*). Typically, content-based methods work by matching product attributes to user-profiles using classification techniques. Collaborative filtering, on the other hand, relies on preferences over a set products that a given user and others have expressed. From the preferences, typically in term of numerical ratings, *correlation-based methods* measure similarities between users (Resnick et al. 1994) (user-based methods) and products (Sarwar et al. 2001) (item-based methods). As content and preferences are complementary, hybrid methods often work best when both types of information is available (Balabanović & Shoham 1997, Basu et al. 1998, Pazzani 1999, Schein et al. 2002, Basilico & Hofmann 2004).

Probabilistic modeling (Breese et al. 1998, Heckerman et al. 2001, Hofmann 2004, Marlin 2004) has been applied to the recommendation problem to some degree and their success has been mixed. Generally, they build probabilistic models that explain data. Earlier methods include Bayesian networks and dependency networks (Breese et al. 1998, Heckerman et al. 2001) have yet to prove competitive against well-known correlation-based counterparts. The more recent work attempts to perform clustering. Some representative techniques are mixture models, probabilistic latent semantic analysis (pLSA) (Hofmann 2004) and latent Dirichlet allocation (LDA) (Marlin 2004). These methods are *generative* in the sense that it assumes some hidden process that generates observed data such as items, users and ratings. The generative assumption is often made for algorithmic convenience and but it does not necessarily reflect the true process of the real data.

Machine learning techniques (Billsus & Pazzani 1998, Basu et al. 1998, Basilico & Hofmann 2004) address the rating prediction directly without making the generative assumption. Rather, they map the recommendation into a classification problem that existing classifiers can solve (Basu et al. 1998, Zhang & Iyengar 2002). The map typically considers each user or each item as an independent problem, and ratings are training instances. However, the assumption that training instances are independently generated does not hold in collaborative filtering. Rather all the ratings are interconnected directly or indirectly through common users and items.

To sum up, it is desirable to build a recommendation system that can seamlessly integrate content and correlation information in a disciplined manner. At the same time, the system should address the prediction and recommendation tasks directly without replying on strong prior assumptions such as generative process and independence. To that end, we propose a probabilistic graphical formulation called Preference Network (PN) that has these desirable properties. The PN is a graph whose vertexes represent ratings (or preferences) and edges represent dependencies between ratings. The networked ratings are treated as random variables of conditional Markov random fields (Lafferty et al. 2001). Thus the PN is a formal and expressive formulation that supports learning from existing data and various inference tasks to make future prediction and recommendation. The probabilistic dependencies between ratings capture the correlations between co-rating users (as used in (Resnick et al. 1994)) and between co-rated items (as used in (Sarwar et al. 2001)).

Different from previous probabilistic models, the PN does not make any generative assumption. Rather, prediction of preferences is addressed directly based on the content and prior ratings available in the database. It also avoids the independence assumption made in the standard machine learning approach by supporting *collective clas-*

sification of preferences. The nature of graphical modeling enables PN to support missing ratings and joint predictions for a set of items and users. It provides some measure of *confidence* in each prediction made, making it easy to assess the nature of recommendation and rank results. More importantly, our experiments show that the PNs are competitive against the well-known user-based method (Resnick et al. 1994) and the item-based method (Sarwar et al. 2001).

2 Recommender Systems

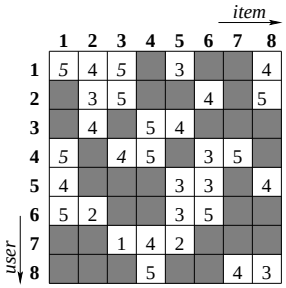
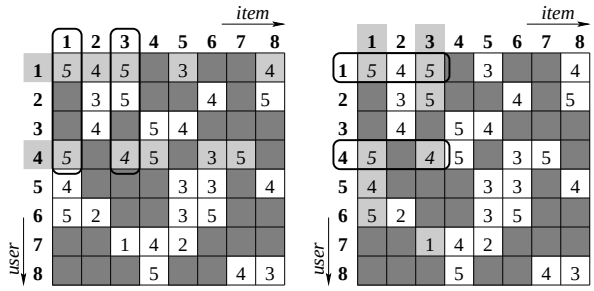


Figure 1: Preference matrix. Entries are numerical rating (or preference) and empty cells are to be filled by the recommender system.



(a) (b)

Figure 2: User-based correlation (a) and Item-based correlation (b).

This section provides some background on recommender systems and we refer readers to (Adomavicius & Tuzhilin 2005) for a more comprehensive survey. Let us start with some notations. Let $\mathcal{U} = \{u_1, \ldots, u_M\}$ be the set of M users (e.g. service subscribers, movie viewers, Website visitors or product buyers), and $\mathcal{I} = \{i_1, \ldots, i_L\}$ be the set of L products or items (e.g. services, movies, Webpages or books) that the user can select from. Let us further denote $\mathcal{M} = \{r_{ui}\}$ the *preference matrix* where u is the user index, i is the item index, and r_{ui} is the preference or the numerical rating of user u over item i (see Figure 1 for an illustration). In this paper, we assume that ratings have been appropriately transformed into integers, i.e. $r_{ui} \in \{1, 2, ..., S\}$.

Typically, a user usually rates only a small number of items and thus making the preference matrix $\mathcal{M}$ extremely sparse. For example, in the MovieLens dataset that we use in our experiments (Section 4), only about 6.3% entries in the $\mathcal{M}$ matrix are filled, and in large e-commerce sites, the sparsity can be as small as 0.001%. The rating prediction task in recommender systems can be considered as filling the empty cells in the preference matrix. Of course, due to the data sparsity, filling all the cells is impractical and often unnecessary because each user will be interested in a very small set of items. Rather, it is only appropriate for a limited set of entries in each row (corresponding to a user). Identifying the most relevant entries and ranking them are the goal of top-N recommendation.

Recommender techniques often fall into three groups: *content-based*, *collaborative filtering*, and *hybrid methods* that combines the former two groups.

Content-based methods rely on the content of items that match a user's profile to make recommendation using some classification techniques (e.g. see (Mooney & Roy 2000)). The content of an item is often referred to the set of attributes that characterise it. For example, in movie recommendation, item attributes include movie genres, release date, leading actor/actress, director, ratings by critics, financial aspects, movie description and reviews. Similarly, user attributes include static information such as age[1], sex, location, language, occupation and marriage status and dynamic information such as watching time (day/night/late night), context of use (e.g. home/theater/family/dating/group/company), and in case of on-demand videos, what other TV channels are showing, what the person has been watching in the past hours, days or weeks.

Collaborative filtering takes a different approach in that recommendation is based not only on the usage history of the user but also on experience and wisdom of related people in the user-item network. Most existing algorithms taking some measure of correlation between co-rating users or co-rated items. One family, known as user-based (sometimes memory-based) methods (Resnick et al. 1994), predicts a new rating of an item based on existing ratings on the same item by other users:

$$r_{ui} = \bar{r}_u + \frac{\sum_{v \in U(i)} s(u,v)(r_{ui} - \bar{r}_v)}{\sum_{v \in U(i)} |s(u,v)|}$$

where $s(u, v)$ is the similarity between user u and user v, $U(i)$ is the set of all users who rate item i, and $\bar{r}_u$ is the average rating by user u. The similarity $s(u, v)$ is typically measured using Pearson's correlation:

$$\frac{\sum_{i \in I(u,v)} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\left[\sum_{i \in I(u,v)} (r_{ui} - \bar{r}_u)^2\right]^{\frac{1}{2}} \left[\sum_{j \in I(u,v)} (r_{vj} - \bar{r}_v)^2\right]^{\frac{1}{2}}}$$

where $I(u, v)$ is the set of all items co-rated by users u and v. See Figure 2a for illustration. This similarity is computed offline for every pair of users who co-rate at least one common item.

The main drawback of user-based methods is in its lack of efficiency at prediction time because each prediction require searching and summing over all users who rate the current item. The set of such users is often very large for popular items, sometimes including all users in the database. In contrast, each user typically rates only a very limited number of items. Item-based methods (Sarwar et al. 2001) exploit that fact by simply exchanging the role of user and item in the user-based approach. Similarity between items $s(i, j)$ can be computed in several ways including the (adjusted) cosine between two item vectors, and the Pearson correlation. For example, the adjusted cosine similarity is computed as

$$\frac{\sum_{u \in U(i,j)} (r_{ui} - \bar{r}_u)(r_{uj} - \bar{r}_u)}{\left[\sum_{u \in U(i,j)} (r_{ui} - \bar{r}_u)^2\right]^{\frac{1}{2}} \left[\sum_{v \in U(i,j)} (r_{vj} - \bar{r}_v)^2\right]^{\frac{1}{2}}}$$

where $U(i, j)$ is the set of all users who co-rate both items i and j. See Figure 2b for illustration. The new rating is

[1] Strictly speaking, age is not truly static, but it changes really slowly as long as selling is concerned.

predicted as

$$r_{ui} = \bar{r}_i + \frac{\sum_{j \in I(u)} s(i,j)(r_{uj} - \bar{r}_j)}{\sum_{j \in I(u)} |s(i,j)|}$$

where $I(u)$ is the set of items rated by user u.

Many other methods attempt to build a model of training data that then use the model to perform prediction on unseen data. One class of methods employ probabilistic graphical models such as Bayesian networks (Breese et al. 1998), dependency networks (Heckerman et al. 2001), and restricted Boltzmann machines (Salakhutdinov et al. 2007). Our proposed method using Markov networks fall under the category of undirected graphical models. It resembles dependency networks in the way that *pseudo-likelihood* (Besag 1974) learning is employed, but dependency networks are generally inconsistent probabilistic models. In (Salakhutdinov et al. 2007), the authors build a generative Boltzmann machine for each user with hidden variables, while our method constructs a single discriminative Markov network for the whole database of all ratings.

Much of other probabilistic work attempts to perform clustering. This is an important technique for reducing the dimensionality and noise, dealing with data sparsity and more significantly, discovering latent structures. Here the latent structures are either communities of users with similar tastes or categories of items with similar features. Some representative techniques are mixture models, probabilistic latent semantic analysis (pLSA) (Hofmann 2004) and latent Dirichlet allocation (LDA) (Marlin 2004). These methods try to uncover some hidden process which is assumed to generate items, users and ratings. In our approach, no such generation is assumed and ratings are modeled conditionally given items and users and prior knowledge.

Statistical machine learning techniques (Billsus & Pazzani 1998, Basu et al. 1998, Zhang & Iyengar 2002, Basilico & Hofmann 2004, Zitnick & Kanade 2004) have also been used to some extent. One of the key observations made is that there is some similarity between text classification and rating prediction (Zhang & Iyengar 2002). However, the main difficulty is that the features in collaborative filtering are not rich and the nature of prediction is different. There are two ways to convert collaborative filtering into a classification problem (Billsus & Pazzani 1998). The first is to build a model for each item, and ratings by different users are treated as training instances. The other builds a model for each user, and ratings on different items by this user are considered as training instances (Breese et al. 1998). These treatments, however, are complementary, and thus, there should be a better way to systematically unify them (Basu et al. 1998, Basilico & Hofmann 2004). That is, the pairs (user,item) are now as independent training instances. Our approach, on the other hand, considers the pair as just a node in the network, thus relaxing the independence assumption.

Hybrid methods exploit the fact that content-based and collaborative filtering methods are complementary (Balabanović & Shoham 1997, Basu et al. 1998, Pazzani 1999, Schein et al. 2002, Basilico & Hofmann 2004). For example, the content-based methods do not suffer from the so-called *cold-start* problem (Schein et al. 2002) in standard collaborative filtering. The situation is when new user and new item are introduced to the database, as no previous ratings are available, purely correlation-based methods cannot work. On the other hand, content information available is sometimes very limited to basic attributes that are shared by many items or users. Prediction by pure content-based methods in that case cannot be personalised and may be inaccurate. Some work approaches the problem by making independent predictions separately using a content-based method and a collaborative filtering method and then combining the results (Claypool et al. 1999). Others (e.g. (Basilico & Hofmann 2004)) create joint representation of content and collaborative features. We follow the latter approach.

3 Preference Networks for Hybrid Recommendation

3.1 Model Description

Let us start with the preference matrix $\mathcal{M} = \{r_{ui}\}$ discussed previously (cf. Sec. 2), where we treat each entry r_{ui} in $\mathcal{M}$ as a random variable, and thus ideally we would be interested in a single joint model over KM variables for both the learning phase and the prediction/recommendation phase. However, in practice, KM is extremely large (e.g., $10^6 \times 10^6$) making computation intractable. In addition, such a modeling is unnecessary, because, as we have mentioned earlier in Section 2, a user is often interested in a moderate number of items. As a result, we adopt a two-step strategy. During the learning phase, we limit to model the joint distribution over existing ratings. And then during the prediction/recommendation phase, we extend the model to incorporate to-be-predicted entries.

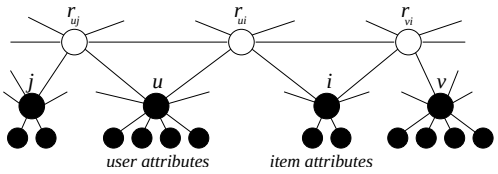


Figure 3: A fragment of the Preference Network.

We build the model by first representing the ratings and their relations using an undirected graph and then defining a joint distribution over the graph. Denote by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ an undirected graph that has a set of vertexes $\mathcal{V}$ and a set of edges $\mathcal{E}$. Each vertex in $\mathcal{V}$ in this case represents a rating r_{ui} of user u over item i and each edge in $\mathcal{E}$ capture a relation between two ratings. The set $\mathcal{E}$ defines a topological structure for the network, and specify how ratings are related.

We define the edges as follows. There is an edge between any two ratings by the same user, and an edge between two ratings on the same item. As a result, a vertex of r_{ui} will be connected with $U(i) + I(u) - 2$ other vertices. Thus, for each user, there is a fully connected subnetwork of all ratings she has made, plus connections to ratings by other users on these items. Likewise, for each item, there is a fully connected subnetwork of all ratings by different users on this item, plus connections to ratings on other items by these users. The resulting network $\mathcal{G}$ is typically very densely connected because $U(i)$ can be potentially very large (e.g. 10^6).

Let us now specify the probabilistic modeling of the ratings and their relations that respect the graph $\mathcal{G}$. Denote $t = (u, i)$ and let $\mathcal{T} = \{t\}$ be the set of a pair index (user, item), which corresponds to entries used in each phase. For notation convenience let $X = \{r_{ui} \mid (u,i) \in \mathcal{T}\}$ denote the joint set of all variables, and the term 'preference' and 'rating' will be used interchangeably. When there is no confusion, we use r_u to denote ratings related to user u and r_i denotes ratings related to item i.

In our approach to the hybrid recommendation task, we consider attributes of items $\{\mathbf{a}_i\}_{i=1}^{L}$, and attributes of users $\{\mathbf{a}_u\}_{i=u}^{M}$. Let $\mathbf{o} = \{\{\mathbf{a}_i\}_{i=1}^{L}, \{\mathbf{a}_u\}_{i=u}^{M}\}$, we are interested in modeling the conditional distribution $P(X|\mathbf{o})$ of all user ratings X given $\mathbf{o}$. We employ the conditional Markov random field (Lafferty et al. 2001) as the underlying inference machinery. As X collectively represents users' preferences, we refer this model as *Preference Network*.

Preference Network (PN) is thus a conditional Markov random field that defines a distribution $P(X|\mathbf{o})$ over the graph $\mathcal{G}$:

$$P(X|\mathbf{o}) = \frac{1}{Z(\mathbf{o})}\Psi(X,\mathbf{o}), \quad \text{where}$$
$$\Psi(X,\mathbf{o}) = \prod_{t\in\mathcal{V}}\psi_t(r_t,\mathbf{o}) \prod_{(t,t')\in\mathcal{E}}\psi_{t,t'}(r_t,r_{t'},\mathbf{o}) \quad (1)$$

where $Z(\mathbf{o})$ is the normalisation constant to ensure that $\sum_X P(X|\mathbf{o}) = 1$, and $\psi(.)$ is a positive function, often known as *potential*. More specifically, $\psi_t(r_t,\mathbf{o})$ encodes the content information associated with the rating r_t including the attributes of the user and the item. On the other hand, $\psi_{t,t'}(r_t,r_{t'},\mathbf{o})$ captures the correlations between two ratings r_t and $r_{t'}$. Essentially, when there are no correlation potentials, the model is purely content-based, and when there are no content potentials, the model is purely collaborative-filtering. Thus the PN integrates both types of recommendation in a seamlessly unified framework.

The contribution of content and correlation potentials to the joint distribution will be adjusted by weighting parameters associated with them. Specifically, the parameters are encoded in potentials as follows

$$\psi_t(r_t,\mathbf{o}) = \exp\left\{\mathbf{w}_v^\top \mathbf{f}_v(r_t,\mathbf{o})\right\} \quad (2)$$
$$\psi_{t,t'}(r_t,r_{t'},\mathbf{o}) = \exp\left\{\mathbf{w}_e^\top \mathbf{f}_e(r_t,r_{t'},\mathbf{o})\right\} \quad (3)$$

where $\mathbf{f}(.)$ is the feature vector and $\mathbf{w}$ is the corresponding weight vector. Thus together with their weights, the features realise the contribution of the content and the strength of correlations between items and users. The design of features will be elaborated further in Section 3.2. Parameter estimation is described in Section 3.3.

3.2 Feature Design and Selection

Corresponding to the potentials in Equations 2 and 3, there are attribute-based features and correlation-based features. Attribute-based features include user/item identities and contents.

Identity Features. Assume that the ratings are integer, ranging from 1 to S. We know from the database the average rating $\bar{r}_i$ of item i which roughly indicates the general quality of the item with respect to those who have rated it. Similarly, the average rating $\bar{r}_u$ by user u over items she has rated roughly indicates the user-specific scale of the rating because the same rating of 4 may mean 'OK' for a regular user, but may mean 'excellent' for a critic. We use two features *item-specific* $f_i(r_{ui},i)$ and *user-specific* $f_u(r_{ui},u)$:

$$f_i(r_{ui},i) = g(|r_{ui}-\bar{r}_i|), \quad f_u(r_{ui},u) = g(|r_{ui}-\bar{r}_u|)$$

where $g(\alpha) = 1-\alpha/(S-1)$ is used to ensure that the feature values is normalized to $[0,1]$, and when α plays the role of rating deviation, $g(\alpha) = 1$ for $\alpha = 0$.

Content Features. For each rating by user u on item i, we have a set of item attributes $\mathbf{a}_i$ and set of user attributes $\mathbf{a}_u$. Mapping from item attributes to user preference can be carried out through the following feature

$$\mathbf{f}_u(r_{ui}) = \mathbf{a}_i g(|r_{ui}-\bar{r}_u|)$$

Similarly, we are also interested in seeing the classes of users who like a given item through the following mapping

$$\mathbf{f}_i(r_{ui}) = \mathbf{a}_u g(|r_{ui}-\bar{r}_i|)$$

Correlation Features. We design two features to capture correlations between items or users. Specifically, the *item-item* $f_{i,j}(\cdot)$ features capture the fact that if a user rates two items then after offsetting the goodness of each item, the ratings may be similar

$$f_{i,j}(r_{ui},r_{uj}) = g(|(r_{ui}-\bar{r}_i)-(r_{uj}-\bar{r}_j)|)$$

Likewise, the *user-user* $f_{u,v}(\cdot)$ features capture the idea that if two users rate the same item then the ratings, after offset by user's own scale, should be similar:

$$f_{u,v}(r_{ui},r_{vi}) = g(|(r_{ui}-\bar{r}_u)-(r_{vi}-\bar{r}_v)|)$$

Since the number of correlation features can be large, making model estimation less robust, we select only item-item features with positive correlation (given in Equation 1), and user-user features with positive correlations (given in Equation 1).

3.3 Parameter Estimation

Since the network is densely connected, learning methods based on the standard log-likelihood $\log P(X|\mathbf{o})$ are not applicable. This is because underlying inference for computing the log-likelihood and its gradient is only tractable for simple networks with simple chain or tree structures (Pearl 1988). As a result, we resort to the simple but effective *pseudo-likelihood* learning method (Besag 1974). Specifically, we replace the log likelihood by the regularised sum of log local likelihoods

$$\mathcal{L}(\mathbf{w}) = \sum_{(u,i)\in\mathcal{T}} \log P(r_{ui}|\mathcal{N}(u,i),\mathbf{o}) - \frac{1}{2}\bar{\mathbf{w}}^\top\bar{\mathbf{w}} \quad (4)$$

where, $\mathcal{N}(u,i)$ is the set of neighbour ratings that are connected to r_{ui}. As we mentioned earlier, the size of the neighbourhood is $|\mathcal{N}(u,i)| = U(i)+I(u)-2$. In the second term in the RHS, $\bar{\mathbf{w}} = \mathbf{w}/\boldsymbol{\sigma}$ (element-wise division, regularised by a prior diagonal Gaussian of mean 0 and standard deviation vector $\boldsymbol{\sigma}$).

Finally, the parameters are estimated by maximising the pseudo-likelihood

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} \mathcal{L}(\mathbf{w}) \quad (5)$$

Not only is this regularised pseudo-likelihood simple to implement, it makes sense since the local conditional distribution $P(r_{ui}|\mathcal{N}(u,i),\mathbf{o})$ is used in prediction (Equation 7). We limit ourselves to supervised learning in that all the ratings $\{r_{ui}\}$ in the training data are known. Thus, $\mathcal{L}(\mathbf{w})$ is a concave function of $\mathbf{w}$, and thus has a unique maximum.

To optimise the parameters, we use a simple stochastic gradient ascent procedure that updates the parameters after passing through a set of ratings by each user:

$$\mathbf{w}_u \leftarrow \mathbf{w}_u + \lambda\nabla\mathcal{L}(\mathbf{w}_u) \quad (6)$$

where $\mathbf{w}_u$ is the subset of parameters that are associated with ratings by user u, and $\lambda > 0$ is the learning rate. Typically, 2-3 passes through the entire data are often enough in our experiments. Further details of the computation are included in Appendix A.

3.4 Preference Prediction

Recall from Section 3.1 that we employ a two-step modeling. In the learning phase (Section 3.3), the model includes all previous ratings. Once the model has been estimated, we extend the graph structure to include new ratings that need to be predicted or recommended. Since the number of ratings newly added is typically small compared to the size of existing ratings, it can be assumed that the model parameters do not change.

The prediction of the rating r_{ui} for user u over item i is given as

$$\hat{r}_{ui} = \arg\max_{r_{ui}} P(r_{ui} \mid \mathcal{N}(u,i), \mathbf{o}) \quad (7)$$

The probability $P(\hat{r}_{ui}|\mathcal{N}(r_{ui}), \mathbf{o})$ is the measure of the *confidence* or ranking level in making this prediction. This can be useful in practical situations when we need high precision, that is, only ratings with the confidence above a certain threshold are presented to the users.

We can jointly infer the ratings r_u of given user u on a subset of items $\mathbf{i} = (i_1, i_2, ..)$ as follows

$$\hat{r}_u = \arg\max_{r_u} P(r_u \mid \mathcal{N}(u), \mathbf{o}) \quad (8)$$

where $\mathcal{N}(u)$ is the set of all existing ratings that share the common cliques with ratings by user u. In another scenario, we may want to recommend a relatively new item i to a set of promising users, we can make joint predictions r_i as follows

$$\hat{r}_i = \arg\max_{r_i} P(r_i \mid \mathcal{N}(i), \mathbf{o}) \quad (9)$$

where $\mathcal{N}(i)$ is the set of all existing ratings that share the common cliques with ratings of item i. It may appear non-obvious that a prediction may depend on unknown ratings (other predictions to be made) but this is the advantage of the Markov networks. However, joint predictions for a user are only possible if the subset of items is small (e.g. less than 20) because we have a completely connected subnetwork for this user. This is even worse for joint prediction of an item because the target set of users is usually very large.

3.5 Top-N recommendation

In order to provide a list of top-N items to a given user, the first step is usually to identify a candidate set of C promising items, where $C \geq N$. Then in the second step, we rank and choose the best N items from this candidate set according to some measure of relevance.

Identifying the candidate set.
This step should be as efficient as possible and C should be relatively small compared to the number of items in the database. There are two common techniques used in user-based and item-based methods, respectively. In the user-based technique, first we identify a set of K most similar users, and then take the union of all items co-rated by these K users. Then we remove items that the user has previously rated. In the item-based technique (Deshpande & Karypis 2004), for each item the user has rated, we select the K best similar items that the user has not rated. Then we take the union of all of these similar items.

Indeed, if $K \to \infty$, or equivalently, we use all similar users and items in the database, then the item sets returned by the item-based and user-based techniques are *identical*. To see why, we show that every candidate j returned by the item-based technique is also the candidate by the user-based technique, and vice versa. Recall that a pair of items is said to be similar if they are jointly rated by the same user. Let $I(u)$ be the set of items rated by the current user u. So for each item $j \notin I(u)$ similar to item $i \in I(u)$, there must exist a user $v \neq u$ so that $i, j \in I(v)$. Since u and v jointly rate i, they are similar users, which mean that j is also in the candidate set of the user-based method. Analogously, for each candidate j rated by user v, who is similar to u, and $j \notin I(u)$, there must be an item $i \neq j$ jointly rated by both u and v. Thus $i, j \in I(v)$, and therefore they are similar. This means that j must be a candidate by the item-based technique.

In our Preference Networks, the similarity measure is replaced by the correlation between users or between items. The correlation is in turn captured by the corresponding correlation parameters. Thus, we can use either the user-user correlation or item-item correlation to identify the candidate set. Furthermore, we can also use both the correlation types and take the union of the two candidate sets.

Ranking the candidate set.
The second step in the top-N recommendation is to rank these C candidates according to some scoring methods. Ranking in the user-based methods is often based on item popularity, i.e. the number of users in the neighbourhood who have rated the item. Ranking in the item-based methods (Deshpande & Karypis 2004) is computed by considering not only the number of raters but the similarity between the items being ranked and the set of items already rated by the user.

Under our Preference Networks formulation, we propose to compute the change in system energy and use it as ranking measure. Our PN can be thought as a stochastic physical system whose energy is related to the conditional distribution as follows

$$P(X|\mathbf{o}) = \frac{1}{Z(\mathbf{o})} \exp(-E(X, \mathbf{o})) \quad (10)$$

where $E(X, \mathbf{o}) = -\log \Psi(X, \mathbf{o})$ is the system energy. Thus the lower energy the system state X has, the more probable the system is in that state. Let $t = (u, i)$, from Equations 2 and 3, we can see that the system energy is the sum of node-based energy and interaction energy

$$E(X, \mathbf{o}) = \sum_{t \in \mathcal{V}} E(r_t, \mathbf{o}) + \sum_{(t,t') \in \mathcal{E}} E(r_t, r_{t'} \mathbf{o})$$

where

$$E(r_t, \mathbf{o}) = -\mathbf{w}_v^\top \mathbf{f}_v(r_t, \mathbf{o}) \quad (11)$$

$$E(r_t, r_{t'}, \mathbf{o}) = -\mathbf{w}_e^\top \mathbf{f}_e(r_t, r_{t'}, \mathbf{o}) \quad (12)$$

Recommending a new item i to a given user u is equivalent to extending the system by adding new rating node r_{ui}. The change in system energy is therefore the sum of node-based energy of the new node, and the interaction energy between the node and its neighbours.

$$\Delta E(r_t, \mathbf{o}) = E(r_t, \mathbf{o}) + \sum_{t' \in \mathcal{N}(t)} E(r_t, r_{t'}, \mathbf{o})$$

For simplicity, we assume that the state of the existing system does not change after node addition. Typically, we want the extended system to be in the most probable state, or equivalently the system state with lowest energy. This means that the node that causes the most reduction of system energy will be preferred. Since we do not know the correct state r_t of the new node t, we may guess by predicting $\hat{r}_t$ using Equation 7. Let us call the energy reduction by this method the *maximal energy change*. Alternatively, we may compute the *expected energy change* to account for the uncertainty in the preference prediction

$$\mathbb{E}[\Delta E(r_t, \mathbf{o})] = \sum_{r_t} P(r_t|\mathcal{N}(t), \mathbf{o}) \Delta E(r_t, \mathbf{o}) \quad (13)$$

4 Experiments

In this section, we evaluate our Preference Network against well-established correlation methods on the movie recommendation tasks, which include rate prediction and top-N item recommendation.

