Conferences in Research and Practice in
Information Technology

Volume 121

Data Mining and Analytics 2011 (AusDM'11)

,

Data Mining and Analytics 2011

Proceedings of the
Ninth Australasian Data Mining Conference
(AusDM'11), Ballarat, Australia,
1–2 December 2011

Peter Vamplew, Andrew Stranieri, Kok–Leong Ong,
Peter Christen and Paul J. Kennedy, Eds.

Volume 121 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 2011. Proceedings of the Ninth Australasian Data Mining Conference (AusDM'11), Ballarat, Australia, 1–2 December 2011

Conferences in Research and Practice in Information Technology, Volume 121.

Editors:

Peter Vamplew
Centre for Informatics and Applied Optimisation
School of Science, Information Technology and Engineering (SITE)
University of Ballarat
University Drive, Mount Helen, PO Box 663
Ballarat, Vic. 3353, Australia
Email: p.vamplew@ballarat.edu.au

Andrew Stranieri
Centre for Informatics and Applied Optimisation
School of Science, Information Technology and Engineering (SITE)
University of Ballarat
University Drive, Mount Helen, PO Box 663
Ballarat, Vic. 3353, Australia
Email: a.stranieri@ballarat.edu.au

Kok–Leong Ong
School of Information Technology
Deakin University
Burwood, Vic. 3125, Australia
Email: leong@deakin.edu.au

Peter Christen
Research School of Computer Science
ANU College of Engineering and Computer Science
The Australian National University
Canberra ACT 0200, Australia
Email: peter.christen@anu.edu.au

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

Series Editors:
Vladimir Estivill-Castro, Griffith University, Queensland
Simeon J. Simoff, University of Western Sydney, NSW
Email:crpit@scm.uws.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 121.
ISSN 1445-1336.
ISBN 978-1-921770-02-9.

Document engineering by CRPIT, November 2011.

Table of Contents

Preface

We are delighted to welcome you to the Ninth Australasian Data Mining Conference (AusDM'11) being held this year in Ballarat, Victoria. AusDM started in 2002 and is now the annual flagship meeting for data mining and analytics professionals in Australia. Both scholars and practitioners present the state–of–the–art in the field. Endorsed by the peak professional body, the Institute of Analytics Professionals of Australia, AusDM has developed a unique profile in nurturing this joint community. The conference series has grown in size each year from early events held in Canberra (2002, 2003), Cairns (2004), Sydney (2005, 2006), the Gold Coast (2007), Glenelg (2008) and Melbourne (2009). This year's event has been supported by

- Togaware, again hosting the website;
- University of Ballarat, for providing the venue, registration facilities, for coordinating the conference management system, the review process and other essential expertise;
- Deakin University, for sponsoring the best paper award;
- the Institute of Analytic Professionals of Australia (IAPA) for facilitating the contacts with the industry;
- the Australian Computer Society, for publishing the conference proceedings;
- data mining students from University of Ballarat for their local support.

The conference program committee reviewed 50 submissions, out of which 22 submissions were selected for publication and presentation. This was an acceptance rate of 44%. AusDM follows a rigid blind peer-review process and ranking-based paper selection process. All papers were extensively reviewed by at least three referees drawn from the program committee. We would like to thank all those who submitted their work to the conference. We will continue to extend the conference format to be able to accommodate more presentations.

In addition, three keynote speakers were invited. Associate Professor Christopher W. Clifton from Purdue University talked about advances in a decade's worth of research in privacy–preserving data mining and assessed its impact. Professor Hussein Abbass from the University of New South Wales gave a talk about mining "big data streams", a timely issue about the future of data mining. Dr Musa Mammadov from the University of Ballarat and the National ICT Australia discussed approaches towards mining drug-drug interactions.

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

Jiuyong Li
University of South Australia

Peter Christen
The Australian National University

Paul J. Kennedy
University of Technology, Sydney

AusDM 2011 Programme Chairs
December 2011

Conference Organisation

Program Chairs

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

Conference Chairs

John Yearwood, University of Ballarat, Ballarat
KokLeong Ong, Deakin University, Victoria
Jiuyong Li, University of South Australia, Adelaide

Other Conference Committee Members

David Yost, University of Ballarat, Ballarat
Peter Vamplew, University of Ballarat, Ballarat
Long Jia, University of Ballarat, Ballarat
Richard Dazeley, University of Ballarat, Ballarat
Sitalakshmi Venkatraman, University of Ballarat, Ballarat
Shamsul Huda, University of Ballarat, Ballarat
Sally Firmin, University of Ballarat, Ballarat
Sunam Pradhan, University of Ballarat, Ballarat
Zari Dzalilov, University of Ballarat, Ballarat

Steering Committee Chairs

Simeon Simoff, University of Western Sydney
Graham Williams, Australian Taxation Office

Other Steering Committee Members

Peter Christen, Australian National University, Canberra
Paul Kennedy, University of Technology, Sydney
Kok-Leong Ong, Deakin University, Victoria
John Roddick, Flinders University, Adelaide

Program Committee

Hussein Abbass, University of New South Wales @ADFA, Australia
Adil Bagirov, University of Ballarat, Australia
Rohan Baxter, CSIRO ICT Centre, Australia
Peter Christen, The Australian National University, Australia
Eugene Dubossarsky, Analyst First, Australia
Vladimir Estivill-Castro, Griffith University, Australia
Junbin Gao, Charles Stuart University, Australia
Raj Gopalan, Curtin University of Technology, Australia
Warwick Graco, Australia Tax Office
Lifang Gu, Australian Tax Office, Australia
Ping Guo, Beijing Normal University, China

Robert Hilderman, University of Regina, Canada
Paul Kennedy, University of Technology, Sydney, Australia
Paul Kwan, University of New England, Australia
Vincent Lee, Monash University, Australia
Bradley Malin, Vanderbilt University, USA
Musa Mammadov, University of Ballarat, Australia
Arturas Mazeika, Max-Planck-Institut fr Informatik, Germany
Sumudu Mendis, Strategic Information Design and Governance, Department of Human Services, Australia
Richi Nayak, Queensland University of Technology, Australia
Christine O'Keefe, CISRO, Australia
Tom Osborn, The Leading Edge, Sydney
Robert Pearson, Health Insurance Commission, Australia
Francois Poulet, University of Rennes, France
David Taniar, Monash University
Peter Vamplew, University of Ballarat, Australia
Warren Jin, CSIRO Mathematics, Informatics and Statistics, Australia
Ting Yu, The University of Sydney, Australia
Huaifeng Zhang, Centrelink, Australia
Ji Zhang, The University of Southern Queensland, Australia
Yanchang Zhao, RDataMining.com, Australia

AusDM Sponsors

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

http://www.togaware.com

http://www.iapa.org.au

Centre for Informatics and Applied Optimization
http://guerin.ballarat.edu.au/ard/itms/CIAO/ciao.shtml

Deakin University
http://www.deakin.edu.au/

University of Ballarat
http://www.ballarat.edu.au/

Keynotes

Privacy–Preserving Data Mining at 10: What's Next?

Christopher W. Clifton

Department of Computer Science, Purdue University,
305 N. University Street, West Lafayette, IN 47907, USA
Email: clifton@cs.purdue.edu

Abstract

It has been 10 years since the first papers entitled "Privacy–Preserving Data Mining" appeared. The past decade has witnessed a flood of papers with new techniques to protect and mine data, but little real–world impact. Is privacy dead? Or are there still challenges that must be addressed?

This talk will begin with a brief retrospective on advances in privacy–preserving data mining. We will then look at legal and societal issues, and the mismatches between these issues and the technology. We will then look at some new research in the area, including privacy metrics and techniques applicable to data mining, and data/computation outsourcing.

Mining Big Data Streams: The Fallacy of Blind Correlation and the Importance of Models

Hussein Abbass

University of New South Wales at the Australian Defence Force Academy,
Northcott Drive, Canberra ACT 2600, AUSTRALIA
Email: h.abbass@adfa.edu.au

Abstract

Big data streams mark a new era in artificial intelligence and the data mining literature. Video and voice streams have grown rapidly in recent years. A single lab–based human–computer interaction experiment with one human subject collecting Cognitive, Physiological, and other data can easily generate a few terabytes of data in a single hour; growing rapidly to a Petabyte within a timeframe less than a month. In an article in the Wired Magazine, 2008, by Chris Anderson, he wrote "the data deluge makes the scientific method obsolete". He predicted that in the age of Petabyte and beyond, a meaningful correlation analysis is enough! Chris comment was provocative; but some started believing it. So was Chris right or wrong? Why? What can we do to face the outburst of big data? Do we have the data mining tools to manage these data? Where is the future of data mining heading? In this talk, I will discuss the above questions and demonstrate some answers using examples of my work and analysis.