4.1 Data and Experimental Setup

We use the MovieLens data[2], collected by the GroupLens Research Project at the University of Minnesota from September 19th, 1997 through April 22nd, 1998. We use the dataset of 100,000 ratings in the 1-5 scale. This has 943 users and 1682 movies. The data is divided into a training set of 80,000 ratings, and the test set of 20,000 ratings. The training data accounts for 852,848 and 411,546 user-based item-based correlation features.

We transform the content attributes into a vector of binary indicators. Some attributes such as sex are categorical and thus are dimensions in the vector. Age requires some segmentation into intervals: under 18, 18-24, 25-34, 35-44, 45-49, 50-55, and 56+. We limit user attributes to age, sex and 20 job categories [3], and item attributes to 19 film genres [4]. Much richer movie content can be obtained from the Internet Movie Database (IMDB)[5].

4.2 Accuracy of Rating Prediction

In the training phrase, we set the learning rate $\lambda = 0.001$ and the regularisation term $\sigma = 1$. We compare our method with well-known user-based (Resnick et al. 1994) and item-based (Sarwar et al. 2001) techniques (see Section 2). Two metrics are used: the mean absolute error (MAE)

$$\sum_{(u,i)\in\mathcal{T}'} |\hat{r}_{ui} - r_{ui}|/(|\mathcal{T}'|) \tag{14}$$

where $\mathcal{T}'$ is the set of rating indexes in the test data, and the mean 0/1 error

$$\sum_{(u,i)\in\mathcal{T}'} \delta(\hat{r}_{ui} \neq r_{ui})/(|\mathcal{T}'|) \tag{15}$$

In general, the MAE is more desirable than the 0/1 error because making exact prediction may not be required and making 'closed enough' predictions is still helpful. As item-based and user-used algorithms output real ratings, we round the numbers before computing the errors. Results shown in Figure 4 demonstrate that the PN outperforms both the item-based and user-based methods.

Sensitivity to Data Sparsity.
To evaluate methods against data sparsity, we randomly subsample the training set, but fix the test set. We report the performance of different methods using the MAE metric in Figure 5 and using the mean 0/1 errors in Figure 6. As expected, the purely content-based method deals with the sparsity in the user-item rating matrix very well, i.e. when the training data is limited. However, as the content we use here is limited to a basic set of attributes, more data does not help the content-based method further. The correlation-based method (purely collaborative filtering), on the other hand, suffers severely from the sparsity, but outperforms all other methods when the data is sufficient. Finally, the hybrid method, which combines all the content, identity and correlation features, improves the performance of all the component methods, both when data is sparse, and when it is sufficient.

4.3 Top-N Recommendation

We produce a ranked list of items for each user in the test set so that these items do not appear in the training set.

[2] http://www.grouplens.org

[3] Job list: administrator, artist, doctor, educator, engineer, entertainment, executive, healthcare, homemaker, lawyer, librarian, marketing, none, other, programmer, retired, salesman, scientist, student, technician, writer.

[4] Film genres: unknown, action, adventure, animation, children, comedy, crime, documentary, drama, fantasy, film-noir, horror, musical, mystery, romance, sci-fi, thriller, war, western.

[5] http://us.imdb.com

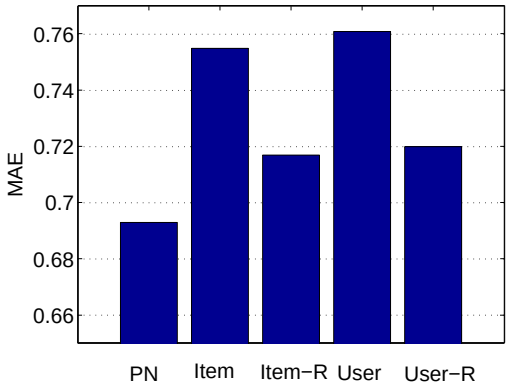


Figure 4: The mean absolute error of recommendation methods (Item: item-based method, Item-R: item-based method with rounding, User: user-based method, and User-R: user-based method with rounding).

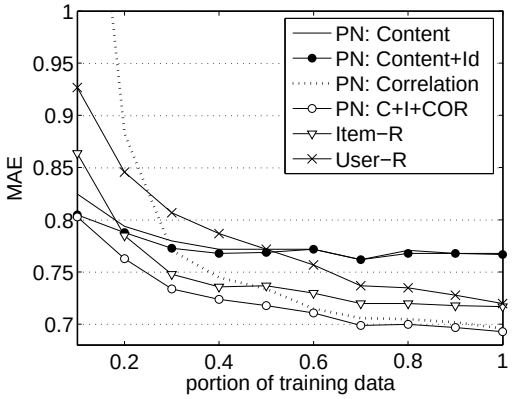


Figure 5: The mean absolute error (MAE) of recommendation methods with respect to training size of the MovieLens data. (Item-R: item-based method with rounding, User-R: user-based method with rounding, Content: PNs with content-based features, and C+I+CORR: PNs with content, identity and correlation features).

When a recommended item is in the test set of a user, we call it is a hit. For evaluation, we employ two measures. The first is the *expected utility* of the ranked list (Breese et al. 1998), and the second is the MAE computed over the hits. The expected utility takes into account of the position j of the hit in the list for each user u

$$R_u = \sum_j \frac{1}{2^{(j-1)/(\alpha-1)}} \tag{16}$$

where α is the viewing halflife. Following (Breese et al. 1998), we set $\alpha = 5$. Finally, the expected utility for all users in the test set is given as

$$R = 100\frac{\sum_u R_u}{\sum_u R_u^{max}} \tag{17}$$

where R_u^{max} is computed as

$$R_u^{max} = \sum_{j\in I'(u)} \frac{1}{2^{(j-1)/(\alpha-1)}} \tag{18}$$

where $I'(u)$ is the set of items of user u in the test set.

For comparison, we implement a user-based recommendation in that for each user, we choose 100 best (positively) correlated users and then rank the item based on the

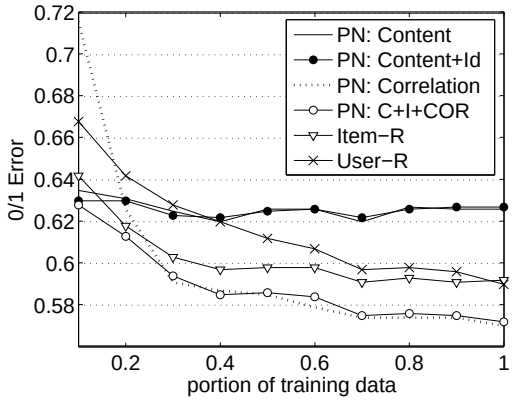


Figure 6: The mean 0/1 error of recommendation methods with respect to training size of the MovieLens data. (Item-R: item-based method with rounding, User-R: user-based method with rounding, Content: PNs with content-based features, and C+I+CORR: PNs with content, identity and correlation features).

number of times it is rated by them. Table 1 reports results of Preference Network with ranking measure of maximal energy change and expected energy change in producing the top 20 item recommendations.

Method	MAE	Expected Utility
User-based	0.669	46.61
PN (maximal energy)	0.603	47.43
PN (expected energy)	0.607	48.49

Table 1: Performance of top-20 recommendation. PN = Preference Network.

We vary the rate of recall by varying the value of N, i.e. the recall rate typically improves as N increases. We are interested in how the expected utility and the MAE changes as a function of recall. The expected energy change is used as the ranking criteria for the Preference Network. Figure 7 shows that the utility increases as a function of recall rate and reaches a saturation level at some point. Figure 8 exhibits a similar trend. It supports the argument that when the recall rate is smaller (i.e. N is small), we have more confidence on the recommendation. For both measures, it is evident that the Preference Network has an advantage over the user-based method.

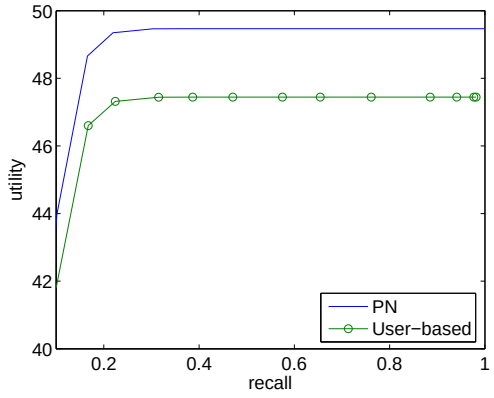


Figure 7: Expected utility as a function of recall. The larger utility, the better. PN = Preference Network.

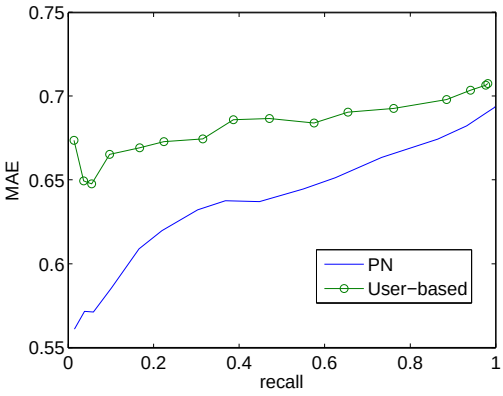


Figure 8: MAE as a function of recall. The smaller MAE, the better. PN = Preference Network.

5 Discussion and Conclusions

We have presented a novel hybrid recommendation framework called Preference Networks that integrates different sources of content (content-based filtering) and user's preferences (collaborative filtering) into a single network, combining advantages of both approaches, whilst overcoming shortcomings of individual approaches such as the cold-start problem of the collaborative filtering. Our framework, based on the conditional Markov random fields, are formal to characterise and amenable to inference. Our experiments show that PNs are competitive against both the well-known item-based and user-based collaborative filtering methods in the rating prediction task, and against the user-based method in the top-N recommendation task.

Once learned, the PN is a probabilistic database that allows interesting queries. For example, the set of most influential items for a particular demographic user group can be identified based on the corresponding energies. Moreover, the conditional nature of the PN supports fusion of varieties of information into the model through weighted feature functions. For example, the features can capture the assertion that if two people are friends, they are more likely to have similar tastes even though they have not explicitly provided any common preferences[6].

Finally, one main drawback the PNs inherit from the user-based methods is that it may be expensive at prediction time, because it takes into account all users who are related to the current one. On-going work will investigate clustering techniques to reduce the number of pair-wise connections between users.

References

Adomavicius, G. & Tuzhilin, A. (2005), 'Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions', *Knowledge and Data Engineering, IEEE Transactions on* **17**(6), 734–749.

Balabanović, M. & Shoham, Y. (1997), 'Fab: content-based, collaborative recommendation', *Communications of the ACM* **40**(3), 66–72.

Basilico, J. & Hofmann, T. (2004), 'Unifying collaborative and content-based filtering', *Proceedings of the twenty-first international conference on Machine learning* .

[6] Friends are a influential factor of consumer behaviour via the 'word-of-mouth' process.

Basu, C., Hirsh, H. & Cohen, W. (1998), 'Recommendation as classification: Using social and content-based information in recommendation', *Proceedings of the Fifteenth National Conference on Artificial Intelligence* .

Besag, J. (1974), 'Spatial interaction and the statistical analysis of lattice systems (with discussions)', *Journal of the Royal Statistical Society Series B* **36**, 192–236.

Billsus, D. & Pazzani, M. (1998), 'Learning collaborative information filters', *Proceedings of the Fifteenth International Conference on Machine Learning* pp. 46–54.

Breese, J., Heckerman, D., Kadie, C. et al. (1998), 'Empirical analysis of predictive algorithms for collaborative filtering', *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence* **461**.

Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D. & Sartin, M. (1999), 'Combining content-based and collaborative filters in an online newspaper', *ACM SIGIR Workshop on Recommender Systems* .

Deshpande, M. & Karypis, G. (2004), 'Item-based top-N recommendation algorithms', *ACM Transactions on Information Systems (TOIS)* **22**(1), 143–177.

Heckerman, D., Chickering, D., Meek, C., Rounthwaite, R. & Kadie, C. (2001), 'Dependency networks for inference, collaborative filtering, and data visualization', *The Journal of Machine Learning Research* **1**, 49–75.

Hofmann, T. (2004), 'Latent semantic models for collaborative filtering', *ACM Transactions on Information Systems (TOIS)* **22**(1), 89–115.

Lafferty, J., McCallum, A. & Pereira, F. (2001), Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data, *in* 'ICML', pp. 282–289.

Marlin, B. (2004), 'Modeling user rating profiles for collaborative filtering', *Advances in Neural Information Processing Systems* **16**, 627–634.

Mooney, R. & Roy, L. (2000), 'Content-based book recommending using learning for text categorization', *Proceedings of the fifth ACM conference on Digital libraries* pp. 195–204.

Pazzani, M. (1999), 'A Framework for Collaborative, Content-Based and Demographic Filtering', *Artificial Intelligence Review* **13**(5), 393–408.

Pearl, J. (1988), *Probabilistic reasoning in intelligent systems: networks of plausible inference*, Morgan Kaufmann, San Francisco, CA.

Resnick, P., Iacovou, N., Suchak, M., Bergstorm, P. & Riedl, J. (1994), GroupLens: An Open Architecture for Collaborative Filtering of Netnews, *in* 'Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work', ACM, Chapel Hill, North Carolina, pp. 175–186.

Salakhutdinov, R., Mnih, A. & Hinton, G. (2007), Restricted Boltzmann machines for collaborative filtering, *in* 'ICML'.

Sarwar, B., Karypis, G., Konstan, J. & Reidl, J. (2001), 'Item-based collaborative filtering recommendation algorithms', *Proceedings of the 10th international conference on World Wide Web* pp. 285–295.

Schein, A., Popescul, A., Ungar, L. & Pennock, D. (2002), 'Methods and metrics for cold-start recommendations', *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval* pp. 253–260.

Zhang, T. & Iyengar, V. (2002), 'Recommender systems using linear classifiers', *Journal of Machine Learning Research* **2**(3), 313–334.

Zitnick, C. & Kanade, T. (2004), 'Maximum entropy for collaborative filtering', *Proceedings of the 20th conference on Uncertainty in artificial intelligence* pp. 636–643.

A Markov Property and Learning Log-linear Models

This paper exploits an important aspect of Markov networks known as Markov property that greatly simplifies the computation. Basically, the property ensures the conditional independence of a variable r_t with respect to other variables in the network given its neighbourhood

$$P(r_t|x\backslash r_t, \mathbf{o}) = P(r_t|\mathcal{N}(t), \mathbf{o}) \tag{19}$$

where $\mathcal{N}(t)$ is the neighbourhood of r_t. This explains why we just need to include the neighbourhood in the Equation 7. This is important because $P(r_t|\mathcal{N}(t), \mathbf{o})$ can be easily evaluated

$$P(r_t|\mathcal{N}(t), \mathbf{o}) = \frac{1}{Z_t}\psi_t(r_t, \mathbf{o}) \prod_{t' \in \mathcal{N}(t)} \psi_{t,t'}(r_t, r_{t'}, \mathbf{o})$$

where $Z_t = \sum_{r_t} \psi_t(r_t, \mathbf{o}) \prod_{t' \in \mathcal{N}(t)} \psi_{t,t'}(r_t, r_{t'}, \mathbf{o})$.

The parameter update rule in Equation 6 requires the computation of the gradient of the regularised log pseudo-likelihood in Equation 4, and thus, the gradient of the log pseudo-likelihood $L = \log P(r_t|\mathcal{N}(t), \mathbf{o})$. Given the log-linear parameterisation in Equations 2 and 3, we have

$$\begin{aligned}
\frac{\partial \log L}{\partial \mathbf{w}_v} &= \mathbf{f}_v(r_t, \mathbf{o}) - \sum_{r'_t} P(r'_t|\mathcal{N}(t), \mathbf{o})\mathbf{f}_v(r'_t, \mathbf{o}) \\
\frac{\partial \log L}{\partial \mathbf{w}_e} &= \mathbf{f}_e(r_t, r'_t, \mathbf{o}) - \sum_{r'_t} P(r'_t|\mathcal{N}(t), \mathbf{o})\mathbf{f}_e(r'_t, r_{t'}, \mathbf{o})
\end{aligned}$$

Classification for accuracy and insight: A weighted sum approach

Anthony Quinn **Andrew Stranieri** **John Yearwood**

School of Information Technology and Mathematical Sciences
University of Ballarat,
Gear Ave, Ballarat, Victoria 3350,
Email: quinn@clearmail.com.au

Abstract

This research presents a classifier that aims to provide insight into a dataset in addition to achieving classification accuracies comparable to other algorithms. The classifier called, **A**utomated **W**eighted **Sum** (AWSum) uses a weighted sum approach where feature values are assigned weights that are summed and compared to a threshold in order to classify an example. Though naive, this approach is scalable, achieves accurate classifications on standard datasets and also provides a degree of insight. By insight we mean that the technique provides an appreciation of the influence a feature value has on class values, relative to each other. AWSum provides a focus on the feature value space that allows the technique to identify feature values and combinations of feature values that are sensitive and important for a classification. This is particularly useful in fields such as medicine where this sort of micro-focus and understanding is critical in classification.

Keywords: data mining, insight, conditional probability.

1 Introduction

Many classifiers provide a high level of classification accuracy, yet their use in real world problems is limited because they provide little insight into the data. The classifier presented in this research, **A**utomated **W**eighted **Sum** (AWSum), provides a degree of insight into the data whist maintaining accuracy that is comparable with other classifiers.

By insight we mean that the technique provides an analyst with an appreciation of the influence that a feature value has on the class value. For example it is intuitive to ask the question: what influence does high blood pressure have on the prospects of having a heart disease? Or, does smoking suggest heart disease more than it suggests a lack of heart disease? A classifier that can provide simple to grasp answers to these sorts of questions could be expected to provide a degree of insight and be useful in real world data mining, particularly if its classification accuracy is comparable to other techniques.

Probabilistic approaches, such as Naive Bayes (Duda and Hart 1973) rely largely on maximising the probability that an example belongs to a given class and only indirectly provide any indication of the influence the feature values have on the classification. Connectionist approaches such as neural networks offer little or no direct insight although some attempts at deriving meaning from internal connection weights have been made (Setiono and Liu 1996). Geometric approaches such as Support Vector Machines (SVM) (Vapnik 1999) clearly identify the feature values in the support vectors as being the most important but it is difficult to generalise from this. Rule and tree based approaches provide some insight, though features are not certain to appear in the rules, or trees, even if they are influential to classification.

A further advantage provided by AWSum, that can be useful in real world data mining situations, is an assessment of the confidence of a classification. For example a forward feed neural network trained with back propagation can indicate that an example belongs to a given class but not whether this is a strong assertion or a weak assertion. The ability to assess the confidence of a classification is important in many diverse real world situations. In the medical field we may chose to medicate a patient if we are only reasonably sure of a cancer diagnosis but operate when we are very sure of the diagnosis. In a political scenario we may choose not to direct campaign time to those voters we are very confident will vote for us but dedicate resources to those voters that we are only mildly confident will votes for us.

AWSum focuses at the feature value level in order to identify the feature values and combinations of feature values that are sensitive and important to a classification. This is useful in fields such as medicine where a micro-focus on the influences on classification and an understanding of the data is critical. Other techniques such as trees and probabilistic approaches consider the importance of the values of a feature as a group. A simple example of this can be found in the Cleveland Heart dataset. If the values of the feature *age* are considered as a group, the relationship between *age* and *heart disease* identified would be that as *age* increases so does *heart disease* as seen in figure 1. This fails to identify the reversal of trend as we tend toward the extreme of *age*.

AWSum assesses the contribution of each feature value to the classification individually by assigning it a weight that indicates its influence on the class value. A weighted sum approach is taken, combining these influence weights into an influence score for the example. Figure 2 shows the weights AWSum has assigned to each feature value on a scale. The class values are placed at the extremes of the scale, -1 and 1. These extremes represent the points at which the probability of the relevant class outcome is 1. The influence weight of -0.03 assigned to *age 50* indicates that this value of *age* influences an outcome of *heart disease = yes* approximately the same amount of times as it influences an outcome of *heart disease = no*. The ratio of the occurrence of *heart disease = yes* to *heart disease = no* strengthens in favour of the class value

represented at the extreme as it nears that extreme. The reversal of the influence of *age* on *heart disease* can now be readily identified.

The intuition behind AWSum's approach is that each feature value has an influence on the classification that can be represented as a weight and that combining these influence weights gives an influence score for an example. This score can then be compared to a threshold in order to classify the example.

The algorithm for calculating and combining weights, and determining thresholds is explained in section 2.

Figure 1: Feature level focus

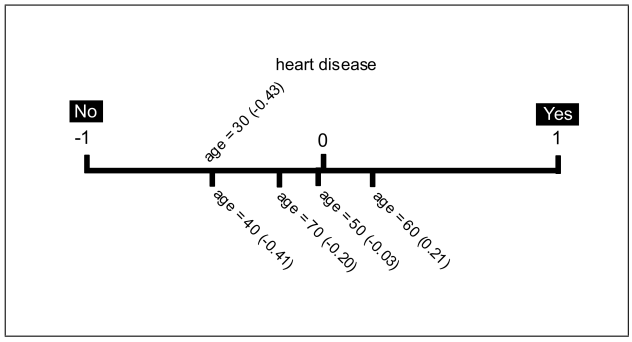


Figure 2: AWSum feature value focus

2 The Algorithm

The algorithm can be split into two major parts described separately below;

- Influence - Influence weights are established for each feature value that give a measure of the feature value's influence on the outcome and threshold/s are calculated
- Classification - New examples are classified by calculating an influence score for the example from the influence weights of the component feature values. This can be seen as a combination of evidence for the classification.

2.1 Influence

The first phase of the AWSum approach lays the foundations for classification and provides insight into the dataset by providing an influence weight for each feature value. For simplicity we will only consider binary classification tasks. Higher order tasks will be discussed later in section 5. For any feature value the sum of the conditional probabilities for each possible class value is 1 as the events are mutually exclusive, as illustrated for a binary outcome in equation 1.

$$Pr(O_1|F_v) + Pr(O_2|F_v) = 1 \quad (1)$$

Where: O_1 and O_2 are the first and second value on the class feature. F_v is the feature value.

A feature value's influence weight, W represents its influence on both class values and so it needs to simultaneously represent both probabilities from equation 1. To do this we arbitrarily consider one class outcome to be positive, and map probabilities to a range of 0 to +1. The other class is considered to be negative and map probabilities to a range of 0 to -1. The range of mapped probabilities for both feature values is therefore -1 to +1. By summing the two mapped probabilities we arrive at a single influence weight that represents the feature value's influence on both class values. Equation 2 demonstrates this calculation and figure 3 shows an example where $Pr\,(O_1|Fv_1) = 0.2$, or -0.2 when mapped and $Pr\,(O_2|Fv) = 0.8$.

$$W = Pr\,(O_1|Fv) + Pr\,(O_2|Fv) \quad (2)$$

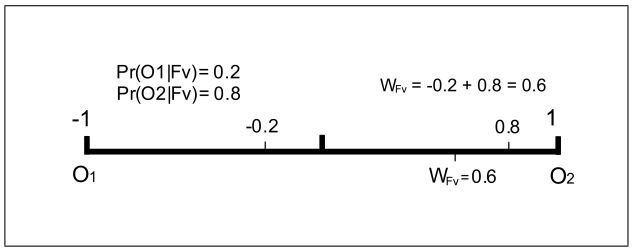


Figure 3: Binary class example

Additional assumptions are required to be made in the case of class features that are ternary or of a higher order. This is discussed below in Section 5.

2.2 Classification

Classification of an example is achieved by combining the influences from each of the example's feature values into a single score. By summing and averaging feature value influences we are able to arrive at a score that represents the evidence that the example belongs to one class and not to another. Equation 3 depicts this. Performing the combination by summing and averaging assumes each feature value influence is equally comparable. Although this is a relatively naive approach, it is quite robust as described later in this section.

$$e_1 = \frac{1}{n}\sum_{m=1}^{n} W_m \quad (3)$$

e_1 = the influence weight of the i^{th} example
n = the number of examples

The influence score for an example is compared to threshold values that divide the influence range into as many segments as there are class values. For instance, a single threshold value is required for a binary classification problem so that examples with an influence score above the threshold are classified as one class value, and those with a score below the threshold are classified as the other class value. Each threshold value is calculated from the training set by ordering the examples by their weight and deploying a search algorithm based on minimising the number of incorrect classifications. This is a simple linear optimisation problem that is solved by calculating the misclassification rate at each point along the scale. For instance, the examples with total influence scores that fall to the left of the threshold in Figure 4 are classified as class outcome A by AWSum. This however includes two examples that belong to class B in

the training set so these two examples are misclassified. Two examples to the right of the threshold are misclassified as class B when they are A's. In cases where there are equal numbers of correctly and incorrectly classified examples the threshold is placed at the mid-point under the assumption that misclassification of class A and B is equally detrimental.

New examples can be classified by comparing the example's influence score to the thresholds. The example belongs to the class in which its influence score falls.

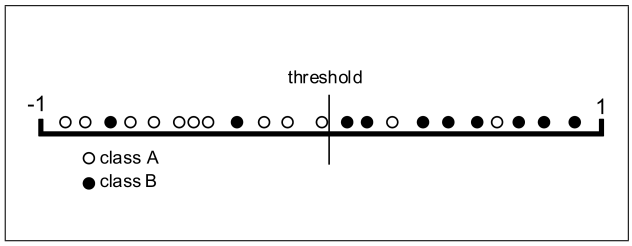


Figure 4: Threshold optimisation

AWSum is suited to nominal feature values and class outcomes although it is not necessary that they are ordinal. Continuous numeric features require discretisation before use in AWSum.

Classification accuracy of the AWSum approach compares favourably with that of many other algorithms. Experimental results are presented in section 4.

3 Extending the algorithm

The combining of influence weights for single feature values into a total influence score for an example and using this to classify is intuitively based however, it is plausible that feature values may individually not be strong influences on a class outcome but when they occur together the pair is a strong influence. For example both *drug A* and *drug B* may individually be influential toward low blood pressure but taken together lead to an adverse reaction that results in exceedingly high blood pressure. This sort of insight into a dataset can be very useful, particularly in medical domains.

The influence weights for each feature value pair can be calculated in the same way as they were for the single feature values. Equation 4 shows this calculation.

$$W = Pr\left(O_1|Fv_1, Fv_2\right) + Pr\left(O_2|Fv_1, Fv_2\right) + \quad (4)$$

$$\cdots Pr\left(O_k|Fv_1, Fv_2\right)$$

Where:
W =the influence weight of the pair.
O_1 is the first class value and O_k is the k^{th} class values.
Fv_1 is the 1^{st} feature value of the pair and Fv_2 is the 2^{nd}.

These pairs have the ability to both increase insight because the influences on the outcome are now more granular and increase accuracy. When using a feature value pair in the classifier the corresponding single feature weights are not used in order to avoid double counting the influence of the feature values.

3.1 Model selection

There is a need to select which feature value pairs to include in the classifier. There have been 2 methods employed in testing. The first , used on the UCI Cleveland Heart, Mushroom and Vote datasets is to include each feature value pair into the classifier and retain it if it improves classification accuracy. The second method, used on the Iris dataset was to select a support threshold for the feature value pairs and include all pairs that meet this threshold. The support for a feature value pair weight is a calculation of the number of times the pair occurs divided by the total number of examples.

3.2 Fine tuning

AWSum includes a technique that can be used to emphasise important feature values. A power is applied to the influence weights. This process occurs before the threshold algorithm is applied. Equation 5 shows the new calculation for the example weights. Note that the original sign of the influence weight is kept. This fine tuning technique gives more emphasis to influence weights whose absolute weight is larger.

$$e_1 = \frac{1}{n}\sum_{m=1}^{n} W_m^p \quad (5)$$

e_1 = the weight of the i^{th} example
n = the number of examples
p = the power to which the features values are raised
nb. the original sign of the influence weight is kept

4 Experiments

Four datasets were sourced from the University of California, Irvine's Machine Learning Repository (Blake et el 1988) for the comparative evaluation of the AWSum approach:

- ***Cleveland Heart***- 14 numeric features, 2 classes, 303 instances, 6 missing values
- ***Iris***- 5 numeric, continuous features, 3 classes - 1 linearly inseparable, 150 instances, 0 missing values
- ***Mushroom*** - 22 nominal features, 2 classes, 8124 instances, 2480 missing values
- ***Vote*** - 17 boolean features, 2 classes, 435 instances, 0 missing values

Classification accuracy has been assessed using 10 fold stratified cross validation. Table 1 represents classification accuracy using single influence weights only. The classification accuracy of AWSum on the four UCI datasets is comparable though not better than the Naive Bayes Classifier, TAN, C4.5 and the Support Vector Machine.