Drug–drug interactions: A Data Mining Approach

Musa Mammadov

Graduate School of Information Technology and Mathematical Sciences
University of Ballarat, 1 University Drive, Mt. Helen,
Ballarat, Victoria, 3350, AUSTRALIA
Email: m.mammadov@ballarat.edu.au

Abstract

Drug–drug interaction is one of the important problems of Adverse Drug Reaction (ADR). This presentation describes a data mining approach to this problem developed at the University of Ballarat. This approach is based on drug–reaction relationships represented in the form of a vector of weights; each vector related to a particular drug can be considered as a pattern in causing adverse drug reactions. Optimal patterns for drugs are determined as a solution to some global optimization problem. Although this approach can be used for solving many ADR problems, we concentrate only on drug–drug interactions. The numerical implementations are carried out on different classes of reactions from the Australian Adverse Drug Reaction Advisory Committee (ADRAC) database. The results obtained extend our understanding of the drug–drug interaction from a data mining point of view.

Contributed Papers

Understanding Risk Factors in Cardiac Rehabilitation Patients with Random Forests and Decision Trees

Alina Van[1] Valerie C. Gay[1] Paul J. Kennedy[1] Edward Barin[2] Peter Leijdekkers[1]

[1]Faculty of Engineering and Information Technology
University of Technology, Sydney
PO Box 123, Broadway 2007, New South Wales, Australia

[2]Department of Cardiology
Royal North Shore Hospital
Reserve Road, St Leonards 2065, New South Wales, Australia

alina.van@student.uts.edu.au, valerie.gay@uts.edu.au, paul.kennedy@uts.edu.au,
edward.barin@gmail.com, peter.leijdekkers@uts.edu.au

Abstract

Cardiac rehabilitation is a well-recognised non-pharmacological intervention recommended for the prevention of cardiovascular disease. Numerous studies have produced large amounts of data to examine the above aspects in patient groups. In this paper, datasets collected for over a 10 year period by one Australian hospital are analysed using decision trees to derive prediction rules for the outcome of phase II cardiac rehabilitation. Analysis includes prediction of the outcome of the cardiac rehabilitation program in terms of three groups of cardiovascular risk factors: physiological, psychosocial and performance risk factors. Random forests are used for feature selection to make the models compact and interpretable. Balanced sampling is used to deal with heavily imbalanced class distribution. Experimental results show that the outcome of phase II cardiac rehabilitation in terms of physiological, psychosocial and performance risk factor can be predicted based on initial readings of cholesterol level and hypertension, level achieved in six minute walk test, and Hospital Anxiety and Depression Score (HADS) anxiety score and HADS depression score respectively. This will allow for identifying high risk patient groups and developing personalised cardiac rehabilitation programs for those patients to increase their chances of success and minimize their risk of failure.

Keywords: cardiac rehabilitation, decision trees, random forests, feature selection, balanced sampling.

1 Introduction

Cardiovascular disease is the leading cause of death in the majority of developing and developed countries. Although the healthcare industry has greatly advanced in detection and treatment of most heart diseases, heart failure continues to produce a heavy burden of cardiovascular morbidity and mortality in the majority of industrialised countries (Davies et al. 2010; Lavie, Milani and Ventura 2009; Noy 1998; Rivett et al. 2009).

Cardiac rehabilitation, the care of patients with heart diseases, was defined by the World Health Organisation as 'the sum of activities required to influence favourably the underlying cause of the disease, as well as the best possible physical, mental and social conditions, so that they may, by their own efforts, preserve or resume when lost, as normal a place as possible in the community' (Noy 1998). It averts the recurrence of cardiovascular events, and increases life expectancy. Particularly, the lowering of cardiovascular morbidity and mortality risk, at least to a certain extent, is ascribed to an increase in exercise capacity (Hansen et al. 2010). Thus, current guidelines highlight the importance of physical activity in cardiac rehabilitation, and ascertain exercise training as a recognised non-pharmacological intervention recommended for both the primary (avoidance of the development of a disease) and secondary (early disease detection) prevention of heart diseases (Guiraud et al. 2010). Importantly, education, counselling and behavioural interventions to promote lifestyle change and modify risk factors have become an increasingly significant part of cardiac rehabilitation programs (Goble and Worcester 1999).

Cardiac rehabilitation conventionally consists of phase I (inpatient cardiac rehabilitation), phase II (outpatient cardiac rehabilitation), and phase III (maintenance). It starts with an inpatient hospital-based program which is delivered on an individual basis or to groups of patients. Due to the short hospital stays and time-consuming examinations, phase I programs are mostly limited to early mobilisation and education and do not include an exercise training component. Furthermore, it is recognised that inpatient cardiac rehabilitation may be ineffective because of the psychological state of patients soon after the acute event. Outpatient hospital-based programs last from two to four months. The content of phase II cardiac rehabilitation varies greatly from hospital to hospital. It usually includes group exercises, education and counselling. In the maintenance phase, exercise training and heart disease risk control 'are supported in minimally supervised or unsupervised setting' (Goble and Worcester 1999). Maintenance programs are even more varied in

content and structure than outpatient cardiac rehabilitation programs. In the case of phase III cardiac rehabilitation, patients may receive further education, psychosocial support and exercise classes. Patients may also be regularly reviewed by a physician.

This study focuses on phase II cardiac rehabilitation and investigates the importance of physiological, psychosocial and performance risk factors of heart disease in terms of the prognostic value for cardiac events. Physiological factors include cholesterol level, body mass index (BMI), waist circumference, hypertension, smoking and diabetes. Psychosocial risk factors include anxiety and depression measured using Hospital Anxiety and Depression Scale (HADS) (Marques Marcolino 2007). Performance risk factors include level and metres achieved in six minute walk test. The primary objective of this study is to build a classification model that can be used to predict the outcome of the phase II cardiac rehabilitation program in terms of physiological, psychosocial and performance risk factors. Note that only the best prediction models for each group of risk factors were included in this paper. Section 2 presents related work in the application domain. Section 3 describes data preparation procedures. Section 4 illustrates results of initial data exploration followed by section 5 which explains methodology used in this study. Section 6 presents results and discusses the experimental results, medical interpretation of results and variable importance results. Lastly, section 7 concludes the paper and gives recommendations for future work.

2 Related Work

Existing research affirms that physiological factors, such as cholesterol level, BMI, waist circumference, hypertension, diabetes and smoking, have a considerable effect on morbidity and mortality in cardiac patients. Physiological benefits of exercise training include improvements in blood lipid parameters, blood haemodynamics, body anthropometrics, peak oxygen intake, exercise capacity and functional status. These factors have a great prognosis value on progression of cardiovascular disease and, therefore, have been examined in many trials (Austin et al. 2005; Austin et al. 2008; Brubaker et al. 2009; Delagardelle and Feiereisen 2005; Kravari et al. 2010). The most important behavioural risk factors of cardiovascular disease and cerebrovascular disease are: unhealthy diet, physical inactivity and tobacco use. These risk factors are responsible for about 80% of coronary heart disease and cerebrovascular disease (World Health Organisation 2011).

Psychosocial factors, such as quality of life, depression, somatisation, anxiety and hostility, have considerable effect on morbidity and mortality in cardiac patients. Cardiac patients can experience discomfort in their everyday activities and health-related quality of life because of their restricted heart capacity. Additionally, this can reduce patients' ability to exercise, which can further reduce physical fitness making their symptoms even worse. Deteriorated quality of life, depression and anxiety have harmful effects 'not only on daily social, domestic, work and leisure activities but also on rehospitalisation and death rates' (Kulcu et al. 2007).

Importantly, a review by Davies et al., conducted in 2010, draws attention to the fact that the risk of death with exercise in people with mild to moderate heart disease did not change after the cardiac rehabilitation program. However, there was a reduction in hospital re-admissions. It has been also recognised that in both the short and long term, exercise training programs improve health-related quality of life compared to usual care without exercise (Almerud Osterberg et al. 2010; Chester 2006; Nilsson et al. 2008; Noy 1998; Piperidou and Bliss 2008).

Recent studies have also reported decreased levels of depression and anxiety in cardiac patients as a result of exercise training (Austin et al. 2005; Davies et al. 2010; Kulcu et al. 2007). A study by Lavie, Milani and Ventura, conducted in 2009, asserts that even a slight improvement in peak oxygen intake leads to improvements in depression and anxiety scores and, therefore, reduces the risk of acute events or hospital readmissions. This aligns with the results of another clinical trial that observed a significant reduction in hospitalisations following improvements in health-related quality of life in the exercise training group compared to the control group (Davies et al. 2010). Importantly, existing research suggests that cardiac rehabilitation is particularly effective at improving health-related quality of life in the long term (Austin et al. 2005).

While the physiological benefits help impede progression of heart disease, exercise capacity and functional status play an important role in patients' health-related quality of life and, thereby, the probability of an acute event (Brubaker et al. 2009; Delagardelle and Feiereisen 2005; Kravari et al. 2010). Numerous studies indicate improvements in functional capacity and exercise tolerance as a result of exercise training. Austin et al. (2008) notes the long term benefit of cardiac rehabilitation in terms of walking distance and perceived exertion. Exercise training has also been shown to decelerate the deterioration from baseline performance which contributes to the main goal of cardiac rehabilitation - enhancement and sustainability of functional performance. Another study, conducted in 2010, observed significant improvements in symptomatology, such as: breathlessness and fatigue; as well as the skeletal muscle metabolism; peripheral inflammatory markers; and exercise capacity in chronic heart failure patients (Kravari et al. 2010).

Importantly, it has been recognised that in chronic heart failure patients, the lack of improvement in exercise capacity after an exercise training program has strong prognostic value for cardiac events independent of other existing symptoms. A study by Tabet et al. (2008) suggests that patients who do not significantly improve in exercise capacity after the cardiac rehabilitation program should be carefully monitored.

The discussion above recognises that physiological, psychosocial and performance risk factors have a great prognosis value on progression of cardiovascular disease. Consequently, they have been a subject of many trials and studies. A data mining experiment by Kajabadi, Saraee and Asgari (2009) attempted to predict low density lipoprotein in a population of 1800 people by means of decision trees, namely Classification and Regression Trees (CART) and achieved 77.6% accuracy. The most important variable found for classification was cholesterol

level. Other important variables for classification were age, BMI, apolipoproteins, triglycerides level, and smoking.

Another study used a combination of logistic regression, C5.0 decision tree, CHAID decision tree, CART, exhaustive CART and discriminant analysis to select the risk factors of hypertension and hyperlipidemia and achieved a combined accuracy of 93.07% and sensitivity of 98.76%. It found blood pressure, triglycerides, BMI, gender, age, glutamate pyruvate transaminase (GPT) and uric acid (UA) as risk factors significant in predicting hypertension. Total cholesterol, triglycerides and systolic blood pressure were listed as the hyperlipidemia indicators (Chang, Wang and Jiang 2011).

Also, Smith et al. (2010) found that natriuretic peptides improve prediction of incident heart failure and atrial fibrillation in the general population in addition to conventional risk factors by using regression analysis on a cohort of over 5000 patients.

Assessment of cardiac risk factors using decision trees established that 'the most important risk factors, as extracted from the classification rules analysis were: 1) for myocardial infarction (MI), age, smoking, and history of hypertension; 2) for percutaneous coronary intervention (PCI), family history, history of hypertension, and history of diabetes; and 3) for coronary artery bypass graft surgery (CABG), age, history of hypertension, and smoking' (Karaolis et al. 2010). The accuracies achieved were 66%, 75%, and 75% for the MI, PCI, and CABG models, respectively.

To summarise, several studies show that data mining algorithms assist in the identification of high and low risk patient groups.

3 Data Preparation

3.1 Data Collection

Patient data used for this study was acquired from the cardiac rehabilitation unit of an Australian hospital. It contains information on 3931 patients collected as part of phase II and phase III cardiac rehabilitation program for the period between 10/01/2000 and 01/03/2011. During that time content of the program has not changed.

3.2 Data Integration

The data was combined into a single table over a number of steps. Firstly, CLIENTS, CARDIAC DATA and REHAB SESSIONS tables were joined based on the CLIENT ID unique identifier (see Figure 1).

Secondly, sub-tables containing information on various characteristics of cardiac rehabilitation were joined on unique key and added to the main table. For instance, MEDICATION TYPES and MEDICATION DESCRIPTIONS joined on medication id and medication type id. Thirdly, data in multi-value fields was aggregated into a single value to avoid oversampling of particular records. For instance, BLOOD PRESSURE and HEART RATE measurements taken each cardiac session were each aggregated into a single row by separating the attribute into two attributes with suffixes "before" and "after". As a result, the final dataset contained one row for each unique CLIENT ID.

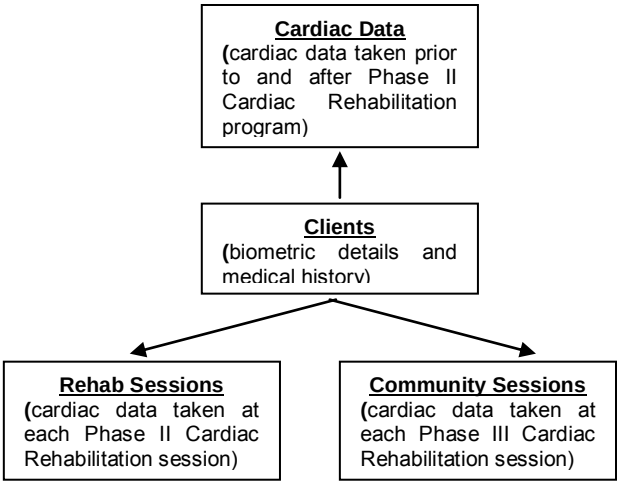


Figure 1: Cardiac Rehabilitation High-Level Database Structure

3.3 Data Cleaning

The original dataset contained missing values in several attributes. In cases where over 25% of values were missing, that attribute was discarded. Otherwise, missing values were replaced with a mean or mode for all samples in the same class for numeric or categorical attributes respectively.

Data points, where patients did not have data recorded post-rehab and where values in target attributes in each classification problem were missing, were also removed. This reduced the dataset from 3931 to 2280 records but ensured data integrity.

3.4 Data Transformation

In order to reduce number of attributes, certain attributes were aggregated into a single column. However, the dataset contains many highly imbalanced categorical attributes with a large number of values. Values of these attributes were re-coded to simplify the data. Lastly, attributes describing drug allergies and medications taken by patients were aggregated into categories and binarised into Boolean values.

To assist mining of the dataset and determine target variables, certain calculated fields were added based on existing fields in the cardiac dataset. Table 1 shows a subset of calculated attributes that is relevant for this paper.

3.5 Data Reduction

Fields identifying patients were omitted from the dataset to preserve privacy. Also, certain fields, identified as irrelevant, were omitted, as were attributes containing a large number of missing values.

4 Initial Data Exploration

After cleaning, the dataset contains biometric data (age, gender, height, and weight), socio-behavioural data (marital status, living arrangements, language, country of birth, smoking, and exercise type), medical history (cardio-vascular, cerebro-vascular, musculo-skeletal, respiratory and other conditions, and diabetes), drug history (types of medications taken and drug allergy),