Table 1: Classifier comparison

Data	AWSum	NBC	TAN	C4.5	SVM
Heart	83.14	**84.48**	81.51	78.87	84.16
Iris	94.00	94.00	94.00	96.00	**96.67**
Mush	95.77	95.83	99.82	**100**	**100**
Vote	86.00	90.11	94.25	**96.32**	96.09
Avg	89.72	91.11	92.40	92.80	**94.23**

Table 2 shows the classification accuracies achieved by including influence pairs and they are quite comparable with other approaches. AWSum performs better than the others on the Cleveland Heart dataset and better than Naive Bayes and TAN on the Iris set. The Support Vector Machine outperforms all others on Iris, C4.5 and SVM perform perfectly on the Mushroom dataset and C4.5 outperforms the others on the Vote data.

Table 2: Classifier comparison including influence pairs

Data	AWSum	NBC	TAN	C4.5	SVM
Heart	**85.83**	84.48	81.51	78.87	84.16
Iris	94.67	94.00	94.00	96.00	**96.67**
Mush	99.37	95.83	99.82	**100**	**100**
Vote	95.86	90.11	94.25	**96.32**	96.09
Avg	93.93	91.11	92.40	92.80	**94.23**

Table 3 shows the best results achieved using influence values, influence pairs and the power based fine tuning method discussed in section 3.2. The average classification using 10-fold cross validation over the four sample datasets is slightly higher than the other approaches. The objective of this study was to advance a classifier that demonstrated comparable classification accuracy while providing some degree of insight about influential factors. Results indicate AWSum achieves comparable accuracy.

Table 3: Classifier comparison including influence pairs, fine tuned

Data	AWSum	NBC	TAN	C4.5	SVM
Heart	**87.18**	84.48	81.51	78.87	84.16
Iris	96.00	94.00	94.00	96.00	**96.67**
Mush	99.93	95.83	99.82	**100**	**100**
Vote	**97.01**	90.11	94.25	96.32	96.09
Avg	**95.03**	91.11	92.40	92.80	94.23

4.1 Insight

Insight is provided by identifying the influence that feature values have in classification. This can be important in identifying key features in a problem domain as well as eliminating features that are not important. Being able to represent the influence that feature values have on class values graphically provides a informative description of the problem domain. Figure 5 shows this information for the Cleveland Heart dataset. The figures in braces on the right of the scale are the influence pairs added to the classification model, although all pair weighings are calculated and can be used for insight.

Insight can be drawn from this figure. For example, if a patient does not get exercise induced angina (exang no) this has an influence weight of -0.39 indicating a moderate influence toward no heart disease. Similarly if the number of major vessels coloured by fluoroscopy is 0 (ca 0) there is a moderate influence toward no heart disease with an influence weight of -0.47, but if these two factors occur together there is a strong influence toward no heart disease as indicated by the influence pair weight of -0.72. These types of insights can help confirm an understanding of the problem domain or provide new and interesting paths for investigation.

The pairs included in figure 5 fall into two categories. Influence pairs, *cp - Typical angina and slope - down* and *cp - typical angina and thal - fixed defect* are examples of rare cases. They appear in the dataset 1% of the time but always lead to the same outcome. It can be important, particularly in a field such as medicine to be able to identify rare cases. Most techniques fail to identify rare cases because they are concentrating at a feature level. For example a rare case may not be include in a tree based classifier if collectively the values of the feature don't split the data well. Likewise an important dependency may not be modeled in an augmented Bayesian approach if collectively the values of the features are not important. The other three pairs in Figure 5 occur frequently and indicate interesting relationships between the feature values in the pair. This interest occurs because their influence as a pair is markedly different to their influence as single values. Their inclusion both increases accuracy and provides insight. Currently, a heuristic search is deployed to locate pairs of possible

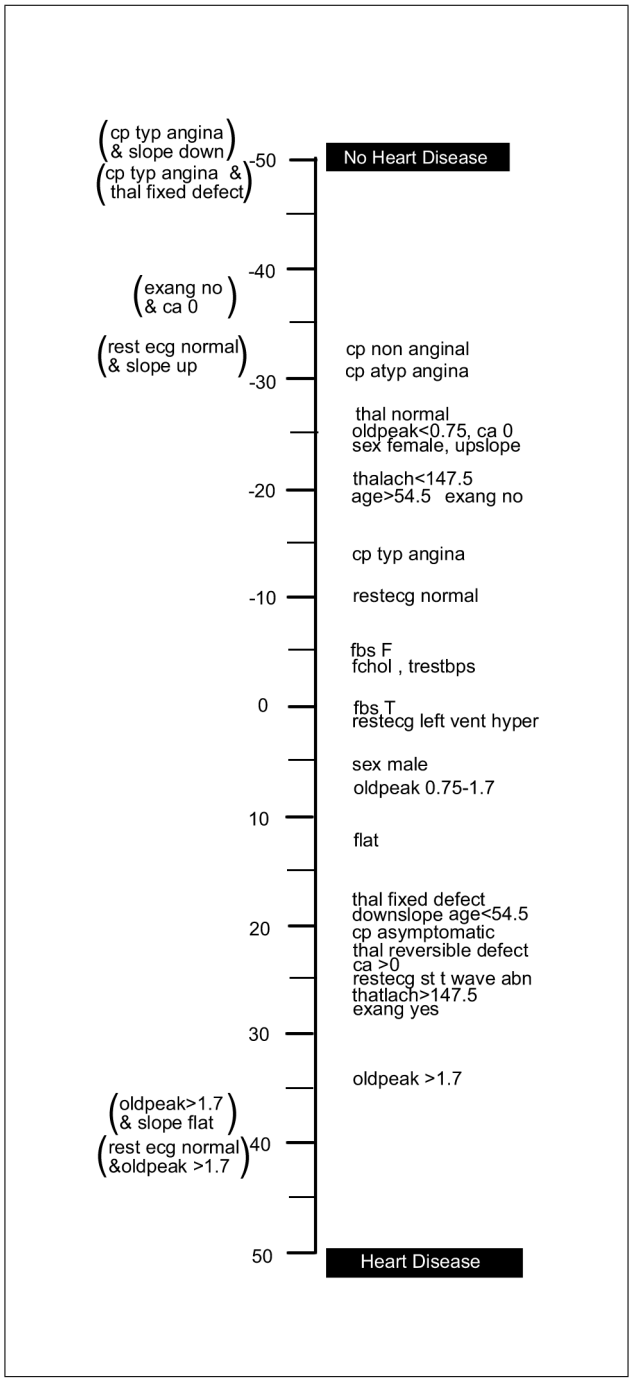


Figure 5: feature value and feature value pair weights

interest. Rare item pairs are considered interesting. Pairs that are not rare are considered interesting if the influence the pair has is markedly different from the average of the influences of each member of the pair. In this way we can identify both rare cases and important relationships that have high levels of support. Work is in progress to apply other search algorithms to enable pairs, triples and higher order pairings that are interesting to be identified.

Tree based classifiers tend to provide more insight than most classifiers and so the insights provided by AWSum are compared to those provided by the implementation of C4.5 (Quinlan 1993) provided by the Weka data mining tool (Witten and Frank 2000). The tree generated selects nodes from the root that are good for splitting the data with regard to the class values. The features closer to the root of the tree could be seen as more important in some senses but this does not convey the relative influence of the feature values in the same way as conveyed by AWSum.

The C4.5 tree generated on the Cleveland Heart dataset uses 9 of the 13 features. Those omitted includes *chol, fbs, trestbps* and *thalach*. It can also be seen that the tree does not necessarily contain all the important or influential features. For instance, *thalach* is identified as important in both AWSum and two feature selection techniques, first best and information gain attribute evaluator (Witten and Frank 2000) yet does not appear in the decision tree. This is understandable because features selected as nodes for a decision tree are those that represent the greatest information gain of the features in contention.

4.2 Discussion

The AWSum approach represents a concentration on feature values that most other techniques do not take. Other techniques tend to consider the values of a feature as a group and identify them as important or otherwise collectively. Probabilistic approaches such as augmented Bayes either relax Naive Bayes' independence assumptions by including dependencies between selected features or they look for independent features. In either case the feature values of a given feature are selected if they are collectively significant Tree based classifiers also focus on features as they search for the best features to split the data on at each node. This again is a collective consideration of the feature values of a given feature. Connectionist approaches such as neural networks include hidden nodes in a pragmatic approach that consumes any concentration on feature values. Geometrical approaches such as SVM (Vapnik 1999) consider a select number of important boundary instances, or support vectors, in order to construct a linear function to separate the class and so are not focusing on identifying the influence of feature values.

In contrast to many other classifiers, AWSum is simple, scalable and easy to implement. Classification processes are easily understood by the non expert and this is often as important as the classification itself. AWSum's use of conditional probability is markedly different to that of Bayesian approaches, such as Naive Bayes. NB compares the probability that the example's feature values were derived by the class outcome, scaled by the prior probability of the class. AWSum, on the other hand, uses conditional probability to calculate a weight that indicates a feature values influence on the class value and combines these influences for each example, comparing the result with a threshold. This style of approach can be seen as a combination of evidence although it is a very different approach to that of the Dempster/Schafer work (Shafer 1976).

The use of pairs or combinations of feature values in AWSum differs from that of probabilistic approaches like Naive Bayes. These style of approaches look for computationally economical ways to model probabilities. Rather than this AWSum is looking for combinations of feature values that may have a strong influence on the class value, and using these as pieces of evidence for a given class outcome.

The addition of pairs involves calculating a weight for each feature value pair in the dataset and so adds to the approach computationally. These calculations can be done in a single pass of the dataset and used in a lookup table to classify. This means that the overhead is not large.

5 Higher dimension class features

In order to represent 3 or more class values on a linear scale certain assumptions need to be made. The class values need to be considered as ordinal. For example if the 3 class outcomes are light, medium and heavy and we have 5 light examples, 0 medium examples and 5 heavy examples we have conditional probabilities of $Pr(light|F_v) = 0.5$, $Pr(medium|F_v) = 0.0$ and $Pr(heavy|F_v) = 0.5$. The feature value, F_v would be assigned a weight of 0 using AWSum which places it in the middle of the influence scale. In terms of conditional probability this is inconsistent as there are no medium examples, but in terms of influence on the outcome it is intuitive because we can reasonably say that the influence of 5 heavy examples and 5 light examples is the same as 10 medium examples. This approach can be demonstrated to classify well even in cases such as the Iris dataset where the outcomes are not ordinal but the visualisation may be misleading in that a value at the middle of the scale could appear there either because there is a high probability of that outcome or because class values at the extremes have the same probability.

For a ternary class outcome, as illustrated in figure 6, the influence value weight can be decided using the the conditional probabilities of the 2 class values represented at the extremes of the scale. Equation 6 illustrates the calculation.

Problems that contain 4 or more class values can simply be seen as combinations of scaled binary outcomes that can be summed to give an influence weight. Figure 7 shows a situation with 4 class values. Each binary feature weight is calculated as per equation 2, with the weight for outcomes 2 and 3 being scaled and summed as per equation 7. This approach to calculating feature value weights can be extrapolated to any number of feature values

$$W = \frac{-Pr\left(O_1|Fv\right) - Pr\left(O_3|Fv\right)}{2} \tag{6}$$

$$W = W_{O_{1,4}} + \frac{1}{3} W_{O_{2,3}} \tag{7}$$

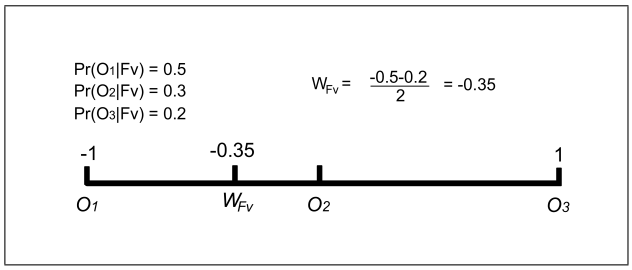


Figure 6: Three class values

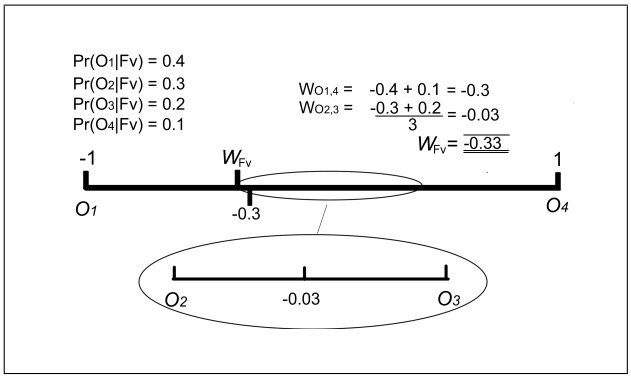


Figure 7: Four class values

6 Conclusion

AWSum demonstrates that classification accuracy can be maintained whist providing insight into the problem domain. This sort of insight can provide important information in itself or be used in preprocessing the data for another approach. It is not intended that AWSum replace traditional approaches but rather that it provides a different and possibly useful resource for analysts to use in approaching real world datasets. It may be that its usefulness is in identifying important features, visualising the problem domain or in its classification ability. It is hoped that in providing insight with classification that data mining can be made more understandable and accessible to the non expert. Future directions for this work include the addition of influence weights for three and four feature value combinations to both test any increase in accuracy and to provide insight into important combinations of feature values. It is also envisaged that when classifying data with more than two class outcomes that a multidimensional scale may be useful to classification if not visualisation.

References

Blake, C.L., Newman, D.J., Hettich, S. & and Merz, C.J. (1988), UCI repository of machine learning databases.

Duda, R., Hart, P. (1973), *Pattern Classification and scene analysis.*, John Wiley and Sons.

Quinlan, J. (1993), *Programs for Machine Learning.*, Morgan Kaufmann

Setiono, R. & Liu, H. (1996), Symbolic Representation of Neural Networks, *in* 'Computer', Vol. 29,IEEE Computer Society Press, Los Alamitos, CA, USA, pp. 71–77.

Shafer, G. (1976), *A Mathematical theory of evidence.*, Princeton University Press.

Vapnik, V. (1999), *The nature of statistical learning theory.*, Springer - Verlag.

Witten, I.H. & Frank, E. (2000), *Data Mining: Practicle machine learning tools and techniques with java implementations.*, Morgan Kaufmann

A New Efficient Privacy-Preserving Scalar Product Protocol

Artak Amirbekyan **Vladimir Estivill-Castro**

School of ICT, Griffith University,
Meadowbrook QLD 4131, Australia
Email: A.Amirbekyan@gu.edu.au

Abstract

Recently, privacy issues have become important in data analysis, especially when data is horizontally partitioned over several parties. In data mining, the data is typically represented as attribute-vectors and, for many applications, the scalar (dot) product is one of the fundamental operations that is repeatedly used.

In privacy-preserving data mining, data is distributed across several parties. The efficiency of secure scalar products is important, not only because they can cause overhead in communication cost, but dot product operations also serve as one of the basic building blocks for many other secure protocols.

Although several solutions exist in the relevant literature for this problem, the need for more efficient and more practical solutions still remains. In this paper, we present a very efficient and very practical secure scalar product protocol. We compare it to the most common scalar product protocols. We not only show that our protocol is much more efficient than the existing ones, we also provide experimental results by using a real life dataset.

Keywords: Privacy Preserving Data Mining.

1 Introduction

Data mining technology allows the analysis of large amounts of data. Analysis of personal data, or analysis of corporate data (by competitors, for example) creates threats to privacy (Estivill-Castro & Brankovic 1999). Moreover, never have globalization and international collaborations placed as much demand on partnerships between governments and/or corporations as they do today. Data mining has been identified as one of the most useful tools for the fight on terror and crime (Mena 2003). However, the information needed resides with many different data holders that must share their data with each other; thus, data privacy becomes extremely important. Parties may not trust each other, but all parties are aware of the benefit brought by such collaboration. In the privacy preserving model, all parties of the partnership promise to provide their private data to the collaboration, but none of them wants the others or any third party to learn much about their private data.

Nowadays, computers can manipulate large databases and perform many data analysis tasks using data-mining techniques. Our purpose is, during such autonomous data analysis, to preserve privacy of individuals and corporations. Data is now available from companies, shops, medical clinics, hospitals and it can be used to detect patterns to identify individuals who can be dangerous to society or obtain information which will help to make preventative decisions. But such a task is not possible without some level of privacy protection as the companies or hospitals policies protect private information. Thus a method that can achieve a balance on knowledge discovery and privacy protection is highly desirable.

For instance, Peter wants to apply for a loan to buy a house. He goes to Bank A and supplies the necessary documentation. Bank A uses k-near-neighbours (k-NN) classification to determine the class of an applicant as either RISKY or SAFE. This automated system classifies Peter as RISKY. Obviously, the k-NN were retrieved for within the database that Bank A holds. It could happen that Bank A does not have many clients that are "close" to Peter and this could lead to a wrong classification. Consequently, Bank A has a possible loss of profit. It is clear that the larger the database, the more accurate the classification. Thus, if it were possible to somehow combine the databases of bank A, bank B and bank C, classification would have been more precise — Peter could have been classified as SAFE. But this scenario is not possible because privacy restrictions would not allow banks to provide access to each other's databases. Here is where privacy-preserving data mining comes to the rescue. This is a typical case of so called horizontally partitioned data. In privacy-preserving data mining settings, banks do not need to reveal their databases to each other. They can still apply, for instance, k-NN classification but preserve the privacy of their data at the same time.

For most data mining algorithms, the data is encoded as vectors in high dimensional space[1]. Therefore, secure computation (preserving privacy) of dot-products (scalar products) is fundamental for many data analysis tasks. The filed of statistics is supported by many operations on vectors and matrices because of its long tradition of manipulating vectorial arrangements of data. An examples of data analysis where privacy preservation has been studied and where the dot-product plays a central role is regression analysis (Amirbekyan & Estivill-Castro 2007*b*, Du et al. 2004, Du & Atallah 2001). Another example where privacy-preserving data analysis uses version of dot-product with some security considerations is the calculation of the product of matrices (Du et al. 2004, Du & Atallah 2001). Examples of core data mining tasks that heavily use scalar products are clustering (Amirbekyan & Estivill-Castro 2006, Estivill-Castro 2004) and decision trees classification (Du &

[1] Attribute-vectors are the common input for learning algorithms like decision trees, artificial neural network or for clustering algorithms like K-Means (Vaidya & C. Clifton 2003) or *DBSCAN* (Amirbekyan & Estivill-Castro 2006).

Zhan 2002). All sorts of data mining tasks that require nearest neighbour calculations, such as k-NN classification (Amirbekyan & Estivill-Castro 2007*a*, Kantarcioğlu & Clifton 2004, Shaneck et al. 2006). k-NN queries also enable Local Outlier Detection (Breunig et al. 2000), Shared Nearest Neighbour Clustering (Shaneck et al. 2006) and are supported by associative queries. In turn, privacy-preserving associative queries (Amirbekyan & Estivill-Castro 2007*a*) are supported by various types of metrics, whose metrics demands privacy-preserving calculations (Amirbekyan & Estivill-Castro 2007*a*, Vaidya & Clifton 2004). Metrics that compute the distance between vectors use dot-products, in fact, the dot-product is essentially a metric in itself, as illustrated by the cosine metric. Moreover, usually the dot-product is used repeatedly in all this large family of applications. The dot-product is therefore, a central building block. Computing it efficiently becomes essential, as an inefficient or slow fundamental operation has a direct and real impact on the performance of all the protocols that use it. Furthermore, efficiency is not the only issue, but practicality is an important issue as well. There is no use to describe a data mining task using a theoretical protocol for the dot-product if this protocol is in fact, difficult to implement in practice. This, means that a secure dot-product protocol should be easily implementable and not based on some theoretical approaches that are hard to implement, which propagates the lack of implementation of data analysis methods in the privacy-preserving context. Thus, we show that our protocol for a privacy-preserving dot-product is efficient and very practical, as opposed to other available alternatives in the literature.

2 Privacy-Preserving Computation

We study collaboration between several parties that wish to compute a function of their collective databases. In fact, they are to conduct data mining tasks on the joint data set that is the union of all individual data sets. Each wants the others to find as little as possible of their own private data.

To focus the discussion on privacy-preserving collaboration, we will regularly name at least two of the parties as Alice and Bob. In the so called horizontally partitioned data (see Fig. 1), some of the records are owned by Alice and the others by Bob. Obviously, for more than two parties, every party will own some part (a number of records) from the database.

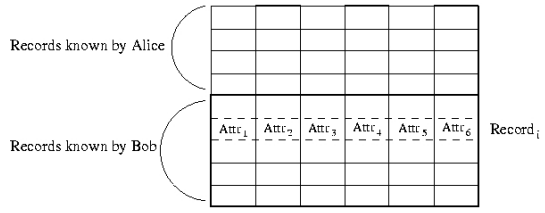


Figure 1: Horizontally partitioned data.

A direct and naive use of data mining algorithms on the union of the data requires one party to receive data (every record) from all other parties, or all parties to send their data to a trusted central place. The recipient of the data would conduct the computation in the resulting union. In settings where each party must keep their data private, this is unacceptable. Note that, for horizontally partitioned data, the more parties are involved, the more records are involved and the larger is the global database.

Our approach is based on the theory developed under the name of "secure multiparty computation"-SMC (Goldreich 2004). Here, Alice holds one input vector $\vec{x}$ and Bob holds an input vector $\vec{y}$. They both want to compute a function $f(\vec{x},\vec{y})$ without each learning anything about the other's input expect what can be inferred from $f(\vec{x},\vec{y})$. Yao's Millionaires Problem (Yao 1982) provides the origin for SMC. In the Millionaires Problem, Alice holds a number a while Bob holds b. They want to identify who holds the larger value (they compute if $a > b$) without either learning anything else about the other's value. The function $f(x,y)$ is the predicate $f(x,y) = x > y$.

Secure multi-party computation under the semi-honest model (Goldreich 2004) has regularly been used for privacy-preserving data mining (Du & Atallah 2001, Du & Zhan 2002, Vaidya & C. Clifton 2003).

Let us give a formal definition for the model of computation for our protocols. The semi-honest party is the one who follows the protocol correctly, but at the same time keeps information received during communication and final output, for further attempt to disclose private information from other parties. A semi-honest party is sometimes called an *honest but curious one* (Ioannidis et al. 2002).

The most common formal definition for the semi-honest model. is the following (Goldreich 2004):

Definition 1 *(privacy w.r.t. semi-honest behavior): Let* $f : \{0,1\}^* \times \{0,1\}^* \longrightarrow \{0,1\}^* \times \{0,1\}^*$ *be a functionality, and* $f_1(x,y)$ *(respectively,* $f_2(x,y)$*) denote the first (resp., second) element of* $f(x,y)$*). Let* Π *be two-party protocol for computing f. The view of the first (resp., second) party during an execution of protocol* Π *on* (x,y)*, denoted* $view_1^\Pi(x,y)$ *(resp.,* $view_2^\Pi(x,y)$*), is* $(x, r_1, m_1, \cdots, m_t)$ *(resp.,* $(y, r_2, m_1, \cdots, m_t)$*), where* r_1 *represents the outcome of the first (resp.,* r_2 *the second) party's internal coin tosses, and* m_i *represents the* i*-th message it has received. The output of the first (resp., second) party after an execution of* Π *on* (x,y) *is denoted* $output_1^\Pi(x,y)$ *(resp.,* $output_2^\Pi(x,y)$*), and is implicit in the party's view of the execution, and* $output^\Pi(x,y) = (output_1^\Pi(x,y), output_2^\Pi(x,y))$

- *(general case) We say that* Π *privately computes* f *if there exists a probabilistic polynomial-time algorithm, denoted* S_1 *and* S_2*, such that*

$$\{S_1(x, f_1(x,y), f(x,y)\}_{x,y\in\{0,1\}^*} \equiv^C \{view_1^\Pi(x,y), output^\Pi(x,y)\}_{x,y\in\{0,1\}^*}$$
$$\{S_2(y, f_2(x,y), f(x,y)\}_{x,y\in\{0,1\}^*} \equiv^C \{view_2^\Pi(x,y), output^\Pi(x,y)\}_{x,y\in\{0,1\}^*}$$

where $\equiv^C$ *denotes computational indistinguishability by (non-uniform) families of polynomial-size circuits. Here* $view_1^\Pi(x,y), view_1^\Pi(x,y), output_1^\Pi(x,y)$ *and* $output_2^\Pi(x,y)$ *are related random variables, defined as a function of the same random execution. In particular,* $output_i^\Pi(x,y)$ *is fully determined by* $view_i^\Pi(x,y)$*.*

This definition basicly says, that a computation is secure if the view of each party during the execution of the protocol can be simulated from the input and the output of that party. Thus, for a security proof, it is enough to show the existence of a simulator for each party that satisfies the above equations, and no other party notices that its partner has been replaced by the simulator. Note however, that whatever information is derived or inferred from final result cannot obviously be kept secret.

For instance, Alice, Bob and Charles each hold private numbers and want to calculate the average of their numbers. Using SMC protocols they discovered that the average is m. Assume Charles holds c that is greater than m. This immediately discloses that at least one of the other parties holds a number less than the value m. This information was inferred from the final result and not during the execution of the protocol, thus the protocol still can be considered secure under the semi-honest model.

Theorem 1 *(Composition theorem for the semi-honest model): (Goldreich 2004) Suppose that g is privately reducible to f and that there exists a protocol for privately computing f. Then there exists a protocol for privately computing g.*

This approach became very popular because cryptography offers a well defined model for privacy, which includes methodologies for proving and quantifying it. Moreover, there exist many tools of cryptographic algorithms that can be used for implementing privacy-preserving data mining algorithms.

The SMC literature has a general solution for all polynomially bound computations (Goldreich et al. 1987). This generic "shares" solution computes $f(\vec{x}, \vec{y})$ for a polynomial-time f using private input $\vec{x}$ from Alice and private input $\vec{y}$ from Bob. Alice learns nothing about $\vec{y}$ except what can be computed from $f(\vec{x}, \vec{y})$ and similarly Bob learns nothing about $\vec{x}$ except what can be inferred from $\vec{y}$ and $f(\vec{x}, \vec{y})$. Why, if such a solution exists, is there so much interest in protocols for SMC? The first aspect is that the general solution requires f to be explicitly represented as a Boolean circuit of polynomial size. Even if represented as a circuit of polynomial size in its input, the input must be very small for the circuit to have practical polynomial size. This means, the sub-task that uses this result must be on small inputs, a constraint difficult to meet in data mining applications. Third, the constants involved are not small, so once the circuit is described the parties enter into a protocol, holding shares of the inputs to gates and shares of the outputs of gates. Fourth, the privacy-preserving literature shows the need for practical and efficient solutions that are not based on this general theoretical solutions. It also shows that much more efficient solutions exist for special cases of f.

3 Related Work

Assume the following scenario: Alice holds a vector $\vec{a}$ and Bob holds vector $\vec{b}$ (both with n elements). The goal is to compute $\vec{a}^T \cdot \vec{b}$ securely. We review the most common secure scalar product protocols (Du & Zhan 2002, Vaidya & C. Clifton 2002, Du & Atallah 2001, Ioannidis et al. 2002). Some secure scalar product protocols (Du & Zhan 2002) use a semi-trusted third party's commodity server. Semi-trusted means that it should not learn any private information from the parties and should not collude with either Alice or Bob. This solution is not considered secure (Vaidya et al. 2006). under the semi-honest model since it uses a third party. The communication cost of this protocol is $4n$, which is 4 times more expensive than in the distributed non-private setting (*DNSP*) cost of a two-party scalar product (the *DNSP* cost of a scalar product is defined as the cost of computing $\vec{a}^T \cdot \vec{b}$ without the privacy constraints, namely one party just sends its data to the other party). The communication cost can be reduced to $2n$ if the commodity server sends just the seeds to the parties.

Another two solutions to the scalar product protocol have been proposed (Du & Atallah 2001, Vaidya & C. Clifton 2002). Both of these provide privacy preservation without using a third party. However the communication and computation costs are more expensive than if one uses the solution with commodity server. Both solutions have communication cost $O(n)$ but with much larger constants (4 rounds with bitwise cost of $2nM$, where M is the maximum number of bits needed to represent any input value) under the $O(n)$ complexity (recall that n stands for the size of the vectors). Computational cost is also high, for instance, the second solution has a $O(n^2)$ computational cost (Vaidya & C. Clifton 2002).

Two solutions are presented by Du and Atallah for the secure scalar product named Protocol 1 and Protocol 3 (Du & Atallah 2001). Protocol 3 is more efficient than Protocol 1. For Protocol 3, by setting the security parameter $\mu = p^t$, which should be large enough, the communication cost of Protocol 3 becomes $4\log(\mu\frac{nd}{\log(n)})$, where d is the number of bits needed to represent any number in the input.

For another secure dot-product protocol (Ioannidis et al. 2002). the authors present two kinds of overhead analysis. First, the communication overhead is the amount of extra communications compared to the *DNSP* cost. That is, the communication overhead with respect to the $n + 1$ messages of *DNSP* (where Bob sends his n values and Alice sends the result back). The second type of overhead is the computation overhead with *DNSP* (that requires n products and $n - 1$ additions that Alice performs). Thus, the computation overhead is the extra computation performed. The authors claim that their protocol is more efficient in both overheads than the protocols that use conventional cryptographic techniques Their experimental results show a total overhead to 4.69 on average. However, we tried to carry out the analysis of this protocol ourselves and we found that this protocol was probably described incorrectly or errors were introduced during publication. Namely, the published description seems to compute a scalar product incorrectly. After private communications with the authors, they acknowledged the problem that we have raised. So, at this stage, we exclude this protocol from further discussion.

4 Review of Tools to be Used

We now describe some cryptographic tools.

4.1 Homomorphic encryption

An encryption scheme is called *additive homomorphic* (Paillier 1999) if it has the following property:

$$E(x_1) \times E(x_2) = E(x_1 + x_2).$$

But other implementations are possible (Naccache & Stern 1998, Okamoto & Uchiyama 1998) including one using RSA. By induction, it is not hard to show than in an homomorphic scheme,

$$E(x_1) \times E(x_2) \times \cdots \times E(x_n) = E(x_1 + x_2 + \cdots + x_n).$$

4.2 Review of the Add Vectors Protocol

The technique was introduced for manipulation of vector operations as the "permutation protocol" (Du & Atallah 2001) and is also known as the "permutation algorithm" (Vaidya & C. Clifton 2003).

In this protocol, Alice has a vector $\vec{x}$ while Bob has vector $\vec{y}$ and a permutation π. The goal is for Alice to obtain $\pi(\vec{x} + \vec{y})$; that is Alice obtains the sum $\vec{s}$ of the vectors in some sense. The entries are randomly permuted, so Alice cannot perform $\vec{s} - \vec{x}$ to find $\vec{y}$.

Also, Bob is not to learn $\vec{x}$. Solutions for $P = 2$, that is, two parties, are based on homomorphic encryption.

Protocol 1 *1. Alice produces a key pair for a homomorphic public key system and sends the public key to Bob. We denote by $E(\cdot)$ and $D(\cdot)$ the corresponding encryption and decryption system.*

2. Alice encrypts $\vec{x} = (x_1, \cdots, x_n)^T$ and sends $E(\vec{x}) = (E(x_1), \cdots, E(x_n))^T$ to Bob.

3. Using the public key from Alice, Bob computes $E(\vec{y}) = (E(y_1), \cdots, E(y_n))^T$ and uses the homomorphic property to compute $E(\vec{x}+\vec{y}) = E(\vec{x}) \times E(\vec{y})$. Then, he permutes the entries by π and sends $\pi(E(\vec{x}+\vec{y}))$ to Alice.

4. Alice obtains $D(\pi(E(\vec{x}+\vec{y}))) = \pi(\vec{x}+\vec{y})$.

This can be extended to the case of $P \geq 3$ vectors, that is $P > 2$ parties are involved. Because the parties do not collude, this protocol accepts faster implementations that do not need to permute the result, because $D(\cdot)$ is known only by Alice, and Alice will get the value $E(\vec{v}_2 + \cdots + \vec{v}_P)$, where $\vec{v}_i$ is the vector owned by i^{th} party. The algorithm is as follows:

Protocol 2 *1. The 1^{st} party (Alice), generates $E(\cdot)$ and $D(\cdot)$, then sends only $E(\cdot)$ to the other parties.*

2. Then, the P^{th} party encrypts his data $E(\vec{v}_P)$ and sends it to the $(P-1)^{th}$ party.

3. Next, the $(P-1)^{th}$ party encrypts his data $E(\vec{v}_{P-1})$ and using the homomorphic encryption property computes

$$E(\vec{v}_{P-1}) \times E(\vec{v}_P) = E(\vec{v}_{P-1} + \vec{v}_P)$$

and sends this to the $(P-2)^{th}$ party.

4. The protocol continues until Alice (the first party) will get $E(\vec{v}_2 + \cdots + \vec{v}_{P-1} + \vec{v}_P)$ and she adds her data in the same way.

5. As Alice owns $D(\cdot)$, she decrypts the results and sends them to all the other parties.

Note that in Step 1, the P^{th} party does not need to permute his result because the $(P-1)^{th}$ party does not know $D(\cdot)$ to decrypt. It is also the case that the encryption mechanism could be much less costly. For example, if we are prepared to know the distribution of values, although no specific value is revealed, we do not need homomorphic encryption. In this case $E(\cdot)$ could be as simple as adding a random number in a sufficiently large additive field F (or X-OR with a random bit mask) and consequently $D(\cdot)$ will be subtracting the random number previously added.

One can easily also notice that for $P > 2$ (when we do not use permutation) this method can be applied for adding not only vectors but matrices or just numbers as well. This, however, sometimes is called the "SECURE SUM PROTOCOL" (Vaidya et al. 2006).

5 New Privacy-Preserving Scalar Product Protocol

In this section we propose a new simple scalar vector product protocol which is based on the ADD VECTORS PROTOCOL (see Section 4.2). This protocol is very simple and it is easy to implement. Depending on the domain and encryption it is also very efficient. In this protocol again, Alice has a vector $\vec{a}$ and Bob has another vector $\vec{b}$ (both with n elements). Alice and Bob use the protocol to compute the scalar product $\vec{a}^T \cdot \vec{b}$ between $\vec{a}$ and $\vec{b}$, such that Alice gets $\vec{a}^T \cdot \vec{b}$. Note that

$$2a_ib_i = a_i^2 + b_i^2 - (a_i - b_i)^2,$$

therefore

$$2\sum_{i=1}^{n} a_ib_i = \sum_{i=1}^{n} a_i^2 + \sum_{i=1}^{n} b_i^2 - \sum_{i=1}^{n}(a_i - b_i)^2;$$

thus the protocol works as follows:

Protocol 3 *(New Scalar Product Protocol)*

1. Alice and Bob apply the ADD VECTORS PROTOCOL *for Alice to obtain $\pi_0(\vec{a} - \vec{b})$, were π_0 is a permutation generated by Bob.*

2. Alice can obtain

$$\sum_{\pi_0(i)=1}^{n} (a_{\pi_0(i)} - b_{\pi_0(i)})^2 + \sum_{i=1}^{n} a_i^2$$

and Bob can compute $\sum_{i=1}^{n} b_i^2$. [2]

3. Now Bob can send $\sum_{i=1}^{n} b_i^2$ to Alice which will allow Alice to compute the scalar product, that is

$$\begin{aligned} 2\vec{a}^T \cdot \vec{b} &= 2\sum_{i=1}^{n} a_ib_i \\ &= \sum_{i=1}^{n} a_i^2 + \sum_{i=1}^{n} b_i^2 - \sum_{\pi_0(i)=1}^{n} (a_{\pi_0(i)} - b_{\pi_0(i)})^2. \end{aligned}$$

Alice learns all the values $\pi(a_i - b_i)$ from the information collected during the protocol's execution. However, this is not enough for Alice to discover any of Bob's private data, because of the permutation applied by Bob. Essentially, this protocol's security is the same as the security of ADD VECTORS PROTOCOL. Note also that Alice learns the value of $\sum_{i=1}^{n} b_i^2$, but again (for $n \geq 2$) is insufficient to learn any of Bob's private data.

Note that the encryption used in the ADD VECTORS PROTOCOL can be as simple as adding a random vector consisting of the same random number. Alice then adds this random vector to her vector and sends the results to Bob, whereas Bob only adds his vector to the sum vector received and performs a permutation before he sends it back to Alice. However, this solution should be handled carefully, because if Bob knows any of the coordinates in Alice's vector, then he immediately discovers the random number and consequently all of Alice's vector.

5.1 Analysis of security for special cases

Assume all the values in Alice's vector are equal, then there are some information leaks in the ADD VECTORS PROTOCOL. First, Alice learns all of Bob's values, although not of all of Bob's information, since she learns them shuffled. That is, she learns the set of values in Bob's vector. On the other hand, because Bob learns all of Alice's encrypted values, Bob learns that Alice's values are equal. Therefore, Bob could detect the pathological case before continuing with the protocol. While this premature abortion of the protocol by Bob will ensure that his values remain

[2] Note that, if they stop execution in this step they will have scalar product distributed in private shares.

private, Alice would not be protected against the fact that Bob discovered Alice holds a vector with all entries constant (although he does not know the value of this constant).

Since our scalar vector product protocol is based on the ADD VECTORS PROTOCOL, we have to take this issue into account. This leads to consider an analysis of the ADD VECTORS PROTOCOL in more detail. First, at least for regression, it is likely that Alice holds a column in a matrix with all values set to a constant. Since the interpolation matrix holds rows with entries $\langle 1, x_i, x_i^2, x_i^3, \ldots \rangle$ for each data point, the first column may be all set to 1. This may be a fact that Bob knows anyways because of the nature of regression, but this should not mean Bob needs to reveal a set of values on one of his vectors. This unpleasant situation can be avoided for our protocol by the fact that the scalar vector product has the following property

$$(\vec{a} + \vec{r})^T \cdot \vec{b} = \vec{a}^T \cdot \vec{b} + \vec{r}^T \cdot \vec{b}.$$

Namely, assume Alice and Bob want to compute $\vec{a}^T \cdot \vec{b}$. Alice can generate a random vector $\vec{r}$ and add to her vector $\vec{a}$. Thus, even if Alice has a constant vector $\vec{a}$, the fact that $\vec{r}$ was randomly generated will ensure that $\vec{a} + \vec{r}$ will be always a vector with non-equal values. Now, they can use our scalar product protocol twice for computing $(\vec{a}+\vec{r})^T \cdot \vec{b}$ and $\vec{r}^T \cdot \vec{b}$. Then, Alice can obtain

$$\vec{a}^T \cdot \vec{b} = (\vec{a} + \vec{r})^T \cdot \vec{b} - \vec{r}^T \cdot \vec{b}$$

and send it to Bob. This solution obviously will double the cost of our scalar product protocol, but it still will be efficient than all existing ones and avoid the problem of the ADD VECTORS PROTOCOL.

6 Experimental Results

Privacy is not without cost. In a situation when privacy protection is necessary, protocols that do not ensure some level of privacy will not be considered at all. The performance of privacy-preserving protocols is usually studied with respect to the cost in *DNSP*. Here we have two kinds of overhead, namely communication overhead and computational overhead. Communication overhead of our protocol is very easy to derive. For each entry in the vector, our protocol requires one messages from Alice to Bob, and again one messages from Bob to Alice. Bob needs to send $\sum_{i=1}^{n} b_i^2$ to Alice, whereas Alice sends the result to Bob. So, the communication cost is $2n + 2$, where n is the dimension of the vectors. In the *DNSP* model, Bob would send his vector to Alice for her to compute the scalar product and Alice will need to send the result to Bob. This is a communication cost of $n + 1$. Thus, the communication overhead is $n + 1$ messages carrying a floating-point value. In special cases when Alice's has a constant vector, the cost is doubled, making it $4n+4$ and the overhead is $3n+3$.

In order to calculate the computational overhead, we performed the following analysis. Alice will compute $\sum_{i=1}^{n} a_i^2$, and Bob will compute $\sum_{i=1}^{n} b_i^2$ locally. Then, there will be one executions of the ADD VECTORS PROTOCOL which implies n data swaps to implement the permutation π_0 as suggested by (Reingold et al. 1977). Also, there needs to be $n-1$ random indexes generated to create this permutation. Moreover, we need n encryptions for the ADD VECTORS PROTOCOL, even if they are implemented as adding a random number. While this overhead is linear in the dimension of the vectors, it now involves a slightly more diverse set of fundamental operations (it is not just additions and multiplications, but there are random bits generated as well as data swaps).

To evaluate the overhead this represents in a realistic setting, we have implemented our algorithm in C++ and with SHELL scripts. We use the CoIL 2000 Challenge (van der Putten & van Someren 2000) dataset which contains information on customers of an insurance company. The data consists of 86 variables with 5821 records and includes product usage data and socio-demographic data derived from zip area codes [3]. We compute the scalar product for 500 pairs of vectors from this datasets by using our protocol. The scalar product computation without privacy constrains, that is *DNSP*, on average requires 1.413 microseconds of execution time (with a confidence interval of 95% given by ± 0.144709). Our protocol requires 6.37 microseconds of execution time on average (with a ± 0.479215 95%-confidence interval). Thus, the overhead on average is around 4.51 times. Given our analysis, this overhead is reasonable and within the expected range.

7 Final Remarks

While our protocol is efficient, if used without the care for special cases, it is as secure as the ADD VECTORS PROTOCOL. The ADD VECTORS PROTOCOL is not covered under the semi-honest model of multiparty computation, because only one party obtains the output. It also suffer the weaknesses mentioned in Section 5.1, when all the values in Alice's vector are equal.

We need to refer to the semi-honest model of computation to clarify the issue further. It is very hard that two parties compute the sum of respective private values and do not disclose information on such private values because each can subtract its private value to the commonly known sum and discover the other side's value. To fit the semi-honest model, we could formulate the problem better. We call the new problem the *add-vectors permuted-outputs computation*. Here, we require that two parties holding n-dimensional private vectors each find the set of values of the sum of these two vectors (and in a way that all possible permutations are equally likely to correspond to the actual vectorial sum).

Note that the current ADD VECTORS PROTOCOL in the literature could almost achieve a solution to the *add-vectors permuted-outputs computation* problem by Alice applying a permutation σ of her own to the values of $\pi(\vec{a}+\vec{b})$ and sending $\sigma(\pi(\vec{a}+\vec{b}))$ to Bob. However, this modified ADD VECTORS PROTOCOL still suffers from the fact that if all of Alice's values are equal, then Bob finds this fact and Alice finds the set of values in Bob's vector.

The ideal, theoretical semi-honest protocol for *add-vectors permuted-outputs computation* would also leak information. A party with a vector with equal entries engaging in an add-vectors permuted-outputs ideal semi-honest protocol, can retrieve from its input and the permuted output, the set of values of the other party. Moreover, the theoretical solution ensures the other party cannot detect this leak. So, perhaps it is an advantage the way the current ADD VECTORS PROTOCOL operates. It is also an interesting problem if the theoretical solution for the add-vectors permuted-outputs setting can be made implementable. However our protocol does not suffers from this fact. Since, as we have shown in Section 5.1, we can always avoid this situation by adding a small overhead in communication and computation costs.

[3] This dataset is in UCI KDD repository.

References

Amirbekyan, A. & Estivill-Castro, V. (2006), Privacy preserving *DBSCAN* for vertically partitioned data, *in* 'IEEE International Conference on Intelligence and Security Informatics, ISI 2006', Springer Verlag Lecture Notes in Computer Science 3975, San Diego, CA, USA, pp. 141–153.

Amirbekyan, A. & Estivill-Castro, V. (2007*a*), The privacy of k-nn retrieval for horizontal partitioned data — new methods and applications, *in* J. Bailey & A. Fekete, eds, 'Eighteenth Australasian Database Conference (ADC2007)', Vol. 63 of *Conferences in Research and Practice in Information Technology (CRPIT)*, CORE, Australian Computer Society, Ballarat, Victoria, Australia, pp. 33–42.

Amirbekyan, A. & Estivill-Castro, V. (2007*b*), Privacy preserving regression algorithms, *in* 'The 3rd WSEAS International Symposium on Data Mining and Intelligent Information Processing, ISDM 2007', Beijing, China, pp. 37–45.

Breunig, M., Kriegel, H.-P., Ng, R. & Sander, J. (2000), Lof: Identifying density-based local outliers, *in* W. Chen, J. F. Naughton & P. Bernstein, eds, 'Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data', ACM, Dallas, Texas, USA, pp. 93–104.

Du, W. & Atallah, M. (2001), Privacy-preserving cooperative statistical analysis, *in* 'Proceedings of the 17th Annual Computer Security Applications Conference (ACSAC)', ACM SIGSAC, IEEE Computer Society, New Orleans, Louisiana, pp. 102–110.

Du, W., Han, Y.-S. & Chen, S. (2004), Privacy-preserving multivariate statistical analysis: Linear regression and classification, *in* B. M. W., U. Dayal, C. Kamath & D. Skillicorn, eds, '2004 SIAM International Conference on Data Mining', Lake Buena Vista, Florida.

Du, W. & Zhan, Z. (2002), Building decision tree classifier on private data, *in* V. Estivill-Castro & C. Clifton, eds, 'Privacy, Security and Data Mining', IEEE ICDM Workshop Proceedings, Volume 14 in the Conferences in Research and Practice in Information Technology Series, Australian Computer Society, Sydney, Australia, pp. 1–8.

Estivill-Castro, V. (2004), Private representative-based clustering for vertically partitioned data, *in* R. Baeza-Yates, J. Marroquin & E. Chávez, eds, 'Fifth Mexican International Conference on Computer science (ENC 04)', SMCC, IEEE Computer Society Press, Colima, Mexico, pp. 160–167.

Estivill-Castro, V. & Brankovic, L. (1999), Data swapping: Balancing privacy against precision in mining for logic rules., *in* 'Data Warehousing and Knowledge Discovery, First International Conference', Vol. 1676 of *Lecture Notes in Computer Science*, Springer, pp. 389–398.

Goldreich, O. (2004), *The Foundations of Cryptography*, Vol. 2, chapter General Cryptographic Protocols, Cambridge University Press.

Goldreich, O., Micali, S. & Wigderson, A. (1987), How to play any mental game (extended abstract), *in* A. Aho, ed., 'Proceedings of the 19th ACM Annual Symposium on Theory of Computing', ACM Press, New York, pp. 218–229.

Ioannidis, I., Grama, A. & Atallah, M. J. (2002), A secure protocol for computing dot-products in clustered and distributed environments., *in* '31st International Conference on Parallel Processing (ICPP)', Vancouver, BC, Canada, pp. 379–384.

Kantarcioğlu, M. & Clifton, C. (2004), Privately computing a distributed k-nn classifier, *in* J.-F. Boulicaut, ed., '8-th European Conference on Principles and Practice of Knowledge Discovery in Databases PKDD', Vol. 3202, Springer Verlag Lecture Notes in Computer Science, pp. 279–290.

Mena, J. (2003), *Investigative Data Mining for Security and Criminal Detection*, Butterworth-Heinemann, US.

Naccache, D. & Stern, J. (1998), A new public key cryptosystem based on higher residues, *in* L. Gong & M. Reiter, eds, '5th ACM Conference on Computer and Communications Security', SIGSAC, ACM Press, San Francisco, CA., pp. 59–66.

Okamoto, T. & Uchiyama, S. (1998), A new public-key cryptosystem as secure as factoring, *in* K. Nyberg, ed., 'EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques', Springer Verlag Lecture Notes in Computer Science 1403, Espoo, Finland, pp. 308–318.

Paillier, P. (1999), Public-key cryptosystems based on composite degree residuosity classes., *in* 'EUROCRYPT', Vol. 1592 of *Lecture Notes in Computer Science*, Springer, pp. 223–238.

Reingold, E., Nievergelt, J. & Deo, N. (1977), *Combinatorial Algorithms, Theory and Practice*, Prentice-Hall, Inc., Englewood Cliffs, NJ.

Shaneck, M., Kim, Y. & Kumar, V. (2006), Privacy preserving nearest neighbor search, Technical Report 06-0149, University of Minnesota, Department of Computer Science and Engineering, 4-192 EECS Building, 200 Union St SE, Minneapolis, MIN, USA. To appear in the 2006 IEEE International Workshop on Privacy Aspects of Data Mining, December 2006.

Vaidya, J. & C. Clifton, C. (2002), Privacy preserving association rule mining in vertically partitioned data, *in* 'The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', SIGKDD, ACM Press, Edmonton, Canada, pp. 639–644.

Vaidya, J. & C. Clifton, C. (2003), Privacy-preserving k-means clustering over vertically partitioned data, *in* 'Proceedings of the SIGKDD-ACM Conference of Data Mining', ACM Press, Washington, D.C., US, pp. 206–215.

Vaidya, J. & Clifton, C. (2004), Privacy-preserving outlier detection., *in* '4th IEEE International Conference on Data Mining (ICDM 2004)', IEEE Computer Society, Brighton, UK.

Vaidya, J., Clifton, C. & Zhu, M. (2006), *Privacy Preserving Data Mining*, Vol. 19 of *Advances in Information Security*, Springer, New York, NY, USA.

van der Putten, P. & van Someren, M. (2000), CoIL challenge 2000: The insurance company case., Technical Report 2000-09, Leiden Institute of Advanced Computer Science, Amsterdam.

Yao, A. (1982), Protocols for secure computation, *in* 'IEEE Symposium of Foundations of Computer Science', IEEE Computer Society, pp. 160–164.

An E-Market Framework to Determine the Strength of Business Relationships between Intelligent Agents

Khandaker Shahidul Islam
Faculty of Information Technology
University of Technology, Sydney(UTS)
NSW, Australia
islamksh@it.uts.edu.au

Abstract

When an agent enters in an e-Market for the first time, it has no historical information that can be used to determine the strength of business relationship with participant agent, and must therefore rely on the reporting of other agents to prepare for negotiation with that agents. Beliefs of individual agents change through interaction with participant agents and are reflected in their on-going relationships. An understanding of business relationships is fundamental to understanding trade between both human agents in traditional markets and software agents in electronic markets. Two parties in the market establish agreement for mutual beneficial deals or contracts and therefore execute that deal or contract. Contextual information e.g., constraints, preferences, deadlines etc., during execution of the contract are unknown to each of the parties while they established the contract. Deviations between signed contract and executed contract are observed and used to measure the strength of relationship between two parties. We have presented an E-Market framework to describe how Institution Agent can assist for mining Outcome of Contract Execution by observing Argumentation Dialogues to determine how business relationship develops and evolves. In this work, development of an argumentation system is going on where Institution Agent observes the argumentation dialogue between buyer and seller agents. The results of observation are used to determine the strength of business relationship for future interactions between buyer agent and seller agent.

Keywords: Business Relationship, Commitment, e-Market, Institution Agent, Argumentation

1 Introduction

For hundreds, if not thousands, of years trade has principally taken place between agents (merchants) who trust each other. A weak form of trust may be derived from an agent's reputation. The strongest form of trust evolves from business relationships in which two or more agents have a history of reliable trade and, perhaps, the sharing of confidential information. A basic assumption of this work is that trust between software agents will evolve similarly from the relationships between agents in electronic market places. These relationships will involve the exchange of both goods and services, and information.

A Business Relationship is evidenced by an expectation of reliable and trusted trade in the future. If Intelligent Agents are to trade effectively in an e-Market they must therefore be able to model business relationships, and must understand how those relationships strengthen and grow, and how they weaken and die.The world of E-business and Multi Agent Systems differs from traditional markets in the speed at which trade can occur, and so too in the speed at which relationships can develop.

A negotiating agent are capable of exchange proposals, evaluate proposal, and also accept or reject proposals to reach mutual deals. The agent can exchange some additional meta-level information within the messages in the form of argument to explain her current position and future plans with an intention of successful negotiation (Jennings et al. 1998). A systematic comparison of argument based and bargaining based negotiation framework (Rahwan et al. 2004*b*) shows that new information in arguments may help agents to change preferences, increase the probability to establish deals and increase the quality of deal. The authors (Rahwan et al. 2004*b*) also agreed that argumentation may lead agents to worse outcome. Different levels of Reputation (Sabater & Sierra 2002), Commitments (Norman et al. 1998, Jennings 1993), Trust (Sierra & Debenham 2005, 2006, Mui et al. 2002), and Relationships (Sierra & Debenham 2007, Ashri et al. 2003, 2005) etc. between two agents are important factors in agents internal decision making process which leads successful or failure termination of argumentation. The authors (Sierra & Debenham 2007) presented a LOGIC framework and uses *Confidence* as a generalized concept of trust, reliability and reputation measures. An agent capable of performing argument based negotiation needs to evaluate arguments and updating the mental states, generate arguments, and finally select argument (Jennings et al. 1998, Rahwan et al. 2004*a*).

Repeated contract establishment and the outcomes of contract execution should have a major effects

directly or indirectly on existing relationships. In our work, contract is composed of a pair of *commitments* between two agents. Intelligent agent involves in e-Market exchange information with other agents through argumentation to establish, modify or sustain contract and relationship develops or evolves between agents due to the result of contract execution especially the deviation between signed contract and executed contract. Hence, we could measure the expectation to establish future contract between two parties in similar dimension by analyzing the outcome of contract execution. The motivation of this work is to measure the strength of business relationships in the e-Market populated by Intelligent Agent.
The following section provide an overview of business relationship with definitions of some important terms used in our work. Section 3 describe the overall e-Market framework for agents to build business relationships and also discusses the issues related to argumentations between agents in an e-market. Section 4 discusses about determining the strength of business relationship showing the need of Outcome database. Next section introduces the evolution of Business Relationship in e-Market. An overall discussion containing concluding remarks are given in Section 6.

2 Background

In the market, a potential number of buyers and sellers are involved in negotiation to buy and sell goods and services. Negotiation is as much of information acquisition and exchange process as it is an offer exchange process- one feeds off the other Debenham (2004*a*).There are some frequent parties in a market who are trying to make deals repeatedly with a minimum amount of effort/time spent by them for negotiation. The history of interactions between different parties plays an important role to develop relationships between them which will simplify negotiation dialogues. Such interaction history involving mutual dealings between two people or parties or institutions are the foundation of any relationship.
Business relationship sometime becomes a never ending relationship, where as sometime relationship might be break down due to some special circumstances. Different types of relationships through interaction in multi-agent systems e.g., dependency, competition, collaboration etc. are identified in Ashri et al. (2003). Customer provider relationship is the expectation of an infinite number of future interactions (or at least the inability to know when the last interaction will occur) that induces customers and providers to cooperate for their mutual gain Schultze (2003). An understanding developed between two parties to provide regular business services for their mutual gain is the basis to build Business Relationship. In the e-Market of Intelligent agents, relationships between agents are developed mainly based on their interaction history especially on the outcome of contract executions.
Contract,C is a pair of commitments $\{(C_\alpha, C_\beta)\}$ between two agents, α and β such that both parties agreed to fulfill their part of commitment in some specific or open time after contract establishment.
Contract Execution, C_E is the enactment of pair of commitments by two parties, where commitments pair are the elements of Contract,C and was enacted with or without any deviations by the parties after Contract Establishment.
Outcome of Contract Execution is an object defined as the evaluation made by Institution Agent on a certain pair of Commitments contained in a contract at the execution time.
Business Relationship between two agents in e-Market is defined as a set of all the historical outcomes of contract executions between two agents from which expectation of future *Contract Execution* can be derived to establish deals or contracts between them.
Strength of Business Relationship in a dimension between two agents is the probability or expectation of the outcomes of contract execution in that dimension and this expectation is derived from historical outcomes of contract execution contained in Business Relationships between two agents.
At any moment of time, t_i relationship between two parties is a set of outcomes represented by $r^{t_i}(\alpha, \beta)$. The strength of relationship in a given dimension, d_j between two parties at time, t_i is the probability or expectation of future contract(φ') execution from a given signed contract(φ) in the dimension, d_j, i.e., $P^{t_i}(\alpha, \beta, \varphi[d_j]) \in [0, 1]$ which means α's estimation of the strength of relationship between the pair of parties (α, β) in the dimension, d_j at time, t_i derived from $r^{t_i}_{d_j}(\alpha, \beta)$.