Attribute	Description	Formula
Age	Age field determines person's age derived from given date of birth and date of entry into the rehabilitation program.	$Age = ProgramDate - DateOfBirth\,(years)$
Body Mass Index (BMI)	BMI was added to provide a measure of weight in relation to height.	$BMI = \frac{weight(kg)}{(height(m))^2}$
Cholesterol Level Change	Cholesterol Level Change is a categorical attribute which was added to encode changes in level of cholesterol in patients post-cardiac rehab.	$CLC_{0\%} = \begin{Bmatrix} \boldsymbol{S}(Success), if\ CL - CL_2 > 0 \\ \boldsymbol{N}(No\ Change), if\ CL - CL_2 = 0 \\ \boldsymbol{F}(Failure), if\ CL - CL_2 < 0 \end{Bmatrix}$ $CLC_{10\%} = \begin{Bmatrix} \boldsymbol{S}, if\ CL - CL_2 > 10\%\ CL \\ \boldsymbol{N}, if -10\%\ CL \leq CL_1 - CL_2 \leq 10\%\ CL \\ \boldsymbol{F}, if\ CL - CL_2 < -10\%\ CL \end{Bmatrix}$ $CLC_{20\%} = \begin{Bmatrix} \boldsymbol{S}, if\ CL - CL_2 > 20\%\ CL \\ \boldsymbol{N}, if -20\%\ CL \leq CL_1 - CL_2 \leq 20\%\ CL \\ \boldsymbol{F}, if\ CL - CL_2 < -20\%\ CL \end{Bmatrix}$
Hypertension	Hypertension is a numerical attribute that was added to illustrate ordinal nature of distribution of values of Hypertension Code attribute.	$Hypertension\,(H) = \begin{Bmatrix} \boldsymbol{4}, if\ Hypertension\ Code = HYPERTENSION\ UNCONTROLLED\ BY\ MEDICATION \\ \boldsymbol{3}, if\ Hypertension\ Code = HYPERTENSION\ CONTROLLED\ BY\ MEDICATION \\ \boldsymbol{2}, if\ Hypertension\ Code = HYPERTENSION\ UNMEDICATED \\ \boldsymbol{1}, if\ Hypertension\ Code = NO\ HYPERTENSION\ UNMEDICATED \end{Bmatrix}$
Hypertension Code Change	Hypertension Code Change is a categorical attribute that shows changes in Hypertension in patients post-cardiac rehab.	$HCC = \begin{Bmatrix} \boldsymbol{S}(Success), if\ H - H_2 > 0 \\ \boldsymbol{N}(No\ Change), if\ H - H_2 = 0 \\ \boldsymbol{F}(Failure), if\ H - H_2 < 0 \end{Bmatrix}$
Level Achieved Change	Level Achieved Change is a categorical attribute that shows changes in Level Achieved in six minute walk test in patients post-cardiac rehab.	$LAC = \begin{Bmatrix} \boldsymbol{S}(Success), if\ LA - LA_2 > 0 \\ \boldsymbol{N}(No\ Change), if\ LA - LA_2 = 0 \\ \boldsymbol{F}(Failure), if\ LA - LA_2 < 0 \end{Bmatrix}$
HADS Anxiety Change	HADS Anxiety Change is a categorical attribute that shows changes in HADS Anxiety score in patients post-cardiac rehab.	$AC = \begin{Bmatrix} \boldsymbol{S}(Success), if\ A - A_2 > 0 \\ \boldsymbol{N}(No\ Change), if\ A - A_2 = 0 \\ \boldsymbol{F}(Failure), if\ A - A_2 < 0 \end{Bmatrix}$
HADS Depression Change	HADS Depression Change is a categorical attribute that shows changes in HADS Depression score in patients post-cardiac rehab.	$DC = \begin{Bmatrix} \boldsymbol{S}(Success), if\ D - D_2 > 0 \\ \boldsymbol{N}(No\ Change), if\ D - D_2 = 0 \\ \boldsymbol{F}(Failure), if\ D - D_2 < 0 \end{Bmatrix}$

Table 1: Subset of calculated attributes (no index - pre-rehabilitation, index 2 - post-rehabilitation)

Number of objects = 2280	Mean	Median	Std Deviation	Missing Values
Age	65.35	66	10.98	0
Height	1.71	1.71	0.09	0
Weight	78.68	77.5	14.58	87
BMI	27.02	26.5	4.33	87
Cholesterol Level	4.88	4.8	1.27	548
Level Achieved	2.75	3	1.00	75
Metres Achieved	396.04	400	89.43	87
BP Systolic	124.56	124	18.32	1075
BP Diastolic	71.8	70	10.39	1076
HADS Anxiety	5.05	4	3.90	1166
HADS Depression	4.04	3	3.31	1166
Waist Circumference	97.51	97	13.15	1490
Weight2	78.81	78	14.80	251
BMI2	27.03	26.58	4.42	251
Cholesterol Level2	4.33	4.2	1.05	316
Level Achieved2	3.32	4	0.94	328
BP Systolic2	127.69	130	17.37	1073
BP Diastolic2	73.18	70	10.00	1074
HADS Anxiety2	4.3	4	3.47	1122
HADS Depression2	2.93	2	2.90	1123
Waist Circumference2	97	97	12.43	1495

Table 2: Summary of values of numeric attributes (no index - pre-rehabilitation, index 2 - post-rehabilitation)

cardiac data (blood pressure, heart rate, and blood lipid profile), psychosocial data (hospital anxiety and depression (HADS) scores) and exercise capacity data (level and metres achieved in six minute walk test). We conducted an initial data exploration to better understand the data and develop the most suitable framework for its analysis.

The dataset contains 19 ordinal attributes (4 categorical, 14 numeric and 1 date) and 30 nominal attributes (18 Boolean and 12 categorical). Table 2 summarises the properties of numeric attributes of the dataset and Table 3 summarises categorical attributes.

5 Methodology

The aim of this study is to explore aspects influencing the outcome of phase II cardiac rehabilitation in terms of physiological, psychosocial and performance risk factors of heart disease. Initially, we collated cardiac rehabilitation data and analysed it using data mining algorithms to build a classifier for each of the selected risk factors. The classifiers were then compared to select the best model for each group of risk factors. The selected model is used to predict the outcome of phase II cardiac rehabilitation and, consequently, to identify high risk patient groups likely to deteriorate post-rehab. The following sections describe methods used in more detail.

5.1 Algorithm Selection

Numerous supervised learning algorithms exist. To choose the most appropriate method, it is necessary to examine the nature of target and input attribute, the computational needs of the methods, the tolerance to missing values, outliers and small numbers of data points, and the model explicability (Linoff and Berry 2011). The cardiac rehabilitation data is characterised by high dimensionality with many missing values and, potentially, noise.

Considering these factors, the most appropriate method for our data is decision trees. Unlike linear regression, neural networks, Bayes learners and support vector machines (SVM), decision trees can handle heterogeneous high dimensional data with missing values (Han and Kamber 2006; Seni and Elder 2010; Tuffery 2011). They also are reasonable in terms of computational costs in comparison to lazy learners, neural networks and SVM. Moreover, decision trees are commonly used in mining data related to medical diagnosis, as they are a reliable and effective technique that provides high accuracy in classification problems and are easily interpretable (Podgorelec et al. 2002).

5.2 Classification Approach

Since our dataset does not contain information on patient outcome in terms of mortality, risk factors were used to measure success/failure of the program. These factors are represented by the attributes: CHOLESTEROL LEVEL, BMI, WAIST CIRCUMFERENCE, HYPERTENSION CODE, DIEBETES CODE, SMOKING CODE, HADS DEPRESSION, HADS ANXIETY, LEVEL ACHIEVED and METRES ACHIEVED. Patients' progress is measured as S (success), F (failure) or N (no change). Change in numeric attributes is calculated using 0%, 10% and 20% threshold, while change in categorical attributes is made possible due to ordinal nature of those features (see Table 1).

Attribute	Values
Sex	F,M
Marital Status	DE FACTO, DIVORCED, MARRIED, SEPARATED, SINGLE, UNKNOWN, WIDOWED
Living Arrangements	LIVING ALONE, LIVING WITH PARTNER, OTHER, SINGLE PERSON WITH FAMILY OR FRIENDS
Country Of Birth	AUSTRALIA AND EXTERNAL TERRITORIES, OTHER
Language	ENGLISH, OTHER
Referral Source	INPATIENT CARDIAC REHAB VISIT, OTHER
Medical Conditions (Cardio-Vascular, Cerebro-Vascular Musculo-Skeletal, Diabetes, Other Conditions, None of above Conditions)	TRUE, FALSE
Drug Allergy	TRUE, FALSE
Medications (ASP, BET, ANG, ACE, ST, CLO, SEI, DEP, Other Drugs)	TRUE, FALSE
Reason for Visit/ Reason for Visit2	CARDIAC REHAB ADMISSION, OTHER
Medical Classification/ Medical Classification2	CORONARY DISEASE, HEART FAILURE, OTHER, SURGERY
Risk Stratification/ Risk Stratification2	H, M, L
Cholesterol Code/ Cholesterol Code2	CHOLESTEROL GREATER THAN 4.5 - MEDICATED, CHOLESTEROL GREATER THAN 4.5 - UNMEDICATED, CHOLESTEROL LESS THAN 4.5 - MEDICATED, CHOLESTEROL LESS THAN 4.5 - UNMEDICATED, CHOLESTEROL UNKNOWN
Cholesterol Type/ Cholesterol Type2	CHOLESTEROL, CHOLESTEROL UNKNOWN, NO CHOLESTEROL
Hypertension Code/ Hypertension Code2	HYPERTENSION CONTROLLED BY MEDICATION, HYPERTENSION UNCONTROLLED BY MEDICATION, HYPERTENSION UNMEDICATED, NO HYPERTENSION UNMEDICATED
Family History/ Family History2	TRUE, FALSE
Smoking Current/ Smoking Current2	TRUE, FALSE
Smoking Code/ Smoking Code2	CURRENTLY SMOKING, NEVER SMOKED, STOPPED SMOKING LESS THAN 3 YEARS AGO, STOPPED SMOKING MORE THAN 3 YEARS AGO
Exercise Type/ Exercise Type2	WALKING, OTHER
Psychosocial Difficulties/ Psychosocial Difficulties2	Y, N

Table 3: Summary of values of nominal attributes (no index - pre-rehabilitation, index 2 - post-rehabilitation)

Decision tree induction was selected for classification of our data. Several optimisation techniques can be used to improve the effectiveness of decision tree models. The challenge with this cardiac dataset comes from its high dimensionality and the fact that the number of data points

in the target classes is unbalanced.

High dimensionality of the data may be addressed with feature selection. Moreover, systematic analysis of the importance of different variables provides deep insights into the different contributions of those features towards classification and is necessary for developing effective prediction models (Pan and Shen 2009).

Random forests are an appropriate method for both feature selection and for understanding the importance of various features. Their performance is generally superior to single decision trees. Random forest consists of many decision tree predictors with randomly selected variable subsets (there is a different subset of training and validation data for each individual model). After generating many trees, the resulting class prediction is based on votes from the single trees. Consequently, lower ranked variables are eliminated based on empirical performance heuristics (Han et al. 2006).

Two measures of variable importance are used in random forests: mean decrease accuracy and mean decrease Gini. These measures can lead to different results based on the size of dataset, heterogeneity of data points and dispersion of class values. Mean decrease accuracy is an internal estimation of the generalisation error generated by computing the out-of-bag error rate for each bootstrap sample (Le Cao and McLachlan 2009). Mean decrease Gini measures the impurity of a split, the Gini index, over all trees (Kuhn et al. 2008). However, as accuracy does not take into account the imbalanced nature of the given dataset, we decided to use the mean decrease Gini as the measure of variable importance for feature selection with random forest.

During initial experiments we discovered that random-forest feature selection considerably improves the effectiveness of the classifier when at least 50% of the features are removed from the original set; while no improvement is evident when 25% of the features only are removed. Thus, we decided to use random forest for feature selection by using the 20 features selected as the most important by random forest (see Figure 2). The number of trees in the random forest to be used for feature selection is defined by the lowest error rate in the initial random forest of 500 trees with 6 variables randomly sampled as candidates at each split (see Figure 4).

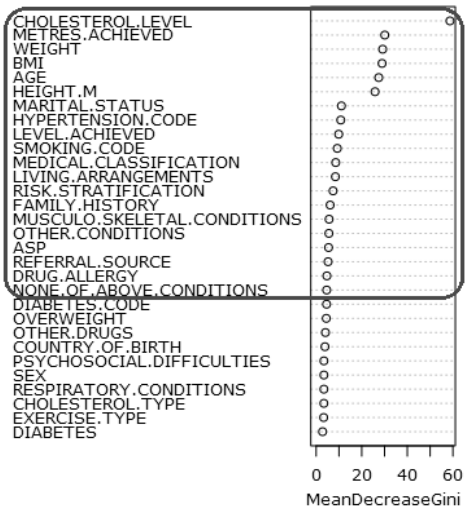


Figure 2: Sample variable importance output of random forest

Imbalanced data is another issue that can greatly affect effectiveness of the predictor because models built on data where examples of one class are greatly outnumbered by examples of the other classes tend to sacrifice accuracy for the underrepresented class in favour of maximizing the overall classification rate (Seiffert, Khoshgoftaar and Van Hulse 2009; Tuffery 2011). One common approach to dealing with imbalanced data is data sampling, which either adds examples to the minority class (oversampling) or removes examples from the majority class (undersampling) to create a more balanced data set. The primary criticism of undersampling is that information is lost when examples are removed from the training data, while oversampling increases dataset size by adding either no (in the case of random oversampling) or synthetic (in the case of more "intelligent" oversampling techniques) information (Seiffert, Khoshgoftaar and Van Hulse 2009). In the case of medical diagnosis data, it is crucial to retain data integrity. Thus, undersampling is more favourable in comparison to oversampling. Figure 3 illustrates how the cardiac dataset was partitioned for sampling purposes. Sizes of datasets used for classification of each separate target attribute depended on a number of missing values; however, sampling was always carried out according to Figure 3. Imbalanced testset of 30% was the largest possible set that left sufficient data for balanced training set.

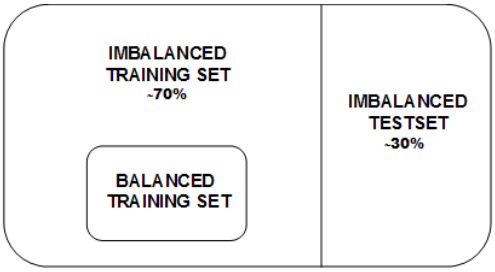


Figure 3: Dataset partitioning and sampling

As a result, it was decided to precede decision tree induction with random forest feature selection and undersampling to construct the optimum predictor. Two sequences of the proposed framework were developed with intention to compare their performance and select the optimal one. R software (version 2.12.2), including rattle and randomForest packages, was used for random forest feature selection. Weka software (version 3.6.4), J48 decision tree, was used for decision tree induction (Hall et al. 2009; Liaw and Wiener 2002; R Development Core Team 2011; Williams 2009). Table 4 outlines the proposed experimental design.

It was expected that models from group B will outperform models from group A due to the fact that features selected by models from group A are most likely to be biased towards over-represented class and, therefore, models based on these features will not perform well in terms of area under the Receiver Operating Characteristic (ROC) curve (AUC), precision and recall (see section 5.3). However, the result may differ due to a degree of imbalanced distribution of classes, noisy data, high variance and various other factors. Experimental results were used to illustrate final outcome.

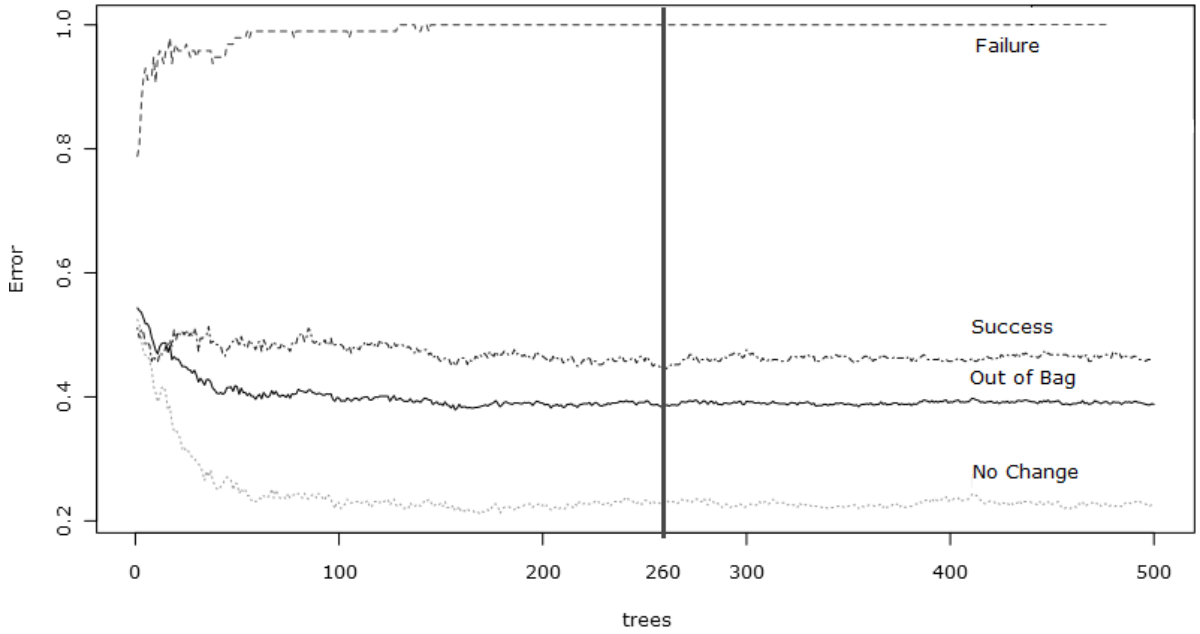


Figure 4: Sample error rate output of random forest

	Model A	Model B	Tool
1	Conduct Feature Selection by Random Forest on **imbalanced training set**	Conduct Feature Selection by Random Forest on **balanced training set**	R
2	Build Decision Tree using selected features on imbalanced training set (**Model A1**) and fine tune it using 10-fold cross validation	Build Decision Tree using selected features on imbalanced training set (**Model B1**) and fine tune it using 10-fold cross validation	Weka
3	Build Decision Tree using selected features on balanced training set (**Model A2**) and fine tune it using 10-fold cross validation	Build Decision Tree using selected features on balanced training set (**Model B2**) and fine tune it using 10-fold cross validation	Weka
4	Select the optimal model between Model A1 and Model A2 using testing against imbalanced test set	Select the optimal model between Model B1 and Model B2 using testing against imbalanced test set	Weka
5	Select the optimal model between Model A and Model B		Weka

Table 4: Classification Approach

5.3 Evaluation Criteria

In order to assess prediction models during fine-tuning and final evaluation, it is necessary to select appropriate evaluation measures. Measures commonly used for classifier performance evaluation include precision, sensitivity/recall, specificity, accuracy and AUC (Eisner 2011; Han and Kamber 2006). While accuracy of the model illustrates a portion of correctly classified instances, it is often not the most objective measure of classifier performance. There are other statistical measures, including sensitivity and specificity, which evaluate the performance of a binary classification test. Sensitivity (or recall) makes assessment in terms of the ratio of actual positives which are correctly labelled as positives, for instance, the percentage of people with coronary heart disease who are correctly identified as having such condition. Specificity, on the other hand, measures the ratio of negatives which are correctly labelled as negatives, for example, the percentage of healthy people who are correctly identified as not having the condition. Precision can be seen as a measure of exactness or quality, whereas recall is a measure of completeness or quantity (Goel and Singh 2010). There is usually a trade-off between sensitivity and specificity. For example, in medical diagnosis situation, the predictor may be biased towards low-risk symptoms (low specificity), in order to reduce the risk of misclassification of seriously ill patients as healthy (high sensitivity). This trade-off can be represented graphically as a ROC curve.