3 An E-Market Framework to Build Business Relationships

In the e-market, agents negotiate to fulfill their need and try to *establish* long term relationship to simplify the process of fulfilling the need. The result of a successful negotiation is the contract signed by two parties which will be executed by the both parties for a specified period in future. There may exist significant deviations meaningful to both parties due to some unknown variables during the contract establishment time. In order to model business relationship, agents use the outcome of executed contract in order to *modify* signed contract if any issue arises or *sustain* on-going or future contracts.

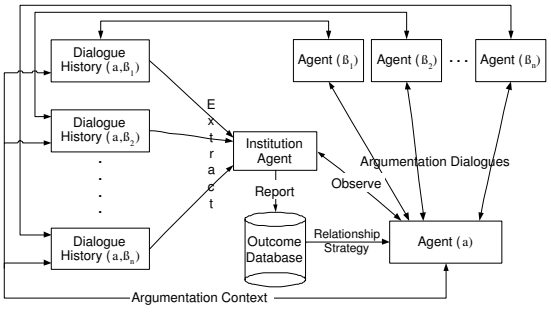


Figure 1: An e-Market Framework to Build Business Relationships

Institution agent observes the execution of contract. Buyer agent or seller agent who counts business relationships will use the result of observation reported in Outcome Database as shown in Figure

1 and use a systematic the steps in our methodology. In our prototype system, we have introduced an institutional agents, who facilitate buyer or seller agent to observe what is happening in the electronic market. We are further developing our system to cover all steps of our methodology. The outcome of our completed work will give us an experimental tool to extend buyer and seller agent to accommodate different algorithm or procedures to study business relationships. Conducting some experiment using our developed prototype system are included in our future research plan. We have introduced Institution Agent to observe the activities between two parties during the execution of the contract and the outcome of observation can be used to measure the strength of business relationship. We are dealing the observation into three phases: Establish Contract, Modify Contract, and Sustain Contract.

- **Establish Contract**
  This phase deals with observing the argumentation between buyer and seller agent and after the agents reach agreement, the Institution Agent will extract the agreed deal from current argumentation dialogue and report *Signed Contract* to both agents. In our work, we describe a *deal* as an aggregate object of an Item and a Free Item i.e., $Item[quantity, price] + FreeItem[quantity, value]$. For example, $Banana[10K, \$20] + Discount[1, 10\%]$ is a deal represents 10% discount on the price \$20 for 10K Banana.

- **Modify Contract**
  Institution Agent receives information during execution time from both agents, and fetch information regarding signed contract and assess the changed condition, and report *Modified Contract* back to the concerned participants. As a result, future interaction between pair of agents will be successful. Free item serves a negotiation stance to modify deals. If buyer agent prefers delivery or some free item, both parties can find some alternate deals by modifying previously signed deal, e.g., $Banana[10K, \$20, quality = Top] + Nothing[1, 0]$ with deals either $Banana[10K, \$20, quality = Medium] + Delivery[1, \$5]$ or $Banana[10K, \$20, quality = Top] + Delivery[1, \$5, DeliveryTime = 2\ days]$ during execution time.

- **Sustain Contract**
  Argumentative illocutions e.g., reward, threat or appeal gives wider options to agents for contract execution to maintain minimum deviation with signed contract. Agent use argumentative illocutions to exchange execution time information so that agents can sustain an on-going contract. Sacrifice in one deal and get reward in future deals will reduce argumentation break down. If buyer demands delivery of a Top quality Item, but seller offers delivery with medium quality. Here buyer can declare reward for providing top quality, by passing some private information as a result seller agree to provide free delivery. $reward(deal, info)$, where $deal = Banana[10K, \$20, quality = Top] + Delivery[1, \$5]$ and $info = Need(Apple[10K, next\ \ week])$. Seller agent can appeal to give Free Pineapple instead of discount. $appeal(deal, info)$, where $deal = Banana[10K, \$20, quality = Top] + Pineapple[1, \$5]$ and $info = Delivery[1, \$5, DeliveryTime = 2\ days\ |delivery\ van = full]$. Sharing private information during contract execution time allows agent to sustain contract.

In this work, we have given concentration mainly on contract establishment. The evolution of relationship will be occurred in two steps: global evolution of the environment i.e., sharing public information and local evolution inside individual agents i.e., sharing private information. Our methodology consists of the following steps and the whole conceptual framework is presented in Figure 2.

- Select a Strategic Moves for argumentation between buyer and seller agents. e.g., Seller will give 10% discount on some items.
- Observing Argumentation Dialogue between two Agents over a period of time
  - Observe the argumentation for contract establishment between agents
  - Institution Agent observes the contract execution between agents
- Calculate the deviation between signed contract and executed contract
- Institution Agent reports deviation to concerned parties
- Search historical dialogues about the contract execution between two parties for the selected and/or similar Strategic Moves
- Based on the deviation on historical data and current contract execution, estimate important parameters that affect the relationships
- Analyze the evolution of Relationship

3.1 Illocution and Language

The illocutionary particles used in Sierra & Debenham (2006) are Offer, Accept, Reject, Withdraw, Inform, Reward, Threat, Appeal. To implement argumentation context mining system, we have added few more illocutionary particles *query, sold, paid, bye, others* etc. having simple semantic meanings and syntax. We have used two illocutionary particles *sold, paid*, which are different from other illocutions because these illocutions are used by Institution Agent for analyzing deviation between utterance and subsequence execution. We are also using a simple content language (info $\in L$) using ProLog like syntax for internal representation of propositional content contained within illocution that both agents have agreed to use. For simplicity of the system, we assumed that both parties have sufficient capacity to communicate with each other using this language. Message contains the vocabularies from defined ontology and deal object. Deal object is a aggregate object derived from objects in item and free item ontology.

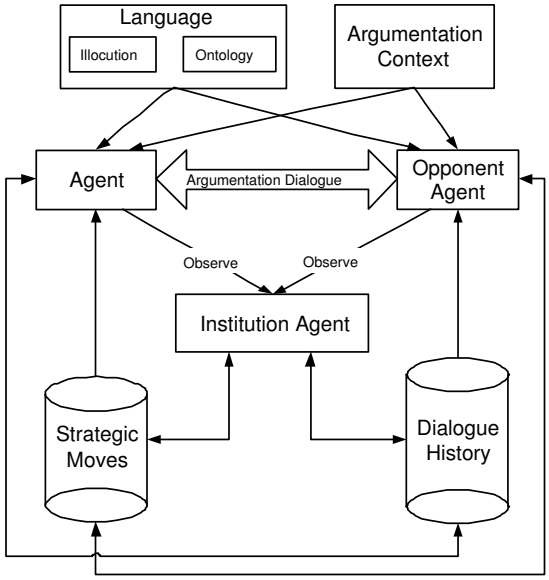


Figure 2: Conceptual Framework of an Argumentation System between Intelligent Agents

In this work, we are using the illocution particles with the following syntax and meaning adopted from Sierra & Debenham (2006).

- *inform(Need(deal))*. Buyer agent informs seller agent about buyer's interest to make a deal or sign a contract in line with some Strategic Moves.
- *offer(deal)*. Seller agent offers a deal to buyer agent which may differ from buyer agent's expected deals in line with some Strategic Moves.
- *accept(deal,[info])*. Buyer agent accept seller agent's previously offered deal. Buyer agents may include additional feedback information to seller agent. The feedback contains positive or negative impression on seller agent's offer.
- *query(deal,[info])*. Any agents ask question to the opponent agent to explain a previous deal. Agents may include additional feedback information to another agent. The feedback contains positive or negative impression on previous deal.
- *reject(deal,[info])*. Buyer agent reject seller agent's previously offered deal. Similar to accept, buyer agents may also include additional feedback information to seller agent. The feedback generally contains negative impression on seller agent's offer.
- *withdraw(deal,[info])*. Agent break down negotiation. Agent may also include additional feedback information, which generally contain negative impression on previous offer.
- *reward(deal,[info])*. Intended to make the opponent accept a proposal with the promise of additional free item as complements. Optionally, additional information in support of the deal can be given.
- *threat(deal,[info])*. Intended to make the opponent accept a proposal by committing some activities which the opponent does not desire. Optionally, additional information in support of the deal can be given.
- *appeal(deal,[info])*. Intended to make the opponent accept a proposal as a consequence of change in belief that the accompanying information might bring about. Agent passes additional information in support of a deal. Appeal can be understood as a combination of an offer and an inform.
- *sold(Item)*. If the previous illocution is accept, the seller agent physically send items and inform buyer by sold illocution.
- *paid(Item)*. After receiving the item, buyer agent inform seller agent what buyer agent have paid. It is also an acknowledgement message.
- *bye()*. The last message by both participants in the e-market is bye(), whether or not the deal is successfully executed.

3.2 Ontology

To interact agents in an electronic market, we need an ontology (Kalfoglou & Schorlemmer 2003) representing the set of concepts, classes, relations, and functions. Two basic ontologies: Item Ontology and FreeItem Ontology are used for defining deal object $deal = Item[issues] + FreeItem[issues]$ and argumentation dialogue using illocution and ontology allows agents to establish, modify and sustain deals.

Item Ontology is defined by its vocabulary, concepts and relationships Vocabulary of ItemOntology={Item, Fruit, Vegetable, Apple, Banana, Tomato, Potato}
Item(Name,Type)
Fruit(Name)
Vegetable(Name)
isa(Banana,Fruit)
isa(Apple,Fruit)
isa(Tomato,Vegetable)
isa(Potato,Vegetable)

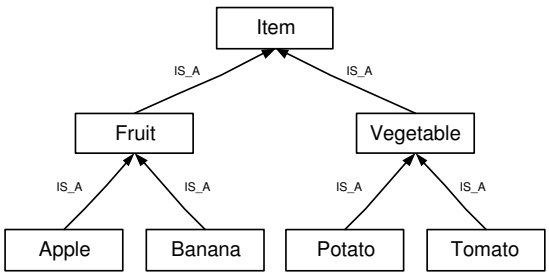


Figure 3: Item Ontology

FreeItem Ontology is defined by its vocabulary, concepts and relationships
Vocabulary of FreeItemOntology={Discount, Delivery, Coupon, Pineapple, Movie, Nothing}

Discount(ItemName, Value)
Delivery(ItemName, Value)
Coupon(ItemName, Value)
Movie(ItemName,Value)
Pineapple(_,Value)
Nothing(_,Value)

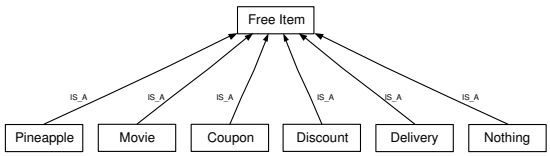


Figure 4: Free Item Ontology

We also measure semantic distance which refers to the notion of relative or useful distance between concepts across the ontology. There observed some deviations between agreed deal and executed deal in some dimensions. So, we need to measure deviation for argument evaluation and decision making process. We use semantic distance between two concepts in Item Ontology on the path length over the ontology tree and the depth of the subsumed concepts on the shortest path between the two concepts (Roddick et al. 2003).

- According to Roddick et al. (2003), if l is the shortest path between the concepts, h is the depth of the deepest concept subsuming both concepts, and k_1, and k_2 are parameters scaling the contribution of shortest path length and depth length respectively, similarity between two concepts c_1 and c_2 are defined as,

$$Sim(c_1, c_2) = e^{-k_1 l} . \frac{(e^{k_2 h} - e^{-k_2 h})}{(e^{k_2 h} + e^{-k_2 h})}.$$

  Function for Semantic Distance between two items in Item Ontology is approximately represents the following information:
  $Sim(c_1, c_2)) =$
  $\begin{cases} 0, & c_1.name = c_2.name \\ 1, & c_1.type = c_2.type \wedge c_1.name \neq c_2.name \\ 2, & c_1.type \neq c_2.type \wedge c_1.name \neq c_2.name \end{cases}$
  For example,
  Sim(Apple,Banana)=1 and
  Sim(Potato,Banana)=2

- We have also used another simple function to estimate semantic distance between two free items. Function for Semantic Distance between two items in FreeItem Ontology is defined as the difference of values of two free items. For Example,
  Sim(Discount(Item,20),Delivery(Item,15))=5
  and,
  Sim(Discount(Item,20),Movie(_,10))=10
  The value of an FreeItem will be determined by seller agent and honest reporting of the value of FreeItem is assumed in this work.

3.3 Argumentation Context

Context (Sierra & Debenham 2007) represents previous agreements, previous illocutions, ontological working context, institution norms, and some external parameters that have direct or indirect affect on agents current argumentation. Argumentation Context should contains a subset of historical dialogues which influence the target of current dialogue in such a way that agent can resolve some conflict or achieve some critical goals or reduce the risks of uncertainty or produce some conflict. Agent can extract some candidate arguments or issues in current negotiation threads through contextual analysis. Contextual analysis helps agent in generating and sequencing alternative goals in order to reach mutual decision in bargaining. Context may be a simple form of representation of bindings of issue-value pairs in line with current dialogue. Extraction of relevant issues with their value in real-time in electronic market is a complex research problem. The values in each issues will be revised using initial value, decay limit distribution function when no further information is received for a given time period. Once, some information arrives the values in each issues will be revised using posterior distribution function provided that the arrived information has a significant impact on future dialogues. We can construct a context tree or graph from the historical data from each agent. While the execution proceeds, due to some new contextual information, the context tree or graph will evolve to reflect the contextual information. Each agent will maintain contextual reflection of its own. For simplicity, we refers context as a set of beliefs that modifies agent's default actions during the offer generation or offer evaluation time. Context might have negative, positive or no effect, and detail investigation on Context Monitor, Context Network and Context Miner to obtain Processed Context from Raw Context is going on.

- In an argumentation dialogue, seller agent identifies contextual information *Buyer agent likes pineapple very much*, i.e., $(\alpha, Likes(Pineapple, Maximum))$. If seller agent offer free pineapple rather than discount in a rejected deal, the buyer agent may positively evaluate the offer and accept the deal containing free *Pineapple*.

- In an another argumentation dialogue, extracted context *Buyer agent prefers discount over delivery*, i.e.,$(\alpha, Prefers(Discount[\omega_1], Delivery[\omega_2]))$. If seller agent offers delivery of higher value than discount, but the buyer agent may not accept that offer.

- Agents private information, e.g., *Seller agent has limited stock of Tomato*, i.e., $(\alpha, HasStock(Tomato, Quantity))$ may be the reason for rejecting offer. Seller agent may wish to sell his stock of Tomato to other buyer agents instead of giving discount to current buyer agent.

- Uncertain contextual information *Seller agent believes that buyer agent like pineapple very much*, i.e., $(\beta, Believes(\alpha, Likes(Pineapple, Maximum)))$ can be used during argumentation. Seller agent may offer free pineapple having value lower than discount value and wait for buyer agent's response.

3.4 Interaction Protocol

An interaction protocol is defined by an environment and an interaction diagram. A protocol (Jennings et al. 2001) is a formal set of conventions governing the interaction among participants. The argumentation protocol specify at each stage of argumentation process, agent is allowed to say which argumentation illocutions. The interaction proposal might be based on last utterance or depend on a more complex history of messages between agents (Rahwan et al. 2004*a*). We have defined an argumentation protocol discussed below and the flowchart is shown in Figure 5.

- Initial Setup: Buyer start with *Inform*, and Seller responses with *Offer*
- Argumentation: Buyer or seller use AIR(Accept, Inform, Reject) and ART(Appeal, Reward, Threat) in argumentation.
- Acceptable Deal:Buyer use Accept
- Termination: Anyone use Withdraw to terminate

Elements of an abstract model for argumentation agent is explained inRahwan et al. (2004*a*). In our work, the argumentation phases: Incoming Argument Interpretation, Argument Generation, Argument Selection, and Outgoing Locution Generation have been implemented. We categorized AIR(Accept, Inform, Reject) as Soft Argumentative Illocutions and ART(Appeal, Reward, Threat) as Strong Argumentive Illocution. The simple strategy is used to select an illocution from ART such that opponent will accept, if not inform, and else reject the deal.

- reward: Receiving agent will evaluate reward as some additional profit for accepting the current deal.
- threat: Receiving agent will evaluate threat as some additional loss for rejecting the current deal.
- appeal: Receiving agent will evaluate appeal as some additional information to accept the current deal.
- inform: Receiving agent will evaluate inform as an alternative proposal to the current deal.

During the entire process of argument interpretation, generation, and selection, we are proposing to use the following three categories of contextual information extracted from the history of messages.

- Illocution history provides reward, threat, offers etc that occurred in previous deals.
- Ontological search from history helps us to find out deals on Fruit, Vegetables or any other ontological categories
- Semantic Distance search on dialogue history provides use similar deals e.g., apple or similar items. Potato or similar items in the history

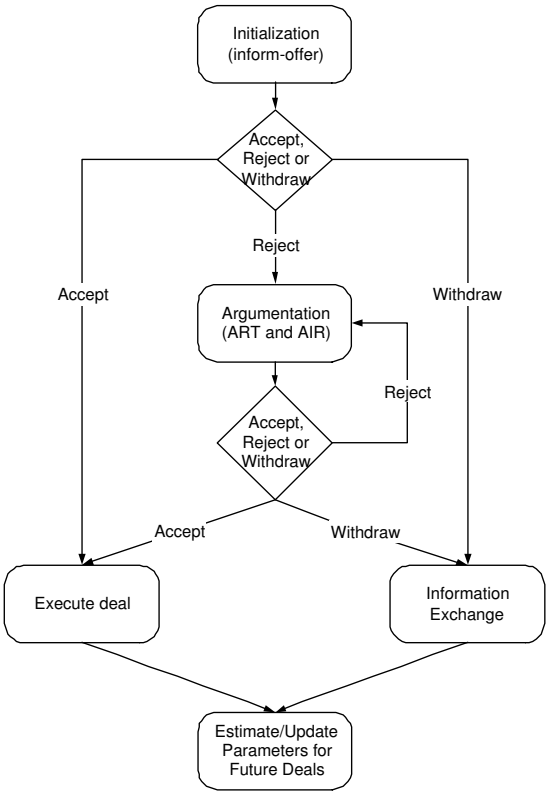


Figure 5: Interaction Protocol

4 Agent Build Business Relationship

Argumentation assist agents to establish deals and build Business Relationships. When agent, α has Need(X), instead of sending message, $\mu(inform, Need(X))$ to a specific partner β_i, agent, α can choose a partner from a set of partners $\{\beta_i | i = 1..n\}$. The problem is how agent, α choose partner agent, β_i from set $\{\beta_i | i = 1..n\}$. The *Confidence(.)* and *Strength(.)* regarding agent's *Need(X)* extracted from entire interaction histories between $\{(\alpha, \beta_1), (\alpha, \beta_2), (\alpha, \beta_3), ...(\alpha, \beta_n)\}$ are used to decide which partner is best to meet current need.
Two parties is an agreement are entered into obligations to supply and to pay for supply. The observer(IA) can determine which dimensions are achieved from the post argumentive illocutions. e.g., sold(quantity=9 Kilo, price \$10/Kilo, delivery day=monday, quality=best). and pay(amount=\$90, payment date=1 day late, quality=average). Institution agent can see argumentation dialogues from interaction history and extract obligation of each parties and see which obligation is not fulfilled by them in post argumentative dialogues and maintain Outcome object and finally, prepare summary measure e.g., reliability of an agent, successful argumentation rate, withdraw rate, commitment fulfillment rate, deviation from agreed value etc. and report it to interacting parties. We assume that IA has such capability.

4.1 Outcome Database

We presented an *outcome* as a tuple of the form $(\alpha, \beta, C_{ID}, deal_{expected}, deal_{actual}, D_{amount}, t, \Delta)$ where a,b are agents, C_{ID} refers to the contract Identifier of the contract with which the current deal relates, expected items ($deal_{expected}$) is a's expected list of items for the deal based on signed contract, actual items ($deal_{actual}$) is the items that actually received by a from b for the executed deal, deal amount(D_{amount}) represents the *formal value* of the deal that occur between two agents, and deviation(Δ) represents the a's estimation of the difference between the formal values of expected items and actual items. In this experiment, *deal* contains any item from Item Ontology and FreeItem Ontology. Similar to Impression database used in REGRET Sabater & Sierra (2002), we have introduced Outcome Database, ODB containing the set of all historical outcomes evaluated by Institution Agents. ODB is used for estimating the strength of relationships between agents. An agent's outcome database $ODB^a \subseteq ODB$ is a set of outcomes containing deals signed by agent, a with some other partner agent.
We define $ODB^a_{item} \subseteq ODB^a$ as the set of outcomes in ODB^a such that $item \in deal_{expected}$ where general form of an outcome in ODB^a_{item} is $(a, \_, \_, \{.., item, ..\}, \_, \_, \_, \_)$. We define $ODB^{a,b} \subseteq ODB^a$ as the set of outcomes in ODB^a where the outcomes is the results of contract executions between agents a, and b. We further define $ODB^{a,b}_{item} \subseteq ODB^{a,b}$ as the set of outcomes in $ODB^{a,b}$ where the outcomes is the results of contract executions between agents a, and b and also $item \in deal_{expected}$ where general from of an outcome in $ODB^{a,b}_{item}$ is $(a, b, \_, \{.., item, ..\}, \_, \_, \_, \_)$. The set $ODB^{a,b}_{item}$ is also a subset or equals to ODB^a_{item} i.e., $ODB^{a,b}_{item} \subseteq ODB^a_{item}$.
Let us consider an example, agent a wants to buy Tomato from agent b. Agent a will consult the $ODB^{a,b}_{Tomato}$ for estimating the strength of Relationship. Any outcomes having $\Delta > \epsilon$ are treated as negative effect on the strength of relationships, whereas $\Delta \leq 0$ are counted as positive effect on the strength of relationships. $ODB^{a,b}_{Tomato}$ will *directly* give us the strength of relationship to make future deals for Tomato. In the absence of sufficient historical outcome in a dimension, *indirect* strength of relationships to make future deals for Tomato using items in Item Ontology other than Tomato using the database $ODB^{a,b}$ can also be measured. The *indirect* strength of relationship between a and b to make future deal for Tomato are measured by using the outcome database, $ODB^{a,b}_{item \neq Tomato}$. Varying the ontological categories and threshold size for semantic distance, agents can show variable flexibility to measure the expectation in future deals.

4.2 Strength of Business Relationship

We have two separate histories: Dialogue History and Outcome Database. Dialogue History contains historical dialogues. Outcome Database is proposed to contains outcome of executed contract especially the deviation between signed contract and executed contract. Institution Agent provide timely information to Outcome Database. In fact, Outcome Database contains summary information extracted by Institution Agent from Dialogue Histories. Structure of Outcome Database is important for measuring trust, honour, reputation and then Confidence on partner agent.
Candidate Partners=Agent has *Confidence(.)* on participant agents to fulfill *Need(X)*.
Negotiation Partner=Select One *Candidate* agent having maximum value of the strength of Business Relationship to fulfill *Need(X)*
Outcome database (ODB) are used for summary measure for choosing possible set of candidate negotiation partners using,

$$Candidate(\alpha, Need(X)) =$$

$$\{\beta_i | \forall_{\varphi \leq Need(X)} Confidence(\alpha, \beta_i, \varphi) > T_c\}$$

$$Negotiator(\alpha, \rho) = arg\ max_i \{Strength(\alpha, \beta_i,$$

$$\{ODB^{\alpha,\beta_i}_{item \leq \rho}\}) | \beta_i \in Candidate(\alpha, Need(\rho))\}$$

and Interaction Histories(more than one agents history) will assist us to select one partner to negotiation based on agents internal world model.
Deviation between signed deal and executed deal is represented as,

$$\Delta = V(deal_{signed}) - V^t(deal_{observed})$$

We need a function to transform the deviation between signed deal and observed deal into number between [0,1] such that less deviation means transformed value is close to 1 and more deviation means transformed value is close to 0. One simple approximation is given by where, $f(ODB^{\alpha,\beta_i}_{\varphi}.\Delta) = e^{-\Delta/\lambda}$. Institution agent will keep track on $deal_{observed}$ and any observation is updated to ODB. Agent uses ODB to estimate the Strength(.) of relationship with other agents in some dimension.

$$Strength(\alpha, \beta_i, \varphi) =$$

$$\frac{\sum_{\Delta \leq 0 : ODB^{\alpha,\beta_i}_{\varphi}} P^t_{\beta_i}(\varphi \in deal_{signed})}{\sum_{deal_{signed} : ODB^{\alpha,\beta_i}_{\varphi}} P^t_{\beta_i}(\varphi \in deal_{signed})} +$$

$$\frac{\sum_{0 < \Delta \leq \epsilon : ODB^{\alpha,\beta_i}_{\varphi}} P^t_{\beta_i}(\varphi \in deal_{signed}).f(ODB^{\alpha,\beta_i}_{\varphi}.\Delta)}{\sum_{deal_{signed} : ODB^{\alpha,\beta_i}_{\varphi}} P^t_{\beta_i}(\varphi \in deal_{signed})}$$

This equation measures the Strength to fulfill the need of a single item. Aggregating the values over a class of items e.g., those φ that belongs to ontology, ρ.

$$Strength(\alpha, \beta_i, \rho) =$$

$$\frac{\sum_{\varphi : \varphi \leq \rho} P^t_{\beta_i}(\varphi).Strength(\alpha, \beta_i, \varphi)}{\sum_{\varphi : \varphi \leq \rho} P^t_{\beta_i}(\varphi)}$$

Similarly, α's overall estimate of β_i's Strength is

$$Strength(\alpha, \beta_i) = \sum_{\rho} P^t_{\beta_i}(\rho).Strength(\alpha, \beta_i, \rho)$$

For example, if α want to buy 10K of Banana, α will compute $Strength(\alpha, \beta_i, Banana[10K])$ for agent β_i selecting from those agent having $Confidence(\alpha, \beta_i, Banana[10K])$ greater than a threshold value, T_c. In Sierra & Debenham (2006), $Build(\alpha, \beta, \rho)$ means "agent α considers agent β to be a potential trading partner for deals in a relationship ρ" and agent estimates probabilities that are attached to $P(Build(\alpha, \beta, \rho))$ representing the certainty that it has in this proposition. $Strength(\alpha, \beta_i, \rho)$ examines the Outcome Database, i.e., dialogue history of accepted or offered deals: exact item-ontology-semantic distance-overall, and measure agents capability to execute a deal in a dimension. According to Sierra & Debenham (2007), we can estimate $Confidence(\alpha, \beta_i, \rho)$ by examining the dialogue history of $reward; accept, threat; \sim accept, inform$ etc. from accepted, rejected, or withdrawn dialogues as well as ongoing dialogues starting from exact item-ontology-semantic distance-overall, etc. Agent α can estimate what happened and then estimate probability to build relationship as a measure of the strength of Business Relationship.

$$P(Build(\alpha, \beta_i, \rho)) = Strength(\alpha, \beta_i, \rho)$$

E.g., $P(Build(\alpha, \beta_i, Banana[10K])) =$

$$Strength(\alpha, \beta_i, \{ODB^{\alpha,\beta_i}_{item=Banana}.deal_{signed}|$$

$$deal_{signed}.item.Quantity \in [9K - 11K]\})$$

If the above deviation is too small to return a suitable partner, α may be flexible to select partner shown below

$$P(Build(\alpha, \beta_i, Banana[10K])) =$$

$$Strength(\alpha, \beta_i, \{ODB^{\alpha,\beta_i}_{item\leq Fruit}.deal_{signed}|$$

$$deal_{signed}.item.Quantity \in [5K - 15K]\})$$

In other words, if agent, α can select a suitable partner from historical trades of any types of Fruit, α may use the following equation

$$P(Build(\alpha, \beta_i, Fruit)) =$$

$$Strength(\alpha, \beta_i, \{ODB^{\alpha,\beta_i}_{item\leq Fruit}.deal_{signed}\})$$

Therefore, agent, α can select a suitable partner from historical trades of any ontological category, ρ, using the following equation

$$P(Build(\alpha, \beta_i, \rho)) =$$

$$Strength(\alpha, \beta_i, \{ODB^{\alpha,\beta_i}_{item\leq \rho}.deal_{signed}\})$$

In general, agent, α can select a suitable partner from historical trades of any items using the following equation

$$P(Build(\alpha, \beta_i)) =$$

$$Strength(\alpha, \beta_i, \{ODB^{\alpha,\beta_i}_{all\ item}.deal_{signed}\})$$

Using our developed system, we have plan to experiment by measuring the outcome of contracts. We observed that agents can categorize the outcome into three groups: positive, neutral and negative. We stored the outcome into our outcome database for further experiments. We are measuring the differences between signed contract and execution contract for the following three different Strategic Moves.