A ROC curve is another measure of accuracy which also shows a trade-off between the true positive rate and the false positive rate for a given model. To evaluate the accuracy of a model, it is necessary to calculate AUC. The AUC is an estimate of the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance. For this reason, the AUC is widely thought to be a better measure than a classification error rate (Rice 2010). AUC ranges from 0.5 to 1.0, where 1.0 represents a model with perfect accuracy (Han and Kamber 2006).

As outlined earlier, in the area of medical diagnosis it is crucial to consider the false negative and false positive rate. Thus, although AUC's position as the superior evaluation measure has recently been questioned, it is the preferred metric in the given scenario over the accuracy. The following is a rough guide used for classifying the predictor based on the AUC:

- **Excellent**: 0.9 - 1.0
- **Good**: 0.8 - 0.9
- **Fair**: 0.7 - 0.8
- **Poor**: 0.6 - 0.7
- **Fail**: 0.5 - 0.6

Moreover, precision and recall are also used to assess prediction of under-represented classes.

6 Results and Discussion

6.1 Experimental Results

Upon completion of the experiment outlined above classification models were compared using the AUC, precision and recall as described in section 5.3. Table 5 to 7 illustrate performance of resultant models within each group of risk factors: physiological, psychosocial and performance risk factors respectively. Note that the best models based on classification performance are shown in **bold**.

Evidently, models with extremely poor performance, including Cholesterol Level Change (20% threshold), BMI Change (0%, 10% and 20% threshold), Waist Circumference Change (0%, 10% and 20% threshold), Diabetes Change, Smoking Change and Metres Achieved Change, were built on exceptionally imbalanced datasets which indicates reasons for unfortunate outcomes .

It was found that although a number of techniques, such as feature selection by means of random forest and balanced sampling, resulted in an improvement in accuracy of classification models; selected models provide Fair or Poor performance only (see section 5.3). In an attempt to further explore datasets used to build selected models, a Principal Component Analysis (PCA) was conducted.

In order to conduct this analysis, categorical attributes of the dataset were re-coded into numeric values. Consequently, all attributes were adjusted to a mean of zero (by subtracting the mean from each value). Also, attributes with constant values were excluded from this analysis. The outcome of the PCA, a plot, remaps the data points from their original coordinates to coordinates of the first two principal coordinates. For the purposes of this study, PCA was carried out on the imbalanced training dataset. Notably, all PCA plots revealed that there are no clearly separable groups of data points which may indicate reasons for poor outcomes. Figure 5 illustrates one of the resultant PCA plots.

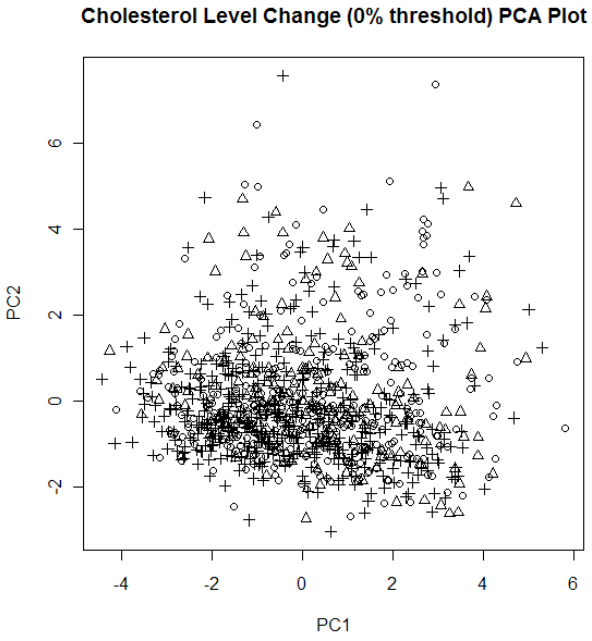


Figure 5: Cholesterol Level Change (0% threshold) PCA Plot (+ - S, △ - F, ○ - N)

Evidently, most models cannot be used for prediction, as they fail to predict under-represented classes. As a result, only the following models are recommended to be used and are described in detail: Cholesterol Level Change (0% threshold), Hypertension Change, HADS Anxiety Change, HADS Depression Change and Level Achieved Change. Table 8 to 12 show results of each individual experiment. Note that selected models are shown in **bold**.

In the Cholesterol Level Change (0% threshold) experiment model A2 and model B2 performed similarly in terms of all evaluation measures. Thus, it is recommended to use both models for prediction.

The Hypertension Change, Level Achieved Change and HADS Depression Change experiments resulted in model B2 being selected. Justifiably, it was expected that models from group B will outperform models from group A due to the fact that features selected by models from group A are most likely to be biased towards over-represented class.

Interestingly, in the HADS Anxiety Change experiment model A2 was selected. This can be explained by the fact that the HADS Anxiety Change dataset was least imbalanced, where under-represented class occupied over 17% of the whole dataset. Thus, random forest feature selection on balanced dataset did not incur a significant improvement in classification performance.

6.2 Medical Interpretation of Results

Based on the experimental results, we conclude that the outcome of phase II cardiac rehabilitation in terms of physiological, psychosocial and performance risk factor can be predicted based on initial readings of cholesterol level, hypertension, level achieved in six minute walk test, HADS anxiety score and HADS depression score with fair certainty.

Although selected models provide Fair classification performance only, it is recommended to use these models to identify high-risk groups of cardiac patients (i.e. Failure and/or No Change) based on performance risk factors and/or psychosocial risk factors of heart disease. Consequently, these patients can be provided with personalised cardiac rehabilitation program. This may include non-intrusive monitoring, advice on exercise training, counselling sessions, education on risk factors of heart disease, phone calls and regular checkups or nurse visits.

6.3 Variable Importance Results

Variable importance was another output of this study. Random forests used for feature selection in the experiment produce a variable importance ranking for each class label based on mean decrease accuracy as described in section 5.2. Note that only the output related to selected models is presented. Evidently, different attributes were defined as important for prediction of different class labels. The first most important attribute was always the attribute used for calculation of the outcome (e.g. CHOLESTEROL LEVEL for Cholesterol Level Change).

Notably, attributes defined as most important by random forest based on mean decrease accuracy are flagged as related to corresponding risk factors in numerous medical experiments:

Risk Factor	Model Parameters	Accuracy#	AUC#	Able to predict minor classes?^	Performance+
Cholesterol (0% Threshold)	**pruned, C = 0.25, M = 31**	**47.76%**	**0.652**	**Yes**	**Poor**
Cholesterol (0% Threshold)	**unpruned, M = 13**	**48.50%**	**0.650**	**Yes**	**Poor**
Cholesterol (10% Threshold)	unpruned, M = 13	47.46%	0.620	Yes	Poor
Cholesterol (20% Threshold)	unpruned, M = 15	68.06%	0.694	No	Poor
BMI (0% Threshold)	pruned, C = 0.25, M = 9	61.12%	0.595	No	Fail
BMI (10% Threshold)	unpruned, M = 12	93.93%	0.611	No	Poor
BMI (20% Threshold)	unpruned, M = 4	98.91%	0.629	No	Poor
Waist Circumference (0% Threshold)	unpruned, M = 26	52.51%	0.609	No	Poor
Waist Circumference (10% Threshold)	unpruned, M = 21	91.32%	0.486	No	Fail
Waist Circumference (20% Threshold)	N/A	N/A	N/A	N/A	N/A~
Hypertension	**pruned, C = 0.25,M = 13**	**49.41%**	**0.647**	**Yes**	**Poor**
Diabetes	unpruned, M = 4	98.26%	0.645	No	Poor
Smoking	unpruned, M = 3	90.18%	0.635	No	Poor

Table 5: Comparative Analysis of Prediction Models on Physiological Risk Factors of Heart Disease

Risk Factor	Model Parameters	Accuracy#	AUC#	Able to predict minor classes?^	Performance+
HADS Anxiety	**pruned, C = 0.25, M = 25**	**50.47%**	**0.691**	**Yes**	**Poor**
HADS Depression	**pruned, C = 0.25,M = 14**	**49.26%**	**0.683**	**Yes**	**Poor**

Table 6: Comparative Analysis of Prediction Models on Psychosocial Risk Factors of Heart Disease

Risk Factor	Model Parameters	Accuracy#	AUC#	Able to predict minor classes?^	Performance+
Level Achieved	**pruned, C = 0.25, M = 10**	**52.92%**	**0.739**	**Yes**	**Fair**
Metres Achieved (0% Threshold)	unpruned, M = 30	85.60%	0.653	No	Poor
Metres Achieved (10% Threshold)	unpruned, M = 15	66.28%	0.632	No	Poor
Metres Achieved (20% Threshold)	pruned, C = 0.25, M = 23	65.30%	0.705	No	Fair

Table 7: Comparative Analysis of Prediction Models on Performance Risk Factors of Heart Disease

Model	Model Parameters	AUC#	Accuracy#	Precision#	Recall#	Able to predict minor classes?^
A1	unpruned, M = 37	0.674	53.73%	0.516	0.537	No
A2	**unpruned, M = 17**	**0.652**	**47.76%**	**0.529**	**0.478**	**Yes**
B1	unpruned, M = 36	0.677	53.85%	0.513	0.536	No
B2	**unpruned, M = 13**	**0.650**	**48.50%**	**0.520**	**0.485**	**Yes**

Table 8: Cholesterol Level Change (0% threshold) Classification Performance

Model	Model Parameters	AUC#	Accuracy#	Precision#	Recall#	Able to predict minor classes?^
A1	unpruned, M = 15	0.669	84.94%	0.839	0.849	No
A2	unpruned, M = 15	0.620	31.57%	0.791	0.316	No
B1	unpruned, M = 15	0.657	84.54%	0.774	0.845	No
B2	**pruned, C = 0.25, M = 13**	**0.647**	**49.41%**	**0.796**	**0.494**	**Yes**

Table 9: Hypertension Change Classification Performance

Model	Model Parameters	AUC#	Accuracy#	Precision#	Recall#	Able to predict minor classes?^
A1	pruned, C = 0.25, M = 7	0.815	76.94%	0.734	0.769	No
A2	pruned, C = 0.25, M = 7	0.723	55.61%	0.674	0.556	Yes
B1	pruned, C = 0.25, M = 7	0.789	75.36%	0.719	0.754	No
B2	**pruned C = 0.25, M = 10**	**0.739**	**52.92%**	**0.668**	**0.529**	**Yes**

Table 10: Level Achieved Change Classification Performance

Model	Model Parameters	AUC#	Accuracy#	Precision#	Recall#	Able to predict minor classes?^
A1	pruned, C = 0.25, M = 19	0.690	57.05%	0.576	0.571	No
A2	**pruned, C = 0.25, M = 25**	**0.691**	**50.47%**	**0.552**	**0.505**	**Yes**
B1	pruned, C = 0.25, M = 15	0.690	56.74%	0.559	0.567	No
B2	pruned, C = 0.25, M = 26	0.673	52.66%	0.569	0.527	Yes

Table 11: HADS Anxiety Change Classification Performance

Model	Model Parameters	AUC#	Accuracy#	Precision#	Recall#	Able to predict minor classes?^
A1	pruned, C = 0.25, M = 13	0.698	59.75%	0.547	0.597	No
A2	pruned, C = 0.25, M = 13	0.676	46.86%	0.521	0.469	Yes
B1	pruned, C = 0.25, M = 19	0.672	62.58%	0.563	0.626	No
B2	pruned, C = 0.25, M = 14	**0.683**	**49.26%**	**0.537**	**0.491**	**Yes**

Table 12: HADS Depression Change Classification Performance

+ Based on proposed scale (see section 5.3).
^ Model is determined as being able to predict under-represented classes when recall and precision on under-represented classes derived from testing on unseen data >=0.4 and >=0.2 respectively.
~ In Waist Circumference Change (20% threshold) almost all data points belong to N class and, therefore, it was of no value to conduct any further analysis of this dataset.
#Overall statistics across all class labels calculated as weighted average based on testing on unseen data (see section 5.2).
M is the minimum number of instances per leaf; C is the confidence factor used for pruning.

- In Cholesterol Level Change (0% threshold), No Change is partially determined by the statin medication (ST) attribute. In reality, statins are known for adverse side effects in terms of muscle pain and damage (Tomlinson and Mangione 2005).
- In the Hypertension Change experiment, the angiotensin-converting enzyme antihypertensive medication (ACE) and BMI appeared to be important for the Success and No Change groups respectively.
- In case of Level Achieved Change, WEIGHT and AGE were determined important for prediction of Success and No Change in Level Achieved Change.
- RESPIRATORY and MUSCULO-SKELETAL CONDITIONS were important for prediction of Failure in Level Achieved Change. Moreover, the HYPERTENSION CODE (showing whether a patient has a hypertension condition and is treated by medications) was also found to be important in prediction of the Failure points. In fact, beta-blockers, one of the antihypertensive medications, can lead to a clear reduction in exercise capacity (Bangalore and Messerli 2006; Chang et al. 2010).
- The antianginal medication (ANG) was found important for prediction of Failure in HADS Anxiety Change. Plausibly, angina pectoris, severe chest pain due to a lack of oxygen supply to the heart muscle, is a significant determinant of patient anxiety (Lewin et al. 2002).
- BP DIASTOLIC and BP SYSTOLIC were interestingly found significant for prediction of No Change and Failure in HADS Depression Change respectively. A study of older men found that men with low diastolic blood pressure had significantly higher depression scores. Moreover, 'depression was more strongly associated with low diastolic than low systolic blood pressure, but low systolic pressure was present in only 22 men who did not also have low diastolic blood pressure' (Barrett-Connor and Palinkas 1993).

The above discussion shows that results of our analysis of variable importance for cardiac risk factors have strong correlations with medical observations.

7 Conclusions and Future Work

Recent research suggests that cardiovascular diseases remain the leading cause of premature death and disability in the majority of industrialised countries (Piperidou & Bliss 2008). In this study we used decision tree induction with random forest feature selection and undersampling for prediction of cardiovascular risk factors. We found that models built on balanced datasets using features selected on balanced datasets generally performed better than other models. Although applied techniques led to an improved classification performance, resultant models provided Poor to Fair performance. Moreover, PCA confirmed that there are no clearly separable groups of data points. This can be explained by a rather diverse population used in this study, a large number of missing values and noisy data. However, despite the low quality of prediction models, variable importance results are particularly accurate in terms of correlation with medical theory and practice.

It is recommended to explore alternative strategies for building data mining models, such as:

- Performing clustering on the dataset followed by classification on each cluster, as it could be likely that under-represented classes occur within the same cluster of patients;
- Applying random forest feature selection using mean decrease accuracy;
- Deploying alternative classification algorithms (such as support vector machines and association rule mining);
- Collecting more data and researching into alternative methods for imbalanced data problem, in an attempt to improve classification performance.

Moreover, since variable importance results were proved to be rather significant, it is recommended to attempt multivariate analysis in combination with random forest feature selection for prediction of the outcome of cardiac rehabilitation in future studies.

This research takes cardiac rehabilitation research to another level where patients are treated based on their risk factor profile. The results suggest a personalised approach to developing the exercise training program as opposed to generalised approach. It also provides maximum benefits to cardiac patients such as improved health-related quality of life, reduced number of hospital readmissions and

decreased rate of mortality and morbidity. As a result, this research will help to reduce the economic burden on the health system caused by cardiovascular disease.

Acknowledgements

This work was supported by the Department of Cardiology of the Royal North Shore Hospital and by the Faculty of Engineering and Information Technology of the University of Technology, Sydney.

References

Almerud Osterberg, S., Baigi, A., Bering, C. and Fridlund, B. (2010): Knowledge of heart disease risk in patients declining rehabilitation. *British Journal of Nursing* **19**(5):288-293.

Austin, J., Williams, R., Ross, L., Moseley, L. and Hutchison, S. (2005): Randomised controlled trial of cardiac rehabilitation in elderly patients with heart failure. *European Journal of Heart Failure* **7**(3):411-417.