- If you spend \$200 this month, I will give you 10% discount next month
- If you spend \$200 this month, I will give you 10% discount next month and Context=Buyer prefer pineapple over 10% Discount, i.e., $(\alpha, Prefers(Pineapple, Discount[10\%]))$
- If you spend \$200 this month, I will give you 10% discount next month and Context=Buyer prefer Delivery over 10% Discount, i.e., $(\alpha, Prefers(Delivery, Discount[10\%]))$

We have developed a prototype system to implement an e-market where agents give values to relationship. Initially a buyer agent and a seller agent is joined in the market and we measure strength of relationship for different items across an ontology. We are developing the system using Java. The entire agent architecture developed so far is shown in Figure 6

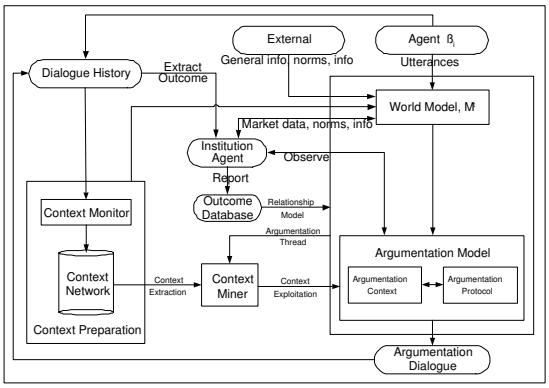


Figure 6: Agent Architecture to Build Business Relationships

5 Introduction Evolution of Business Relationships

Relationship between two parties will evolve depending on the outcome of execution of contracts, some special events in e-Market or group/social influence etc. Any change in relationship between two parties may introduce subsequent change in relationships among other parties in e-Market. Similar to business network in Debenham (2004*b*), relationship network will develop in e-Market which will be basically evolving nature. The evolution of business relationship depending on three broad categories are explained below. This work has mainly focused on the first category.

Evolution based on Outcome of Contract Execution

Relationship between agents are dynamic over time which may evolve after the outcome of any contract execution. Agent may find some commitments not achievable Jennings (1993), which may be also observed in the e-market. If the outcome is successful then relationship will increase. The positive deviation and negative deviation during the enactment of commitment does not always neutralize the effect of them in relationship. Furthermore, two same deviations may have different value on enactment of commitment depending on the contextual information. After several rounds of failure in outcome, if buyer agent estimates the strength of relationship with a seller agent which is less than minimum relationship strength (threshold), the buyer agent can choose another buyer available in the e-Market in order to get better deals.
Therefore, we have to investigate nature of deviation during the execution of contract. We see that only mean value of the outcome of contract are not sufficient to overcome all problems. We can introduce few more statistical parameters, e.g., standard deviation, frequency distribution, maximum deviation, etc. to minimize such problems. Introduction of such parameters will reduce the effect, but does not give us complete structure to evaluate an execution of contract. We are developing the system where we can include functions to estimate different parameters. On the other hand we can further develop the system for using as an efficient tools for simulation.

Evolution based on Events in e-Market

Any events in e-Market give valuable or additional information to agents for deciding future plans. Agents want to observe events in the e-Market especially what other agents actually do. To simplify the observation, we have introduced Institution Agent to observe the events in the market. Some events in e-Market will affect agent's knowledge and trust on it's beliefs. As a result, indirect strength of relationships will evolve after an event occurred which directly or indirectly affect the relationship between two parties. How agents will perform their belief revision process is their internal mechanism. We are dealing with providing an environment for agents to evolve their relationships based on events. In our study of Business Relationships, we only consider those types of events relating the deviation between signed contract and executed contract.

Evolution based on Group or Social Influence

Agents sharing common goals or environments can make groups which will lead them to obtain better benefit from e-Market rather working individually. Any change of relationship between two parties may influence other parties in the same group to update their relationship values using reflexive, transitive rules and as a result network of relationship develops. In our future works, we will investigate for such evolution in details.

Reflexive update: If $r^{a,b}(t_i)$ changes at t_{i+1}, then there is a possibility that $r^{b,a}(t_i)$ may change at t_{i+1} or later.

Transitive Update: If $r^{a,b}(t_i)$ and $r^{b,c}(t_i)$ changes at t_{i+1}, then there is a possibility that $r^{a,c}(t_i)$ may change at t_{i+1} or later.

Network of Relationship: It will be a directed graph and the edge value represents the strength of relationship. No edge means no relationship.

6 Discussion

Managing Relationships in a traditional business is difficult due for the time and cost requirements to communicate updated information, searching new buyers and sellers and evolving existing relationships. E-business using Multi Agent System could make it easier to attract new parties and increase benefits to all involved parties and study relationships among parties. If the customer receives desired commodity from a seller in some previous transactions then the customer would become satisfied on that particular seller and the reputation will increase as a result their relationship will be strengthen. However, after several round of desired outcome received by a buyer, if he receives a commodity which is not as desired level then reputation may decrease but their relationship will not break down immediately. In these situations, if seller tries to replace, refund or any other action on which the customer become happy, then their relationship will be stronger instead of break down. In the marketplace, there should be a set of agents engaged for buying with common interest. There will be a large number of buyer agents, but for the purpose of observing contract execution, we selected one buyer agent and one seller. These two agents negotiate for a specified period of time to execute a number of deals for a set of commodity available in the marketplace within the guideline of signed contract. Deviation between executed deals and signed deals in the contract is measured and this deviation is used to evolve relationship between buyer and seller agents. In some deals, there may arise situations were one party fails to meet standard requirements of the contract terms at least one dimension, e.g., delay delivery, poor quality product, delay in payment, etc. We will further investigate the multi-issue outcome. A set of Strategic Moves has been selected for experiments. We are continuously improving our system to increase functionality and agents capability and we found that the conceptual framework is proving itself as a useful research issue in this field.

References

Ashri, R., Luck, M. & d'Inverno, M. (2003), On Identifying and Managing Relationships in Multi-Agent Systems, *in* 'Proceedings of Eighteenth International Joint Conference on Artificial Intelligentce, Acapulco, Maxico'.

Ashri, R., Ramchurn, S. D., Sabater, J., Luck, M. & Jennings, N. R. (2005), Trust evaluation through relationship analysis, *in* 'Proceedings of the Fourth International Joint Conf on

Autonomous Agents and Multi-Agent Systems, Utrecht, Netherlands', pp. 1005–1011.

Debenham, J. (2004*a*), Bargaining with Information, *in* 'Proceedings of Third International Conference on Autonomous Agents and Multi Agent Systems AAMAS 2004', pp. 664–671.

Debenham, J. (2004*b*), Interacting with Electronic Institution, *in* 'Proceedings of Fifteenth International Conference on Database and Expert Systems Applications DEXA 2004, Zaragoza, Spain', pp. 181–190.

Jennings, N. R. (1993), 'Commitments and conventions: The foundation of coordination in multi-agent systems', *The Knowledge Engineering Review* **3**, 223–250.

Jennings, N. R., Faratin, P., Lomuscio, A. R., Parsons, S., Sierra, C. & Wooldridge, M. (2001), 'Automated negotiation: prospects, methods and challenges', *International Journal of Group Decision and Negotiation* **10**(2), 199–215.

Jennings, N. R., Parsons, S., Noriega, P. & Sierra, C. (1998), On Argumentation-Based Negotiation, *in* 'Proceedings of International Workshop on Multi Agent Systems 1998, Boston, USA'.

Kalfoglou, Y. & Schorlemmer, M. (2003), IF-Map: An Ontology-mapping method based on information-flow theory, *in* 'Journal on Data Semantics I, S. Spaccapietra, S. March, and K. Aberer, Eds., vol. 2800 of Lecture Notes in Computer Science', Springer-Verlag, Heidelberg, Germany, pp. 98–127.

Mui, L., Mohtashemi, M. & Halberstadt, A. (2002), A Computational Model of Trust and Reputation, *in* 'The Proceedings of the 35th Annual Hawaii Conference on Systems Sciences'.

Norman, T. J., Sierra, C. & Jennings, N. R. (1998), Rights and Commitment in multi-agent agreements, *in* 'Proceedings of International Conference on Multi Agent Systems', pp. 222–229.

Rahwan, I., Ramchurn, S. D., Jennings, N. R., McBurney, P., Parsons, S. & Sonenberg, L. (2004*a*), 'Argumentation Based Negotiation', *Knowledge Engineering Review* .

Rahwan, I., Sonenberg, L. & McBurney, P. (2004*b*), Bargaining and argument-based negotiation: Some preliminary comparisons., *in* 'Proceedings of the AAMAS Workshop on Argumentation in Multi-Agent Systems', New York.

Roddick, J. F., Hornsby, K. & de Vries, D. (2003), A Unifying Semantic Distance Model for Determining the Similarity of Attribute Values, *in* 'Proceedings of Twenty-Sixth Australasia Computer Science Conference, ACSC 2003, Adelaide, Australia', pp. 111–118.

Sabater, J. & Sierra, C. (2002), REGRET: A reputation model for gregarious societies, *in* 'Proceedings of First International Joint Conference on Autonomous Agents and Multi-Agent Systems', pp. 475–482.

Schultze, U. (2003), 'Complementing self-serve technology with service relationships: The customer perspective', *e-Service Journal* **3**(1), 7–31.

Sierra, C. & Debenham, J. (2005), An Information-Based Model for Trust, *in* 'Proceedings of Fourth International Conference on Autonomous Agents and Multi Agent Systems AAMAS 2005, Utrecht, Netherlands', pp. 497–504.

Sierra, C. & Debenham, J. (2006), Trust and Honour in Information Based Agency, *in* 'Fifth International Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2006)', Hakodate, Japan.

Sierra, C. & Debenham, J. (2007), The Logic Negotiation Model, *in* 'Sixth International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2007)', Honolulu, Hawaii.

Reflection on Development and Delivery of a Data Mining Unit

Bozena Stewart
School of Computing and Mathematics
University of Western Sydney
Locked Bag 1797 Penrith South DC NSW 1797
b.stewart@uws.edu.au

Abstract

Educators developing data mining courses face a difficult task of designing curricula that are adaptable, have solid foundations, and are tailored to students from different academic fields. This task could be facilitated by debating and sharing the ideas and experiences gained from the practice of data mining as well as from teaching data mining. The shared body of knowledge would be a valuable resource which would help educators design better data mining curricula. The aim of this paper is to make a contribution to such a debate. The paper presents a reflection and evaluation of the author's experience with developing and delivering a postgraduate unit Knowledge Discovery and Data Mining.

Keywords: curriculum, data mining, education, teaching

1 Introduction

Advances in computer technologies during the past decade made it possible to generate and collect vast amounts of data in many areas of human activities. The data contains hidden information that needs to be extracted and analysed to provide rules, patterns and models suitable for use in decision making.

The field of Knowledge Discovery and Data Mining (KDDM) has emerged in response to the practical need to analyse huge quantities of data collected in commerce and industry. KDDM has evolved as an interdisciplinary field at the intersection of machine learning, statistics, artificial intelligence, and database systems.

To fully utilise the power of data mining, industry and commerce need a steady supply of highly trained data mining analysts. The universities in Australia and around the world have responded to this need by creating specialised units or complete courses devoted to knowledge discovery and data mining.

In this paper the term "unit" is used to represent a component of a course. An alternative commonly used name is "subject". The term "course" is used here to represent an academic program leading to a degree or a diploma award. A course is a collection of units.

So far there have been few guidelines published in the literature to help unit and course designers develop data mining curricula. In most cases, data mining courses and units have been developed in ad hoc manner, typically by individual academics according to their own experience and research interests. Such curricula are often biased towards the designer's favourite topics and may not adequately cover topics required for data mining analysts employed in industry or commerce.

Recently, in response to the developments in the field the ACM SIGKDD Executive Committee setup the ACM SIGKDD Curriculum Committee to design a sample curriculum for data mining. The committee released the first draft proposal in April 2006 (SIGKDD, 2006). The proposed curriculum contains a comprehensive set of topics and guidelines which will undoubtedly become the basis of many data mining courses in the future. The curriculum is a work in progress which still needs to include sample units (subjects) to provide educators with the guidance on how to structure and package topics for students from different disciplines.

The field of data mining is expanding at a fast pace as new data mining algorithms and methodologies are invented and applied to new domains. The dynamic nature of data mining demands a curriculum that can adapt to changing needs. Educators face a difficult task of developing units and courses that are dynamic, have solid foundations, and that are tailored to students from different academic disciplines. This task could be facilitated by debating and sharing the ideas and experiences gained from the practice of data mining as well as from teaching data mining. The shared body of knowledge would be a valuable resource which would help educators design better data mining curricula. The aim of this paper is to make a contribution to such a debate. The paper describes the author's experience with teaching a postgraduate coursework masters unit *Knowledge Discovery and Data Mining*.

The remainder of this paper is organised as follows. Section 2 surveys related work, Sections 3 to 5 discuss the background of the data mining unit, the unit objectives, the unit content, and the unit assessment. The student feedback on the unit is analysed in Section 6 and the author's reflection on the unit is given in Section 7, followed by a summary of the paper in Section 8.

2 Related Work

There have been relatively few papers published in the literature describing the design and evaluation of data mining units. In recent years there have been several articles published in the educational literature reporting on experience with designing units for Computer Science majors. For example, Musicant (2006) designed a unit focused on "computer science" aspects of data mining. He used specific research papers and assignments that allowed students to implement data mining algorithms themselves. This approach allows students to identify more closely with data mining researchers and challenges the students to understand the intricacies of data mining algorithms.

Chawla (2005) reports on experience in offering a data mining unit to upper level undergraduate and postgraduate students in computer science and engineering. The unit was based on Witten and Frank (1999) text and utilised the Weka data mining tool and Matlab software. The unit incorporated readings of research papers and a conference-style research project.

Saquer (2007) discusses the design and evaluation of a data mining unit taught successfully to computer science and non-computer science majors. The unit was designed for both undergraduate and graduate students. To provide for different student backgrounds, the unit allowed non-implementation as well implementation projects. Graduate students were required to read at least 3 research papers and write a report on them.

There are some similarities between the KDDM unit described in this paper and the data mining units mentioned above. The KDDM unit and Chawla (2005) and Saquer (2007) had a similar aim of providing grounding in theoretical and practical aspects of data mining, all utilised packages for practical work, and all contained a project. The KDDM unit was similar to Saquer (2007) in that both units were targeted to students with different backgrounds. KDDM's project, however, did not involve implementation of data mining algorithms and did not include a conference-style research paper.

3 Unit Background

The data mining unit described in this paper was first created by the author in 2003 as a special topic for the unit *Emerging Issues in Computing*. The term "special topic" is used here as a synonym for content. The purpose of the *Emerging Issues in Computing* unit was to introduce students to new areas of computing. The content of the unit could vary from year to year, depending on which emerging area was being taught. In 2003 the whole unit was devoted to data mining. In the following year, the author created a new unit named *Knowledge Discovery and Data Mining (KDDM).* This unit was designed as an elective unit for the Software Engineering major in the coursework masters degree Master of Computing. Masters students from other majors also were allowed to enrol, provided that they satisfied the prerequisite requirements. The content and objectives of the new unit were similar to the previous unit but the teaching materials were updated and streamlined to better suit the new unit.

The aim of the KDDM unit was to provide students with a broad knowledge of the basic concepts and techniques used in data mining.

In 2005, unfortunately, due to the rationalisation of computing courses and units, the software engineering major in Master of Computing degree was discontinued and most of its units were closed, including the KDDM unit. Hence there were no further offerings of the KDDM unit beyond 2004.

3.1 Unit Content

The content of the unit covered the core data mining topics including data preparation, association rule mining, classification and clustering, and gave a brief overview of several advanced topics including text mining, Web mining, spatial data mining, and temporal mining. More details on the unit content are provided in Section 4.

3.2 Unit Delivery

The unit was delivered in the traditional face-to-face format consisting of a two-hour lecture followed by one-hour tutorial per week for 13 weeks. Tutorials were used to solve theoretical exercises supplementing the topics discussed in lectures. There were no formal laboratory classes but the students were expected to work on practical tasks in their own time. They could do their practical work in the computing laboratory equipped with the data mining tool Clementine from SPSS or they could do it at home using any suitable open source software such as WEKA.

3.3 Teaching Resources

The textbook used in the unit was Han & Kamber (2001). Another recommended text was Dunham (2003). The unit teaching materials were prepared by the author and were available to students online on the WebCT site. To motivate the students to explore data mining topics beyond the scope of the unit, the author also provided references to relevant journal and conference papers found in the ACM and IEEE digital libraries.

3.4 The Students

The students were full-fee paying students, the majority of whom were from overseas. There were 23 students at the start of the semester and 21 students at the end. The prior educational requirement for Master of Computing was an undergraduate degree in Computer Science, Information Technology, Information Systems, or similar. The assumed knowledge for the KDDM unit was a basic background in statistics, database systems and computer programming.

4 Unit Objectives and Unit Content

The unit objectives were to provide students with the knowledge and skills that would enable them to:

1. Understand the knowledge discovery process
2. Choose an appropriate strategy for data preparation
3. Select appropriate data mining algorithms for different applications
4. Compare and contrast each of the following data mining techniques: association rules, decision trees, neural networks, Bayesian networks, the nearest neighbour algorithm, support vector machines, clustering algorithms, and genetic algorithms
5. Apply data mining algorithms for association rule mining, classification and clustering to solving practical problems
6. Use data mining tools to solve a variety of data mining problems
7. Describe the OLAP operations and the design of a data warehouse
8. Explain the issues related to mining of unstructured and semi-structured data: text data, web data, time series data, and spatial data
9. Pursue higher research studies in data mining

	Unit Objectives								
Topics	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	+								+
2		+				+			+
3			+	+	+	+			+
4			+	+	+	+			+
5			+	+	+	+			+
6			+	+	+	+			+
7			+	+	+	+			+
8			+	+	+	+			+
9			+	+	+	+			+
10							+		+
11								+	+
12								+	+
13								+	+

Table 1: Unit topics versus unit objectives

The unit content was designed to satisfy the unit objectives above. Topics were chosen on the basis of the author's experience and literature survey of data mining textbooks. The emphasis of the content was on the major data mining concepts and algorithms. In addition, the basics of data warehousing and OLAP, and several advanced data mining topics were briefly introduced in the last four weeks of the semester. The main reason for including the topic on data warehousing and OLAP so late in the semester was that the unit did not use data sets stored in databases, and at that time the author did not consider it necessary to introduce this topic earlier. The weekly schedule of topics is listed below.

1. An overview of data mining
2. Data preparation
3. Association rule mining
4. Decision trees
5. Neural networks
6. Bayesian networks
7. Nearest neighbour; Support vector machines; Regression
8. Clustering
9. Genetic algorithms
10. Data warehousing and OLAP
11. Text mining; Web mining
12. Spatial mining
13. Temporal mining

To see how the unit content matched the unit objectives, the relationships between the unit topics and the unit objectives are shown in Table 1. The rows in the table represent the weekly topics and the columns represent the unit objectives. The plus (+) symbol in a given row and column indicates that the topic in that row contributed to achieving the objective in the corresponding column.

The distribution of the '+' symbols in Table 1 shows that all the unit objectives were covered by the content of the unit. For example, the objectives 1, 2 and 7 were discussed in one topic (lecture) each, while the objectives 3 to 6 were addressed by 7 or more topics. This highlights the fact that the emphasis of the unit was on the core data mining algorithms and their associated concepts.

The objective 9, "Pursue higher research studies in data mining", requires knowledge of all core data mining topics as well as knowledge of more advanced material. Hence it could be argued that all of the topics covered in the unit contributed in some way to the students' ability to undertake higher research studies in data mining.

5 Unit Assessment

The unit was assessed on the basis of two components: Continuous Assessment (60%) and Final Examination (40%). Continuous assessment consisted of two individual assignments, each worth 15%, and a group project worth 30% of the total mark. The final examination was an open book 3 hour examination with 10 minutes of reading time.

5.1 Assignments

In both assignments, the students were required to perform practical experiments and submit a written report describing the data mining approach taken and the results obtained. To perform practical experiments the students used data mining tools, either Clementine in the computing laboratory or open source software at home.

Assignment 1 was focused on the topics of data preparation, association rule mining and decision trees. The students were required to solve two data mining problems. The first problem was to mine association rules from transactional data from a retail supermarket,

obtained from the FIMI Repository. The task involved determining suitable levels of minimum support and confidence, finding association rules, determining which rules were interesting, and suggesting how the retail supermarket could apply the interesting rules to increase its sales. In the second problem the students were given a data set from a banking domain containing missing values, errors, and redundant attributes and they had to clean the data and then use the prepared data set to mine decision trees. The students were required to experiment with alternative approaches to data preparation, evaluate resulting decision trees using cross-validation, and use the decision trees to predict unknown labels in a given test set. They submitted a report describing the experiments, results obtained, and explaining the effects of data preparation on the resulting decision trees.

Assignment 2 was concerned with classification and prediction. Its main objective was to compare the predictive performance of several different classification algorithms on a moderately large real-world data set. The students could choose any three classification methods. They were given a training data set from an insurance domain containing an unbalanced class attribute and their first task was to create a balanced data set. Then they used the balanced data set to create 3 different classification models and used each model to predict class labels in a supplied test set. The students were required to evaluate the performance of each algorithm in terms of classification accuracy, precision and recall. In the report they were required to describe the experiments and the results obtained, and also explain how the results could be used to produce a mailing list for a direct marketing campaign.

5.2 Project

The project was a group project conducted in teams of 3 students. Its aim was to give the students practical experience with data mining of a large real-world data set. Each team selected a large data set from the UCI KDD Archive. The teams were free to use any appropriate data mining techniques and any tools available to analyse the data. In the middle of the semester, each team submitted a description of their chosen data set and the plans and timeline for the intended approach. At the end of the semester they submitted the final report presenting the results and describing the work in detail.

5.3 Unit Assessment versus Unit Objectives

To see how well the unit assessment corresponded to unit objectives, the relationships between the assessment tasks and the unit objectives are shown in Table 2. The rows of the table represent the assessment tasks and the columns represent the unit objectives.

The objective 7, concerned with basic concepts of OLAP and data warehousing, and the objective 8, concerned with advanced data mining, were assessed only by means of the final examination. They were introduced only briefly and represented a minor component of the unit content.

	Unit Objectives								
Assessment	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Assignment 1	+	+	+	+	+	+			+
Assignment 2	+	+	+	+	+	+			+
Project	+	+	+	+	+	+			+
Final Exam	+	+	+	+			+	+	

Table 2: Assessment versus objectives

5.4 Student Performance

Have the unit objectives been achieved? This question can be partially answered by looking at the students' performance on the individual assessment tasks.

It can be seen from Table 3 that the students performed well on the specified assessment items. The individual assignments were in most cases of a good standard. Most of the group projects were of high standard, indicating that the students mastered the core concepts and techniques well. Students also performed relatively well on the final examination, with the average mark of 60%. The students' performance suggests that the unit met the unit objectives.

Assessment Task	**Average Mark**	**Average %**	**Min Mark**	**Max Mark**
Assign 1 (out of 15)	11.5	76.8	7.8	14.9
Assign 2 (out of 15)	13.0	86.8	9.9	15.0
Project (out of 30)	24.8	82.5	17.4	29.7
Final Exam (out of 40)	24.0	60.0	14.4	36.4
TOTAL (100)	73.3	73.3	50.1	92.5

Table 3: Assessment and student performance

6 Student Feedback

To obtain a feedback on the delivery of the unit, the author conducted the student evaluation survey (SEEQ) in the last week of the semester. Most of the questions on the SEEQ evaluation form were concerned with the teaching performance of the staff member and only a few questions related to more general aspects of the unit. The mean scores of responses on the more general questions are shown in Table 4 and the details of the distributions of student responses are given in Table 5 and Table 6. For comparison, the tables include the survey results for both data mining units, *Emerging Issues in Computing* (EIC) offered in 2003, and *Knowledge Discovery and Data*

Mining (KDDM) offered in 2004. There were 24 respondents out of 40 in 2003, and 10 respondents out of 21 in 2004.

The results in Table 4 show that the mean scores for the 2003 offering were considerably lower than the corresponding scores for 2004. These differences are also apparent in the distributions of the scores in Tables 5 and 6. However, a detailed comparison of the scores is difficult because of unequal number of respondents in the two evaluation surveys.

There were a number of reasons for the differences in the evaluation scores of the two units. One reason was that the student populations were quite different in 2003 and 2004. The *Emerging Issues in Computing* unit was a general elective not aimed at any particular major. As a result, the students had a wide range of backgrounds and some lacked adequate mathematical and computing knowledge. The *Knowledge Discovery and Data Mining* unit, on the other hand, was aimed at Software Engineering major students who had appropriate background for this unit and consequently faced few difficulties.

Another reason for the differences in the scores was that in 2003 the data mining unit was offered for the first time and the author had no prior experience with teaching a similar unit. In view of the student feedback, the author revised the syllabus and teaching materials for the second offering in 2004.

From the scores in Table 4 for 2004 survey and the bar charts of student responses in Tables 5 and 6 it can be concluded that:

- The respondents found the unit intellectually challenging and stimulating. They considered the content of the unit valuable and their interest in data mining was increased by doing this unit. They learned and understood the subject material in the class.
- They considered the methods of assessment to be appropriate, and the assignments and prescribed readings to be valuable and contributing to their appreciation and understanding of the unit.
- Overall, they compared the unit favourably to other units at the same University.
- The responses on unit workload and difficulty varied from "medium" to "very hard". The number of hours per week required outside class showed a range of values between 2 and 9 hours, with the mean of 5.3 hours. This figure is not excessive considering that for a 10 credit point unit a student is assumed to study for about 10 hours per week. With 3 hours of class contact per week, the students were expected to spend another 7 hours per week on the unit outside class.
- Approximately half of the respondents thought that the pace of the unit was "about right" and half thought that the pace was "too fast". The differences in responses were probably due to different backgrounds of the students. Some students came from more technical computer science courses while several were from relatively non-technical information systems degrees. The latter group experienced some difficulties with understanding of data mining algorithms.

	Question	2003 EIC unit Mean Score out of 9	2004 KDDM unit Mean Score out of 9
Learning/Academic Value	1. You found the class intellectually challenging and stimulating	5.5 61.1%	7.6 84.4%
	2. You have learned something which you considered valuable	6.46 71.8%	7.8 86.7%
	3. You interest in the unit has increased as a consequence of this class	6.13 68.1%	7.2 80.0%
	4. You have learned and understood the subject material in this class	5.75 64.2%	7.8 86.7%
Examinations / Grading	5. Methods of evaluating student work were fair and appropriate	6.08 67.6%	7.8 86.7%
	6. Assessments/Examinations tested units content as emphasised by staff member	6.08 67.6%	8.2 91.1%
Assignments / Readings	7. Required readings/texts were valuable	6.38 70.9%	7.9 87.7%
	8. Readings, assignments, etc. contributed to appreciation and understanding of the unit	6.21 69.0%	7.9 87.7%
Class Rating	9. Overall, how does the class compare with other classes at this institution	N/A	7.9 87.7%
Workload / Difficulty	10. Unit difficulty, relative to other units, was	6.5 72.2%	6.6 73.3%
	11. Unit workload, relative to other units, was	6.41 71.2%	6.5 72.2%
	12. Unit pace was	5.91 65.7%	6.6 73.3%
	13. Average number of hours per week required outside class	5.2	5.3

Table 4: Student feedback

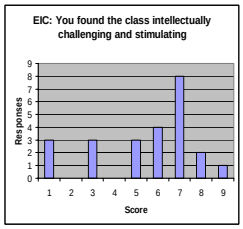
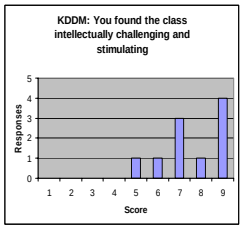
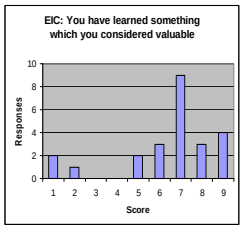
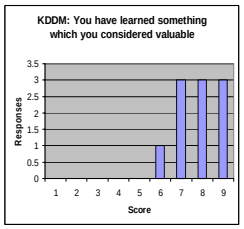
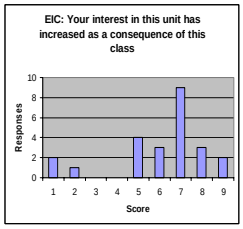
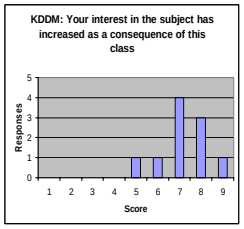
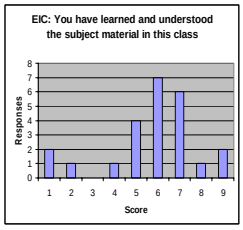
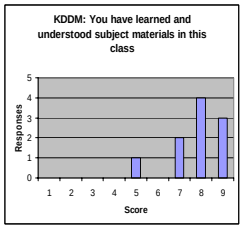
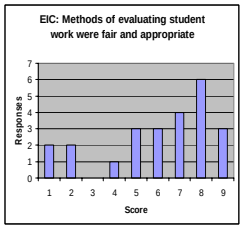
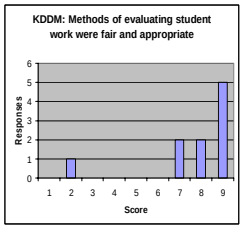
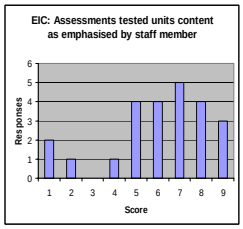
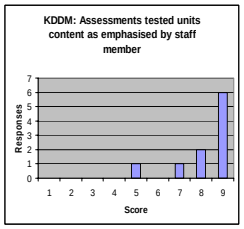
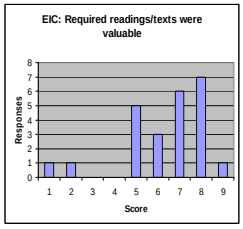
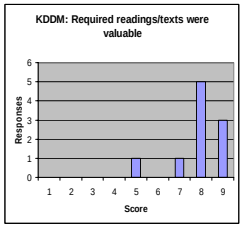
	2003 Emerging Issues in Computing	2004 KDDM
1		
2		
3		
4		
5		
6		
7		

Table 5: Distribution of scores for questions 1-7

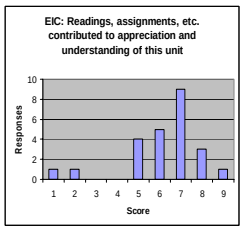
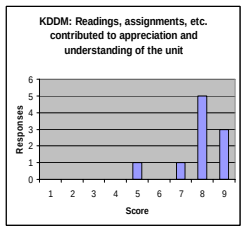
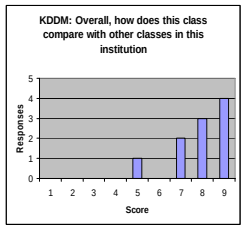
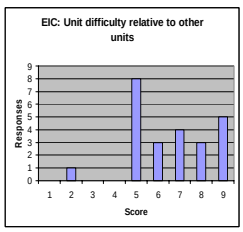
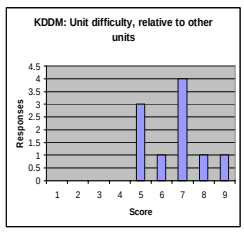
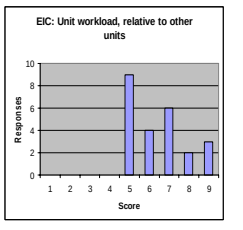
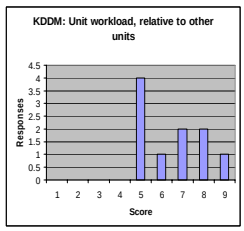
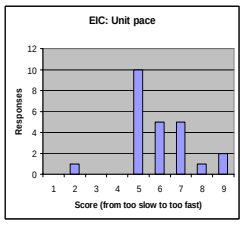
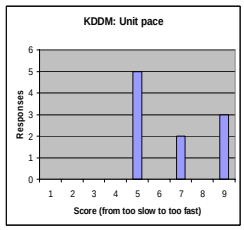
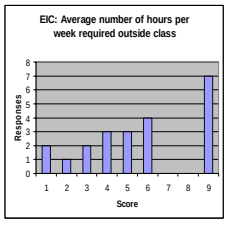
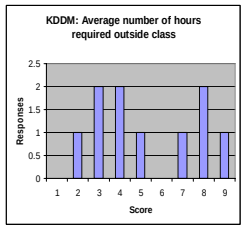
	2003 Emerging Issues in Computing	2004 KDDM
8		
9		
10		
11		
12		
13		

Table 6: Distribution of scores for questions 8-13

7 Reflection

Reflecting back on teaching of the KDDM unit, what aspects of the unit could have been done better? What were the lessons learned?

Overall, the student feedback on the unit was very positive. The only area of concern was the amount of work required which was perceived as excessive by some of the students. In the "additional comments" section of the SEEQ survey, some of the respondents commented that the unit covered too many topics and they had insufficient time to learn all the material. Similar comments were made by several respondents in the 2003 survey.

There are many issues involved in designing a data mining curriculum. The key factors are unit objectives, unit content, and unit assessment. All three must be well coordinated and also must be well matched to the target audience.

7.1 Unit Objectives

Were the unit objectives appropriate? The objectives were chosen to satisfy the core concepts of data mining and ensure a basic understanding of several advanced data mining concepts.

From Table 2 it can be seen that the three continuous assessment items, Assignment 1, Assignment 2, and Project all tested the same six objectives. It would have been better to use finer grained objectives to allow a clear differentiation between the individual assessment tasks. It is important to be able to show how unit objectives are satisfied progressively by different assessment tasks.

Was the range of the unit objectives appropriate? The student feedback and the author's own experience from delivering the two units suggest that similar introductory data mining units should focus their learning objectives towards the core principles and techniques and leave more advanced objectives to later units.

7.2 Unit Content

Was the unit content well chosen? The unit content and unit objectives are closely related. The content must satisfy the unit objectives. Comparing the content topics in the KDDM unit to the ACM SIGKDD guidelines in the draft proposal (SIGKDD, 2006), it can be seen that the KDDM unit corresponded quite closely to the *Foundations (Course I)* unit in the proposed curriculum. However, from students' responses to the evaluation survey, it appears that it would have been better to omit the advanced topics and focus on fewer core topics in more detail.

7.3 Unit Assessment

Were the assessment tasks appropriate? The unit assessment was designed to provide a variety of assessment tasks and to examine as much of the unit content as possible. The assignments and the project provided the students with practical experience in solving data mining problems and using data mining tools. The final examination tested overall understanding of the unit content.

On the basis of students' feedback, it can be concluded that the assessment tasks were appropriate and helped the students to appreciate and understand the subject matter.

7.4 Technical Level of Data Mining Units

One important issue to consider when designing a data mining unit is the technical level of the unit. A good understanding of data mining algorithms requires a solid background in mathematics, statistics and algorithms and data structures. Such background knowledge is normally provided in computer science degree programs but unfortunately not in many information systems and information technology programs.

As the author's experience with teaching *Emerging Issues in Computing* in 2003 showed, students who don't have adequate background in statistics and computing may experience major problems with understanding mathematical and algorithmic notation and may find technical data mining concepts too difficult to grasp.

To make data mining accessible to students in less technical fields, the data mining units for those fields would have to be focused more on business problem solving and business applications rather than on theoretical aspects of data mining algorithms.

For more technical disciplines such as computer science and engineering, the data mining units should include technical details of data mining concepts and algorithms as well as projects involving industrial or scientific applications of data mining. Computer science students could also be given programming tasks to implement some of the data mining algorithms.

Employment opportunities in data mining range from technical positions requiring knowledge of statistics, computer programming, machine learning and artificial intelligence to non-technical positions in business analysis. There is clearly a need for data mining units and courses targeted to students from different fields.

7.5 Advanced Data Mining Topics

How should the advanced data mining topics be taught? Data mining is nowadays used in many diverse areas of business, industry and research. A single unit cannot cover all aspects of data mining, several units are required. The ACM SIGKDD curriculum draft proposal gives guidelines for two units: *Foundations* and *Advanced Topics* (SIGKDD, 2006).

The *Foundations* unit is focused on the core data mining concepts of data pre-processing, data warehousing and OLAP, association rule mining, classification, clustering, time series and sequence mining, text mining and Web mining, visual data mining, and social impact of data mining.

The proposal for the *Advanced Topics* unit includes advanced material for the core topics discussed in *Foundations*, as well as additional advanced data mining

topics. It contains advanced material on data pre-processing, data warehousing and OLAP, association rule mining, classification, clustering, and time series and sequence mining. In addition, it includes material on data streams mining, spatial, spatiotemporal and multimedia mining, biological mining, text mining, hypertext and Web mining, data mining languages, standards and system architectures, data mining applications, data mining and society, and trends in data mining.

The curriculum proposal does not yet provide any recommendations on how to select appropriate topics to create units for students from different disciplines.

The author's experience with teaching the KDDM unit and the feedback received from the students, indicate that it would be better to limit the material in the *Foundations* unit to the basic core topics and cover them in more detail. The topics such as time series mining, sequence mining, text mining, Web mining, and visual data mining could be included only in the *Advanced Topics* unit. The educators developing advanced data mining units will find the *Advanced Topics* unit an excellent source of content from which to choose a suitable subset of topics.

8 Summary

This paper presented the author's reflection on the experience in developing and delivering a data mining unit to a diverse cohort of students. The main outcomes of this study were:

- The students considered the unit content to be valuable and their interest in data mining was increased by doing this unit.
- The students were satisfied with the unit organisation, delivery, and assessment.
- Unit workload and difficulty were considered too high by some of the respondents to the student evaluation survey.
- Unit objectives were not sufficiently detailed to show how different assessment tasks progressively satisfied different unit objectives.
- Unit content was judged to be too broad, beyond the scope of an introductory unit.
- Unit assessment covered all the unit objectives and was considered to be appropriate.
- Students with inadequate background in statistics and computing experienced difficulties with mastering data mining concepts.

The outcomes of this study suggest the following recommendations for designers of data mining units:

- The curriculum for a postgraduate unit in data mining should be designed in relation to the academic background of the student cohort. Students from technical fields such as computer science and engineering appreciate deep knowledge and understanding of theoretical concepts and algorithms. Students from business oriented fields tend to be interested mainly in applications of data mining to solving specific business problems. They view data mining from a user's point of view.
- Unit objectives and unit content should be developed in parallel. This would ensure that the unit content satisfies the unit objectives, and also would help to determine appropriate level of detail for the unit objectives.
- Unit content for an introductory data mining unit should focus on the core concepts and cover them in depth rather than cover many concepts in a superficial way.
- Unit assessment should contain a sufficient diversity of tasks to motivate students to learn, and the tasks should correspond to the unit objectives. Each unit objective should be assessed by at least one assessment task.
- Advanced data mining topics should be covered in a separate advanced data mining unit.

9 References

Chawla, N. V. (2005): Teaching data mining by coalescing theory and applications. *Proc. 35th ASEE/IEEE Frontiers in Education Conference*, Indianapolis, IN, USA, 17-23, IEEE.

Clementine, SPSS Inc. http://www.spss.com/clementine/. Accessed 2 Sept 2007.

Dunham, M. H. (2003): *Data mining: introductory and advanced topics*. Pearson Education.

FIMI Repository, http://fimi.cs.helsinki.fi/data/. Accessed 2 Sept 2007.

Han, J., Kamber, M. (2001): *Data mining: concepts and techniques*. Morgan Kaufmann.

Intensive Working Group of ACM SIGKDD Curriculum Committee: Data mining curriculum: a proposal (Version 1.0) http://www.sigkdd.org/curriculum.php. Accessed 2 Sept 2007.

Musicant, D. R. (2005): A data mining course for computer science: primary sources and implementations. *Proc. 37th SIGCSE technical symposium on Computer science education SIGCSE '06*, **38** (1): 538-542, ACM Press.

Saquer, J. (2007): A data mining course for computer science and non-computer science students. *Journal of Computing Sciences in Colleges* **22** (4):109-114, ACM.

UCI KDD Archive, Information and Computer Science, University of California, Irvine. http://kdd.ics.uci.edu/. Accessed 2 Sept 2007.

Weka 3: Data Mining Software in Java, University of Waikato, NZ www.cs.waikato.ac.nz/ml/weka/. Accessed 2 Sept 2007.

Witten, I. and Frank, E. (1999): *Data mining: practical machine learning tools and techniques with Java implementation*. Morgan Kaufmann.

Evaluation of a Graduate Level Data Mining Course with Industry Participants

Peter Christen

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

Abstract

Data mining courses are increasingly being taught at many universities at both undergraduate and graduate levels. This paper reports on a new graduate level data mining course run for the first time in 2007 at a major Australian university. The course had almost 20% enrolments of industry based participants from both private and public sector organisations. This paper discusses the student population and presents the course structure and assessment. An empirical evaluation of student responses, conducted at the end of the course, is then provided, with an emphasis on differences in responses from graduate students and external participants. To the best of the author's knowledge, this is the first such detailed empirical evaluation of a data mining course.

Keywords: Data mining education, postgraduate studies, course evaluation, industry.

1 Introduction

With many businesses, government organisations and research projects collecting massive amounts of data, the techniques collectively known as *data mining* have in recent years attracted interest from both industry and academia. As a result, there is now an increasing demand from both industry and government agencies for graduates with data mining skills, and data mining courses are now being taught at many universities throughout the world.

Data mining is multi-disciplinary and draws from fields such as statistics, machine learning and AI, database technology, algorithms and data structures, high-performance and parallel computing, visualisation, and privacy and security technologies (Han and Kamber 2006). This wide spectrum challenges the teaching of data mining, as necessarily some prior knowledge in some of the above mentioned disciplines is required. As reported elsewhere (Saquer 2007), one major challenge when teaching data mining is that students will have different backgrounds, and likely have knowledge in some of the core disciplines of data mining. The wide range of concepts and techniques related to, and required by, data mining also limits either the depth or breath of how much can be covered in a normal one-semester university course.

The aim of the new graduate level course discussed in this paper was to cover more than just the core data mining techniques and algorithms (like classification, prediction, clustering and association rule mining), but to also expose students to other important issues relevant to the knowledge discovery in databases (KDD) process, ranging from data quality, pre-processing and integration to privacy and social impacts of data mining. Additionally, a second focus of the course was to give students insight into current data mining research through reading papers and an oral presentation of a selected research paper.

The course was not only advertised as a graduate level course on relevant university Web sites and student handbooks, but also announced to several local e-mail lists containing contacts from government agencies and private sector organisations known to have interests related to data mining. The course syllabus as available in the student handbook was:

> Large amounts of data are increasingly being collected by public and private organisations, and research projects. Additionally, the Internet provides a very large source of information about almost every aspect of human life and society.
>
> This course provides a practical focus on the technology and research in the area. It focuses on the algorithms and techniques and less on the mathematical and statistical foundations.

In the following section a short overview of related work is provided. In Section 3 some details about the student population is discussed, and in Sections 4 and 5 the course structure and assessment, respectively, are presented. Section 6 then contains a detailed discussion of the course evaluation based on an end-of-semester questionnaire and observations by the author. Finally, the paper is concluded in Section 7 with a discussion of potential changes and improvements for future offerings of this course.

2 Related Work

Only a very small number of reports that describe and evaluate data mining courses at university level have been published so far, most of them in the computer science education literature.

An approach to teaching data mining at undergraduate level is described in (Lopez and Ludwig 2001). Their course was using the *Weka* open source data mining toolbox and accompanying text book (Witten and Frank 2000). Students initially had to research a data mining topic of their choice and present their findings, followed by lectures covering the major data mining topics. Practical exercises were conducted using *Weka*, which proved to be a useful learning tool. In the final part of the course, students had to perform a data mining group project

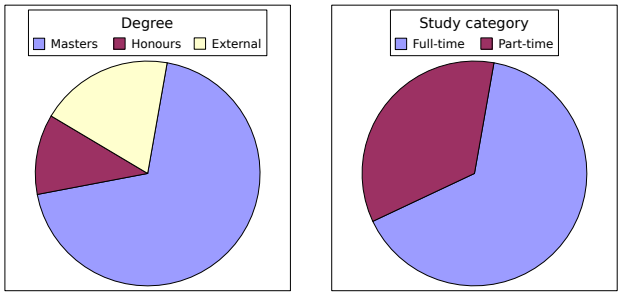


Figure 1: Student degrees and categories.

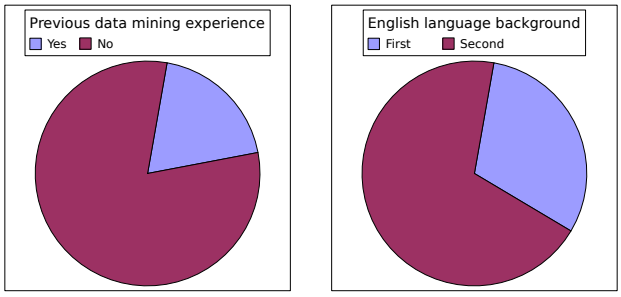


Figure 2: Student's previous data mining experience and English language backgrounds.

using a data set with around 250,000 records provided by the U.S. Forest Service. The authors report that the course was a success, however, they do not discuss what prior knowledge was expected from the students, nor do they present an empirical evaluation of their course.

A rather different approach is presented in (Musicant 2006). This data mining course for computer science undergraduate students is based mainly on selected research papers. The author builds the assignments for the course on these papers, asking students to read them, post questions onto a discussion forum, and then implement the algorithms described in the papers. In a final project students were free to either implement an advanced data mining algorithm of their choice, or conduct a data mining study on data sets of their choosing. The author then remarks that his course is unique as it exposes students to reading research papers and requires their critical thinking by coming up with relevant questions about these papers. He however concludes that marking and grading of the implementation and data mining studies is the most challenging part of his course.

A very recent report on a data mining course aimed at both undergraduate and graduate computer science and non-computer science students is presented in (Saquer 2007). The emphasis of this course was to provide an understanding of the basic (and some advanced) data mining algorithms, with a focus on classification, clustering and association rules. Each of these three topics received about four weeks out of a sixteen week course, which started with an introduction and finished with a group project where student either had to implement an advanced algorithm or learn and present about a data mining topic not covered in the lectures. Two assignments were conducted where students had to select a data set of their choice, use two algorithms to carry out a data mining project, and write a report of their findings. There is no thorough course evaluation provided in this paper; however, the author reports positive student feedback.

3 Course Student Backgrounds

Of the initial 27 students enrolled in the data mining course discussed here, one dropped out throughout the semester, and only one of the remaining 26 students was female. The details of student degrees and study categories are shown in Figure 1, while Figure 2 provides a breakdown in student's previous experience in data mining and their language background (if they were native English speakers or not).

Eighteen students were enrolled in a course-work masters program, while three were fourth-year computer science honours students, and five of the students were external participants that only enrolled for this course (non-award enrolment) and came from various private and public sector organisations (mainly from government agencies). Around 60% of the students were studying full-time.

At the beginning of the course, five of the 26 students indicated they had previous data mining experience (this included three of the five external participants), and three had attended a data mining course before (including one of the five external participants). Almost 70% of all students had English as a second language, with several students having arrived in Australia just this year for their one-year masters course-work studies.

4 Course Structure

The course consisted of nineteen one-hour lectures as summarised in Table 1, with the first five modules presented before the semester break and the rest afterwards. Modules 4, 5, 6, 7 and 9 were mostly based on corresponding chapters from the text book *Data Mining: Concepts and Techniques* (Han and Kamber 2006) which was used in the course, while the other modules were a mix of text book based material and additional material developed by the author. The *End-to-end data mining* lecture was given by an industry based data miner with many years of practical and teaching experience.

Four practical one-hour laboratory sessions were conducted using the open source tool *Rattle* (Williams 2007)[1], which is a graphical user interface built on top of the *R* statistical programming language[2]. *Rattle* provides a logical user interface to many data mining algorithms implemented in *R*, and includes techniques for data exploration and transformation, clustering, association rule mining, various classification techniques, and a variety of evaluation methods. It also allows direct user access to the underlying *R* console and thus facilitates further exploration of data mining algorithms as well as statistical concepts.

In the first practical laboratory session *Rattle* was introduced to the students and a small data set was explored. The second session covered association rule mining, and the students had to conduct various experiments using a publicly available data set sourced from the UCI machine learning repository (Newman et al. 1998). In the third and fourth laboratory sessions students were asked to work with decision tree and support vector machine classifiers, and compare their performance using various evaluation methods, such as ROC, risk and precision-recall curves. *Rattle* was also used for the assignments as discussed in the following section.

Laboratory sessions were held around every two weeks, with tutorials in the intermediary weeks. For each tutorial, students were asked to read two data

[1] http://rattle.togaware.com
[2] http://www.r-project.org

Module	Topic	Hours
1	Course introduction and data mining overview	1
2	Data mining process and data issues in data mining	2
3	Data pre-processing and data integration and linkage	2
4	Mining frequent patterns and associations	2
5	Cluster analysis	2
6	Classification and prediction	4
7	Mining time series and data streams	1
8	Privacy-preserving data mining	1
9	Web and text data mining	2
10	End-to-end data mining, data mining trends and social impacts	2

Table 1: Course lectures overview.

Tutorial	Papers discussed
1	*Data cleaning: Problems and current approaches* (Rahm and Do 2000) *Methods for evaluating and creating data quality* (Winkler 2004)
2	*Fast algorithms for mining association rules* (Agrawal and Srikant 1994) *Selecting the right interestingness measure for association patterns* (Tan et al. 2002)
3	*On comparing classifiers: Pitfalls to avoid and a recommended approach* (Salzberg 1997) *Classifier technology and the illusion of progress* (Hand 2006)
4	*State-of-the-art in privacy preserving data mining* (Verykios et al. 2004) *Names: A new frontier in text mining* (Patman and Thompson 2003)

Table 2: Tutorial papers.

mining papers, selected by the author and listed in Table 2, which were then discussed in the tutorial sessions. These tutorials exposed students to a broad range of topics and were aimed at deepening and complementing the topics covered in the lectures and laboratory sessions, as well as for students to critically read and analyse research papers, and to encourage discussion on these papers. Reading these tutorial papers was also aimed at preparing students for their presentation of a research paper in the final semester week, as discussed in the following section.

All course material was made available on a Web site, with lecture slides normally being uploaded two days before a lecture. Besides lectures, tutorials and laboratory sessions, an electronic discussion board (forum) was set up to allow dissemination of information from the lecturer to students, and to enable students to post questions and discuss them among themselves. Students were encouraged to post questions into this forum rather than e-mail them to the lecturer, with the aim to initiate discussions between students. This turned out to be less successful than expected: most posted messages were student questions, that were followed by answers and clarifications by the lecturer, but there was almost no discussion among students. However, most students reported in the end-of-semester questionnaire to have used the forum regularly as readers, and they commented that it was a useful tool for getting information.

5 Course Assessment

The assessment for the course consisted of two assignments, each worth 15% of the final course mark; a paper presentation and report in the last semester week (worth 20%); and a final written examination which was worth 50% of the final course mark.

The first assignment consisted of two parts, the first being a two-page essay on data issues related to a university data warehouse (such as what data a university is collecting, how it would design a data warehouse, what kind of data mining it would be interested in, and what the data mining challenges would be in such an environment). For the second part, students could choose to either implement, test and evaluate a simple clustering algorithm (such as k-means), or conduct a clustering project using the *Rattle* tool on a publicly available data set of their choice. For both options they had to write a four-page report detailing their findings. Eight students choose to implement a clustering algorithm, all of them masters students, while all others conducted the clustering project. Somewhat surprisingly, all three computer science honours students selected the clustering project rather than implementing an algorithm.

For the second assignment, students had to conduct association rule mining on a small data set made of seven transactions; had to calculate different classifier accuracy measures (such as precision, recall and specificity) on a confusion matrix that was based on their seven-digit university identifier (for example, for an identifier 1234567, the number of true positives were the first four digits 1234, the number of false positives the last three digits 567, the number of true negatives were the first four digits of the reversed identifier 7654, and the number of false negatives the last three digits of the reversed identifier 321); and they had to conduct a classification project using *Rattle*, comparing three different classification techniques (decision trees, support vector machines, and a third classifier of their choice) on a publicly available data set of their choice sourced from the UCI machine learning repository (Newman et al. 1998). For the last part they had to write a report that had to include a ten-line executive summary, details of the data exploration and transformation steps they have conducted, a description of the classifier approach they have taken, details of the results they have achieved (including confusion matrices and ROC curves for all three classifiers), a critical summary of their project, and a reflection of what they have learnt.

The final student presentation was originally planned to consist of a 15-minutes talk by each student, but due to the larger enrolment number this had to be changed to 'lightning' talks of five minutes per student (still resulting in three one-hour sessions of presentations). Students were able to select a data mining research paper of their choice (with the only limitation that it had to be published from the year 2000 onwards), and then had to write a report addressing the techniques described in the paper, data sets used and experiments conducted, measurements employed to assess the quality of complexity of the techniques described, and finally a critical assessment of the paper (detailing, for example, limited experimental studies on only small or only synthetic data sets, claims written by the authors that were not supported by theory or experiments, etc.). They then had to summarise their report onto four slides and give a short presentation. They were able to receive up to ten marks for their report, and up to five marks each for their slides and their oral presentation.

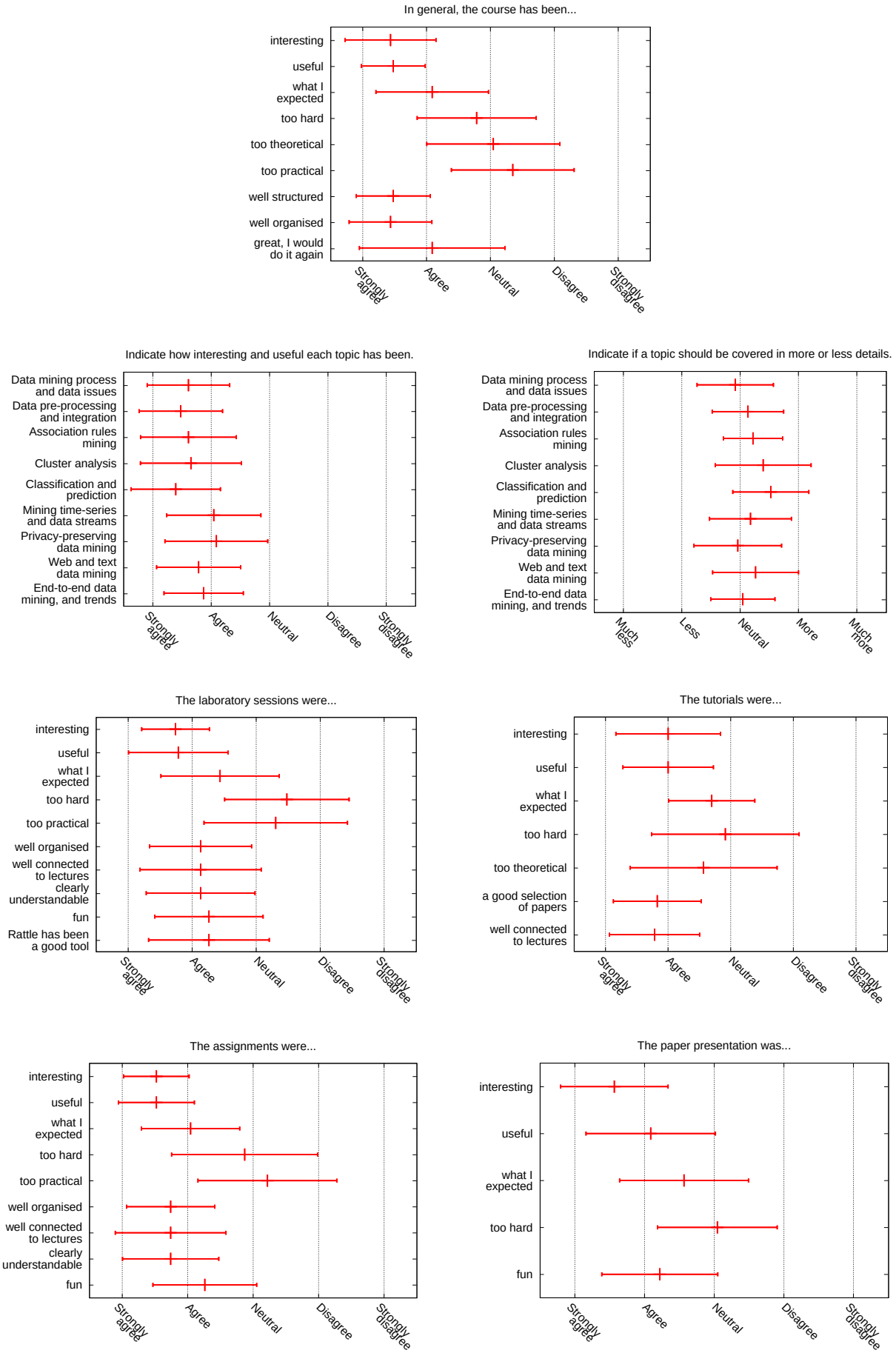


Figure 3: Course evaluation results from all students (mean and standard deviation).

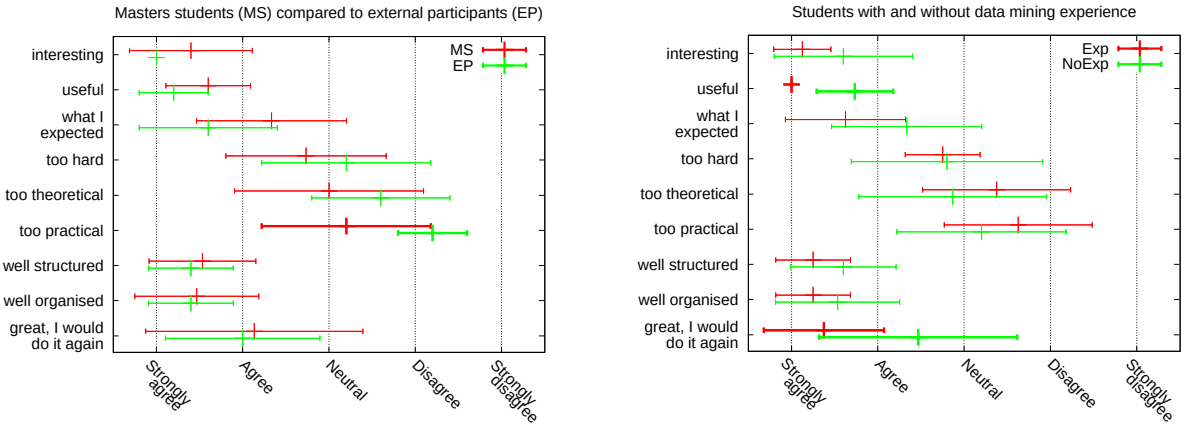


Figure 4: In general, the course has been... (Statistically significant differences ($p < 0.05$) are highlighted bold in this and all following figures.)

Half of the total course mark was based on a final three-hour written examination. While this was not an open book examination, the students were allowed to take one sheet of A4 paper with written notes into the examination. The emphasis of the questions, which covered the whole course material, was on explaining concepts rather than formulas, definitions or implementation details of algorithms.

Overall, the students put a lot of effort into their assignments, and especially into their presentations, which resulted in excellent written project reports and oral presentations. The high quality of the submitted material is reflected in good average student marks, as will be discussed in Section 6.1 below.

6 Course Evaluation and Discussion

At the end of the semester a four-page questionnaire, designed by the author, was handed out to all 26 students. Given that this was a new course, run for the first time in 2007, the main objective of this questionnaire was to get feedback about the course that would allow improvements in coming years. The questionnaire consisted of seven tables where students had to rate the different aspects of the course, plus eleven free-format questions allowing students to provide additional comments. The main results from the seven questionnaire tables are shown in Figures 3 to 10, and they will be discussed in detail in Section 6.2 below. These figures display mean and standard deviation values. First, in the following section, a short overview of the student marks from the various course assessments is provided.

6.1 Distribution of Student Marks

The quality of the submitted reports has been mainly good for all four assessments of this course. The average mark for the first assignment was 10.8 (out of 15), with a standard deviation of 1.9 and marks ranging from 5.5 to 13.5. For the second assignment, the marks were higher, with a mean of 13.0 (out of 15), a standard deviation of 1.6 and marks from 9.5 to 14.5. The student presentation marks had a mean of 16.5 (out of 20) with a standard deviation of 1.6, and marks ranging from 13.5 to 19.5. The average final exam mark (out of 50) was 37.7, with a standard deviation of 9.6 and marks ranging from 21 to 48.5. All students passed the course, with ten students achieving a high-distinction course mark (80 or above out of 100) and eleven a distinction (between 70 and 79). Only five students had a course mark below 70.

6.2 Questionnaire Results

Of the 26 students enrolled, 23 returned the questionnaire, including all three computer science honours students and all five external participants. The overall results of all returned questionnaires are shown in Figure 3. As can be seen, general feedback was very positive, with most students agreeing that the course had been useful and interesting, and that it had an appropriate mix of theory and practice. Most of the topics covered in the course were well received, with more detailed coverage desired mainly for the core topics of clustering, classification and prediction. The laboratory sessions could have been made slightly harder and more practical, while some of the papers discussed in the tutorials were felt to be too theoretical. In hindsight, it would have been advantageous to have provided additional material, containing mainly statistical and mathematical background information, together with the tutorial papers.

In order to be able to discuss the differences in feedback between graduate (masters) students (15 responses) and external participants (5 responses), as well as to differentiate between those who indicated they had previous data mining expertise (8 responses) and those who did not (15 responses), Figures 4 to 10 show two graphs each with the corresponding student sub-groups. Note that in the analysis of masters students and external participants (left side in the figures) the results of the three honours students are not included. However, they are included in the graphs showing previous data mining expertise or not (right side in the figures). Statistically significant differences in the results, as measured with a $p < 0.05$ confidence using the two-tailed Mann-Whitney U test (Sheskin 2004), are shown in bold.

6.2.1 General Course Impressions

Figure 4 shows the results of the question *'In general, the course has been...'*. The main, statistically significant, difference between masters students and external participants was that the external participants would have liked the course to be more practical. Other differences between masters students and external participants were that the course was closer to what the external participants expected; that masters students found the courses harder; and that external participants also would have liked the course to contain more theory. A similar distinction can be seen between those students who had previous data mining experience and those who did not. Significant

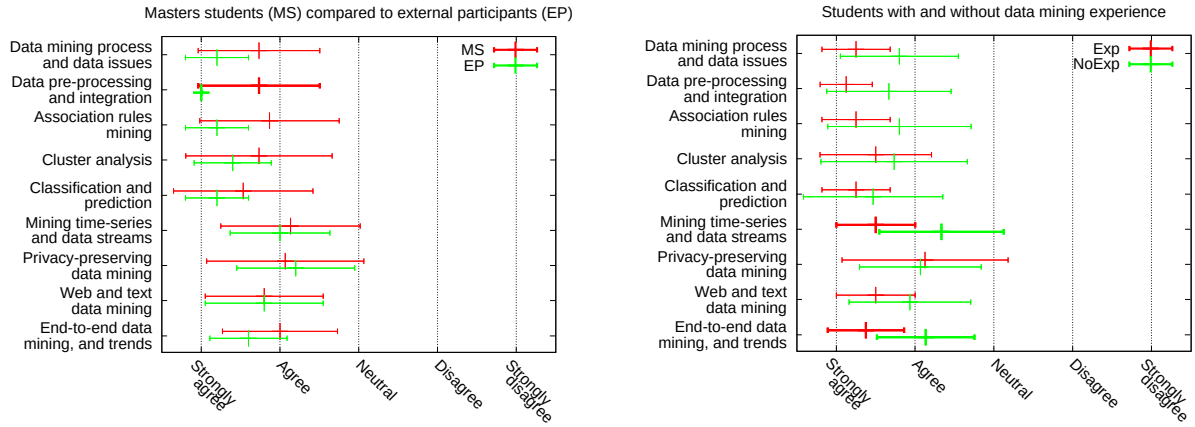


Figure 5: Indicate how interesting and useful each topic has been.

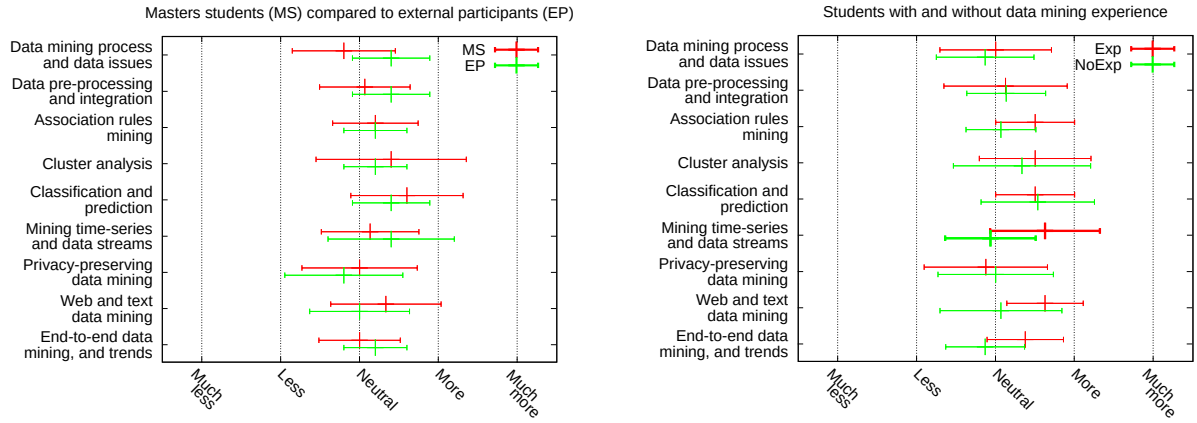


Figure 6: Indicate if a topic should be covered in more or less detail.

differences were that those with previous experience found the course to be more useful (they all strongly agreed that the course was useful) and that they more likely would do such a course again. Those with previous experience also found the course more interesting, they agreed more in that the course was what they expected, and they would have liked the course to be both more theoretical and more practical.

Some students commented in their questionnaire that they appreciated the clear communication by the lecturer, which included schedules, material put online well in time, as well as clear formulation of what was expected in assignments. Many students also commented that they liked the text book used (Han and Kamber 2006), as it allowed them to read more about the concepts that were only covered briefly in the lectures.

6.2.2 Interestingness and Usefulness of Topics

The results on interestingness and usefulness of the topics covered in the course are shown in Figure 5. A statistically significant difference was that the external participants agreed stronger that the data preprocessing and integration topics were more interesting and useful compared to the masters students. This is likely to be because the external participants had experience with real world data issues beforehand, and are thus more aware of the importance of these topics. Other significant differences were that the students with previous data mining experience rated the time-series and data streams topic, as well as the *End-to-end data mining* and social impacts lectures, more interesting and useful than those without previous experience. With the exception of privacy-preserving data mining and Web and text mining, the external participants rated all other topics to be more interesting and useful than the masters students did. Those with previous data mining experience also rated all topics, except privacy-preserving data mining, more interesting and useful than those without previous experience.

6.2.3 Coverage of Topics

As Figure 6 shows, the only statistically significant difference with regard to the coverage of topics was that the students with previous data mining experience wanted to hear more about data stream and time-series data mining. Other differences included that masters students mainly wanted to have more coverage of clustering, classification and prediction, as well as text and Web mining. The external participants, on the other hand, would have preferred to also learn more about the data related first two topics, and classification and prediction. Students without previous data mining experience would have liked more detailed coverage of the classification and prediction topics, while those with previous experience would have preferred increased coverage of almost all topics, with the exception of privacy-preserving data mining and the initial data mining process and data issues topics. They were very likely already famil-

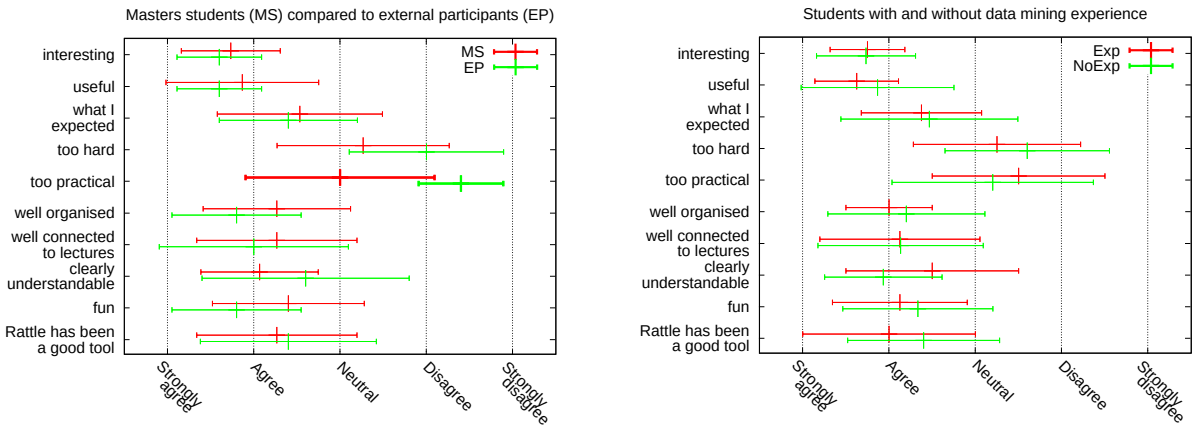


Figure 7: The laboratory sessions were...

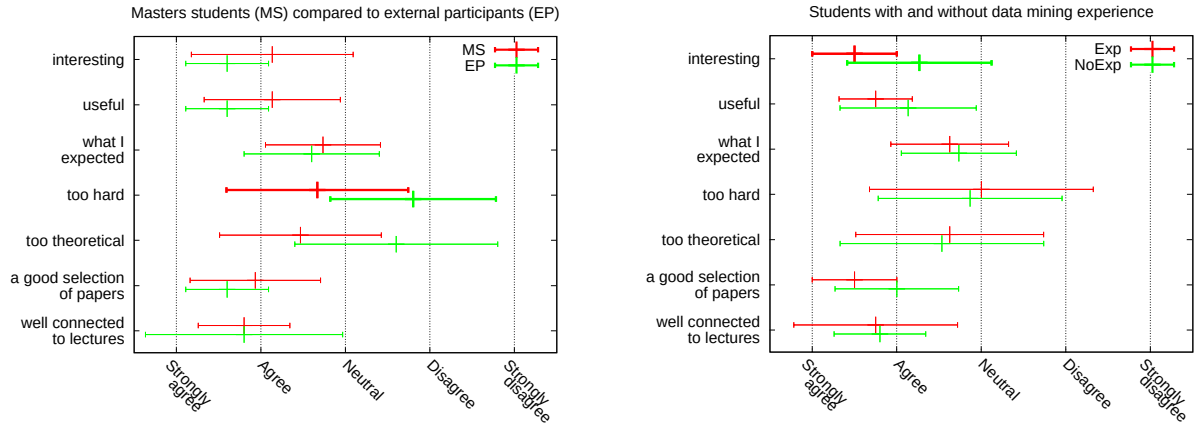


Figure 8: The tutorials were...

iar with the introductory issues discussed in this first topic. Some students commented in the questionnaire that the *End-to-end data mining* lecture was one of the best parts of the course.

Several students wrote in the questionnaire that they would have liked topics to be covered in more detail rather than just having overviews of many concepts (such as introductions to many recently developed clustering techniques). Others would have preferred to hear about the core data mining techniques in more depth. One way to allow more detailed coverage in the future will be to increase the number of lectures. Mentioned several times in the questionnaire was the wish to include more practical examples into the lectures. Some students would also have liked to either have more basic explanations of the mathematical and statistical concepts, or provision of resources that contained such background material.

6.2.4 Practical Laboratory Sessions

The results about laboratory sessions in Figure 7 show a clear distinction between masters students and external participants, in that the latter group would have preferred the laboratories to be both more practical (a statistically significant difference) and harder. On the other hand, the external participants found the laboratory sessions to be less understandable, which was likely due to a different computing environment used at the university compared to their workplaces, as well as that the masters students were more familiar with the practice of computer science laboratory sessions. There were no significant differences between those students with previous data mining experience and those without. However, those with previous experience found the laboratory sessions to be less understandable than those without experience. This might be due to the fact that they have been using a different data mining software previously, and were unfamiliar with the *Rattle* tool used in this course.

Several students criticised the use of *Rattle* as the only data mining tool presented in the course. They would have preferred to learn more about other available tools, or even use different tools in the laboratory sessions and assignments. Other students commented that they would have liked more documentation on the functionality of the software, including detailed explanations of how parameter settings influence algorithms. Some students also wrote that they would have liked to have practical demonstrations of data mining tools in the lectures. Another criticism was that the initial installation of the *Rattle* tool had some minor deficiencies at the beginning of the semester and was not very stable in several instances (a potential risk when using any open source tool that is still under development).

6.2.5 Research Paper Tutorials

The main statistically significant differences in feedback on the tutorials, as shown in Figure 8, are that the masters students found these tutorials much harder than the external participants, and that the

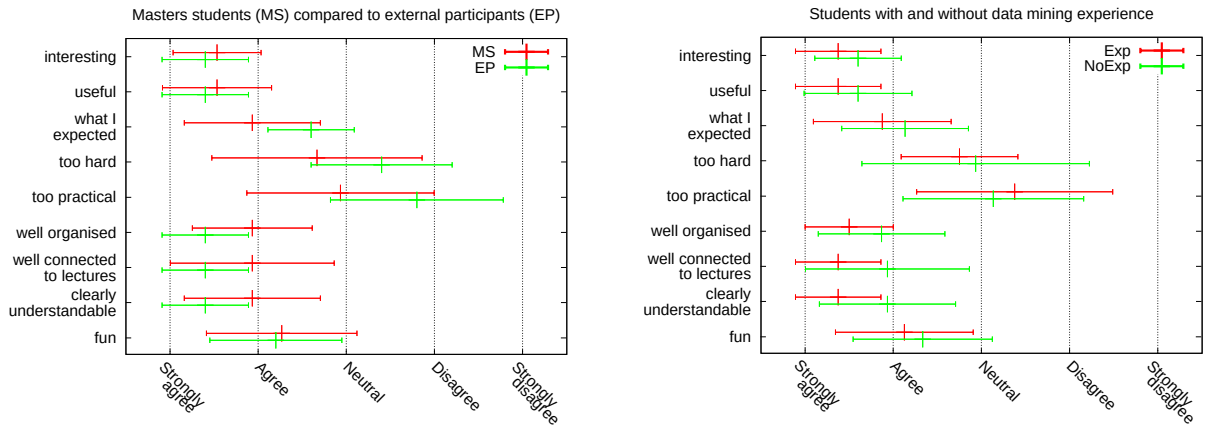


Figure 9: The assignments were...

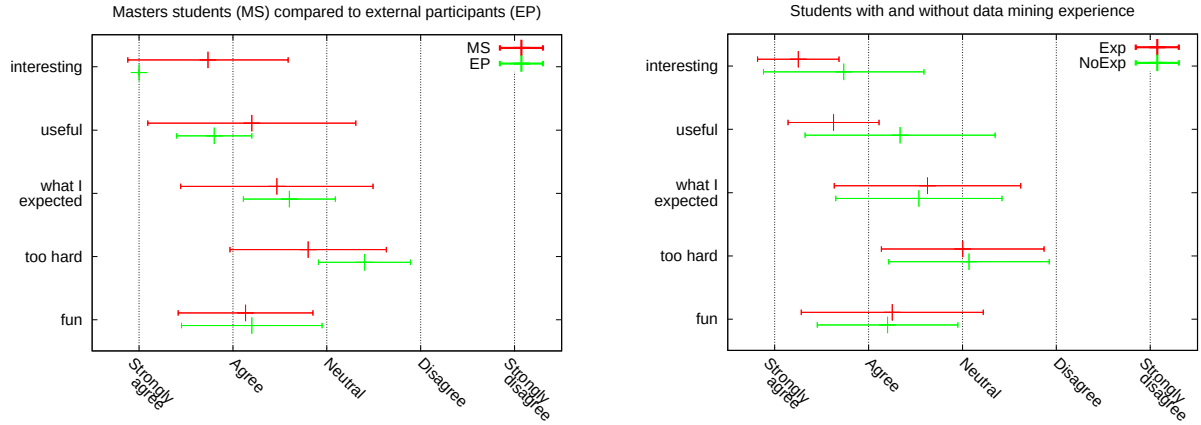


Figure 10: The paper presentation was...

students with previous data mining experience found the tutorials to be more interesting than those without previous experiences. The external participants also reported that reading these papers was more useful and interesting than the masters students did, and they also found these papers to be less theoretical to read than the masters students. On the other hand, those with and without previous data mining experience both reported similar agreements with regard to how hard and how theoretical the papers were. However, those with previous experience agreed stronger that the papers selected were appropriate, and they also found the tutorials to be more useful.

One explanation for these differences is that, due to students other commitments and room limitations, two tutorial groups were run, the first with ten students (all of them masters students) while the second group with sixteen students included all external participants. This resulted in rather different discussions of the tutorial topics covered. For future offerings of such tutorials based on paper readings, either a more balanced distribution of students should be aimed for, or even better only one single tutorial group that would include all students. Comments provided in the questionnaires included that students would have liked the tutorials to be more connected to both lectures and the text book, to only include papers with less mathematical and statistical content, as well as that specific questions were provided by the lecturer for each paper to make the reading more specific. Some students would also have preferred more practical laboratory sessions rather than tutorials.

6.2.6 Assignments

The feedback on the two assignments, shown in Figure 9, is mainly positive. There are no statistically significant differences between the two pairs of student sub-groups. The main differences between masters students and external participants were that the former found the assignments harder but practical enough, while the external participants would have preferred to have assignments of a more practical form. The external participants were also more agreeing that the assignments were well organised and understandable, as well as better connected to the lectures than the masters students.

On the other hand, masters students were more agreeing that the assignments were what they expected. This is likely due to their experience of assignments in other courses they attend, while the external participants only attended this one course and thus had less experience in what to expect. Those with previous data mining experience found the assignments less hard than those without previous experience, but better organised and understandable, and better connected to the lectures.

Student comments on the assignments included that they would have liked more freedom in the use of data sets (for example, one external participant would have liked to use a data set from his workplace), with less emphasis on writing reports and more practical programming in the assignments.

6.2.7 Research Paper Presentation

The results for the paper presentation in Figure 10 are quite different for masters students and external participants, although none of these differences are statistically significant. Masters students found the work involved and giving a presentation harder but also less useful and less interesting. A comment from one external participant indicates that the large number of 26 presentations provided him with an exposure to a wide range of topics, including several papers in his area of interest. Similar differences in interestingness and usefulness can also be seen between those with and without previous data mining experience (again not statistically significant).

Some students critically commented in the questionnaire that the five minutes time given for a presentation was too short, and having the second assignment and student presentation both in the last semester week was a very high workload. Others would have preferred the presentations to be linked better with the tutorial sessions, as well as to have student presentations spread throughout the semester rather than having all of them in the last week.

7 Conclusions

This paper has discussed the student population, course structure and assessment of a new graduate level data mining course taught at a major Australian university in 2007, and presented an empirical evaluation of student perception of the course. Overall, the feedback was positive, with students reporting that the course has been interesting, useful, and that it contained about the right levels of theory and practice. They also reported to have liked the assessment consisting of two project based assignments and one presentation of a data mining research paper.

The results of the empirical evaluation indicate that it is possible to run a data mining course with both masters students and participants from industry and government organisations, even though these two groups likely have very different backgrounds of prior knowledge and experiences, and also different expectations in such a course. External participants, some with considerable expertise in data management and processing, were especially active during tutorial discussions of research papers, and contributed with their practical knowledge and experience. This was enriching the course experience for the students that so far did not have industry experience.

For future offerings of the data mining course described in this paper, there are several issues that can be improved. First, increasing the number of lectures will allow coverage of topics in more details. Second, having had two tutorial groups, one with only masters students, was clearly not optimal. A better arrangement would have been one single tutorial group only, to facilitate discussions and exchange of experiences and ideas between external participants and graduate students. Additionally, in order to make students think more actively and critically about the tutorial papers, it will be useful to ask them to provide written questions about these papers before the tutorial sessions, similar to (Musicant 2006). Third, allowing more choice in both selection of data sets and data mining tools for their assignments would allow students to better explore areas of their interest.

8 Acknowledgements

The author would like to thank the students of COMP8400 (semester 1, 2007) for providing their feedback on this course, and to Tom Gedeon and Paul Thomas for providing valuable feedback on a draft version of this paper and for proof-reading it.

References

Agrawal, R. & Srikant, R. (1994), Fast algorithms for mining association rules in large databases, *in* 'International Conference on Very Large Data Bases', Santiago de Chile, Chile, pp. 487–499.

Han, J. & Kamber, M. (2006), *Data Mining: Concepts and Techniques*, 2nd edition, Morgan Kaufmann.

Hand, D.J. (2006), 'Classifier technology and the illusion of progress', *Statistical Science*, vol. 21, no. 1, pp. 1–14.

Lopez, D. & Ludwig, L. (2001), Data mining at the undergraduate level, *in* 'Midwest Instruction and Computing Symposium', Cedar Falls, Iowa.

Musicant, D.R. (2006), A data mining course for computer science: Primary sources and implementations, *in* 'SIGCSE-06: Proceedings of the 37th SIGCSE technical symposium on computer science education', ACM Press, Houston, Texas, pp. 538–542.

Newman, D.J., Hettich, S., Blake, C.L. & Merz, C.J. (1998), UCI Repository of machine learning databases, `http://www.ics.uci.edu/~mlearn/MLRepository.html`, University of California, Department of Information and Computer Science, Irvine, California.

Patman, F. & Thompson, P. (2003), Names: A new frontier in text mining, *in* 'First NSF/NIJ Symposium (ISI-2003)', Tucson, AZ, Springer LNCS 2665, pp. 27–38.

Rahm, E. & Do, H.H. (2000), 'Data Cleaning: Problems and Current Approaches', *IEEE Data Engineering Bulletin*, vol. 23, no. 4, pp. 3–13.

Salzberg, S.L. (1997), 'On comparing classifiers: Pitfalls to avoid and a recommended approach', *Data Mining and Knowledge Discovery*, Springer, vol. 1, no. 3, pp. 317–328.

Saquer, J. (2007), 'A data mining course for computer science and non-computer science students', *J. Comput. Small Coll.* , vol. 22, no. 4, pp. 109–114, Consortium for Computing Sciences in Colleges.

Sheskin, D.J. (2004), *The Handbook of Parametric and Nonparametric Statistical Procedures*, 3rd edition, Chapman & Hall/CRC.

Tan, P.N., Kumar, V. & Srivastava, J. (2002), Selecting the right interestingness measure for association patterns, *in* 'ACM SIGKDD international conference on knowledge discovery and data mining', Edmonton, Canada, pp. 32–41.

Verykios, V.S., Bertino, E., Fovino, I.N., Provenza, L.P., Saygin, Y., & Theodoridis, Y. (2004), 'State-of-the-art in privacy preserving data mining', *SIGMOD Record*, vol. 33, no. 1, pp. 50-57.

Williams, G.J. (2007), 'Data Mining with Rattle and R', Togaware, Canberra, `http://datamining.togaware.com/survivor/`.

Winkler, W.E. (2004), 'Methods for evaluating and creating data quality', *Information Systems*, Elsevier, vol. 29, no. 7, pp. 531–550.

Witten, I.H. & Frank, E. (2000), *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann.

Author Index

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 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, *University of Queensland*. 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 (AUIC2008), 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 and John Yearwood. 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 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*, Wenyuan 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 85 - Advances in Ontologies 2007
Edited by Thomas Meyer, *Meraka Institute, South Africa* and Abhaya Nayak, *Macquarie University, Australia*. December, 2007. 978-1-920682-66-8.
Contains the proceedings of the 3rd Australasian Ontology Workshop, Gold Coast, Queensland, Australia.