Austin, J., Williams, W.R., Ross, L. and Hutchison, S. (2008): Five-year follow-up findings from a randomized controlled trial of cardiac rehabilitation for heart failure. *European Journal of Cardiovascular Prevention and Rehabilitation* **15**(2):162-167.

Bangalore, S. and Messerli, F.H (2006): Beta-blockers and exercise. *Journal of the American College of Cardiology* **48**(6):1283-1288.

Barrett-Connor, E. and Palinkas, L.A. (1994): Low blood pressure and depression in older men: a population based study. *British Medical Journal* **308**(6926):446-449.

Brubaker, P.H., Moore, J.B., Stewart, K.P., Wesley, D.J. and Kitzman, D.W. (2009): Endurance exercise training in older patients with heart failure: results from a randomized, controlled, single-blind trial. *Journal of the American Geriatrics Society* **57**(11):1982-1989.

Chang, C.D., Wang, C.C. and Jiang, B.C. (2011): Using data mining techniques for multi-diseases prediction modeling of hypertension and hyperlipidemia by common risk factors. *Expert Systems with Applications* **38**(1):5507-5513.

Chang, C.L., Mills, G.D., McLachlan, J.D., Karalus, N.C. and Hancox, R.J. (2010): Cardio-selective and non-selective beta-blockers in chronic obstructive pulmonary disease: effects on bronchodilator response and exercise. *Internal Medicine Journal* **40**(3):193-200.

Chester, T. (2006): Cardiac rehabilitation for patients with heart failure: a service development unit. *British Journal of Cardiac Nursing* **1**(10):487-495.

Davies, E.J., Moxham, T., Rees, K., Singh, S., Coats, A.J., Ebrahim, S., Lough, F. and Taylor, R.S. (2010): Exercise based rehabilitation for heart failure. *The Cochrane Library* **4**(1):1-57.

Delagardelle, C. and Feiereisen, P. (2005): Strength training for patients with chronic heart failure, *Europa Medicophysica* **41**(1):57-65.

Eisner, R.: Basic evaluation measures for classifier performance, University of Alberta. http://webdocs.cs.ualberta.ca/~eisner/measures.html. Accessed 1 June 2011.

Goble, A.J. and Worcester, M.U.C.:Best Practice Guidelines for Cardiac Rehabilitation and Secondary Prevention, Department of Human Services Victoria. http://www.health.vic.gov.au/nhpa/downloads/bestpracticecardiacrehab.pdf. Accessed 9 May 2010.

Guiraud, T., Juneau, M., Nigam, A., Gayda, M., Meyer, P., Mekary, S., Paillard, F. and Bosquet, L. (2010): Optimization of high intensity interval exercise in coronary heart disease. *European Journal of Applied Physiology* **108**(4):733-740.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I.H. (2009): The weka data mining software: an update. *SIGKDD Explorations* **11**(1).

Han, J. and Kamber, M. (2006): *Data mining: concepts and techniques*. San Francisco, Morgan Kaufmann.

Han, L., Embrechts, M.J., Szymanski, B., Sternickel, K. and Ross, A. (2006): Random forests feature selection with K-PLS: detecting ischemia from magnetocardiograms. *Proc. European Symposium on Artificial Neural Networks*, Bruges, Belgium, **14**:221-226, ESANN.

Hansen, D., Dendale, P., Raskin, A., Schoonis, A., Berger, J., Vlassak, I. and Meeusen, R. (2010): Long-term effect of rehabilitation in coronary artery disease patients: randomized clinical trial of the impact of exercise volume. *Clinical Rehabilitation*. **24**(4):319-327.

Kajabadi, A., Saraee, M.H. and Asgari, S. (2009): Data mining cardiovascular risk factors. *Proc. AICT International Conference on Application of Information and Communication Technologies*, Baku, Azerbaijan, **3**:1-5, AICT.

Karaolis, M., Moutiris, J.A., Hadjipanayi, D. and Pattichis, C.S. (2010): Assessment of the risk factors of coronary heart events based on data mining with decision trees. *IEEE Transactions on Information Technology in Biomedicine* **14**(3):559-566.

Kravari, M., Vasileiadis, I., Gerovasili, V., Karatzanos, E., Tasoulis, A., Kalligras, K., Drakos, S., Dimopoulos, S., Anastasiou-Nana, M. and Nanas, S. (2010): Effects of a 3-month rehabilitation program on muscle oxygenation in congestive heart failure patients as assessed by NIRS. *International Journal of Industrial Ergonomics* **40**(2):212-217.

Kuhn., S., Egert, B., Neumann, S. and Steinbeck, C. (2008): Building blocks for automated elucidation of metabolites: machine learning methods for NMR prediction. *BMC Informatics* **9**(1):400-418.

Kulcu, D.G., Kurtais, Y., Tur, B.S., Gulec, S. and Seckin, B. (2007): The effect of cardiac rehabilitation on quality of life, anxiety and depression in patients with congestive heart failure. A randomized controlled trial, short-term results. *Europa Medicophysica* **43**(4):489-497.

Lavie, C.J., Milani, R.V. and Ventura, H.O. (2009): Exercise training and heart failure in older adults-dismal failure or not enough exercise? *Journal of the American Geriatrics Society* **57**(11):2148-2150.

Le Cao, K.A. and McLachlan, G.J. (2009): Statistical analysis of microarray data: selection of gene prognosis signatures. In *Computational biology: issues and applications in oncology*. 55-76. T. Pham (ed). Springer-Verlag.

Lewin, R.J.P., Furze, G., Robinson, J., Griffith, K., Wiseman, S., Pye, M. and Boyle, R. (2002): A randomised controlled trial of a self-management plan for patients with newly diagnosed angina. *British Journal of General Practice* **52**(476):194-201.

Liaw, A. and Wiener, M. (2002): Classification and regression by randomForest. *R News* **2**(3):18-22.

Linoff, G.S. and Berry, M.J. (2011): *Data mining techniques: for marketing, sales, and customer relationship management*. Indianapolis, Wiley Publishing.

Marques Marcolino, J.A., da Silva Telles Mathias, L.A., Piccinini Filho, L., Guaratini, A.A., Mikio Suzuki, F. & Cunha Alli, L.A. (2007): Hospital anxiety and depression scale: a study on the validation of the criteria and reliability on preoperative patients. *Revista Brasileira de Anestesiologia* **57**(1): 52-62.

Menze, B.H., Kelm, M., Masuch, R., Himmelreich, U., Bachert, P., Petrich, W. and Hamprecht, F.A. (2009): A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data. *BMC Bioinformatics* **10**(1):213-228.

Nilsson, B.B., Hellesnes, B., Westheim, A. and Risberg, M.A. (2008): Group-based aerobic interval training in patients with chronic heart failure: Norwegian Ullevaal Model. *Physical Therapy* **88**(4):523-535.

Noy, K. (1998): Cardiac rehabilitation: structure, effectiveness and the future. *British Journal of Nursing* **7**(17):1033-1040.

Pan, X.Y. and Shen, H.B. (2009): Robust prediction of B-factor profile from sequence using two-stage SVR based on random forest feature selection. *Protein and peptide letters* **16**(1):1447-1454.

Piperidou, E. and Bliss, J. (2008): An exploration of exercise training effects in coronary heart disease. *British Journal of Community Nursing* **13**(6):271-277.

Podgorelec, V., Kokol, P., Stiglic, B. and Rozman, I. (2002): Decision trees: an overview and their use in medicine. *Journal of medical systems* **26**(5):445-463.

R Development Core Team (2011): *A language and environment for statistical computing*. Vienna, R Foundation for Statistical Computing.

Rivett, M.J., Tsakirides, C., Pringle, A., Carroll, S., Ingle, L. and Dudfield, M. (2009): Physical activity readiness in patient withdrawals from cardiac rehabilitation. *British Journal of Nursing* **18**(3):188-191.

Seiffert, C., Khoshgoftaar, T.M. and Van Hulse, J. (2009): Hybrid sampling for imbalanced data. *Integrated Computer-Aided Engineering* **16**(3):193-210.

Seni, G. and Elder, J. (2010): *Ensemble methods in data mining: improving accuracy through combining predictions*. Chicago, Morgan and Claypool Publishers.

Smith, J.G., Newton-Cheh, C., Almgren, P., Struck, J., Morgenthaler, N.G., Bergman, A., Platonov, P.G., Hedblad, B., Engstrom, G., Wang, T.J. and Melander, O. (2010): Assessment of conventional cardiovascular risk factors and multiple biomarkers for the prediction of incident heart failure and atrial fibrillation. *Journal of the American College of Cardiology* **56**(21):1712-1719.

Tabet, J.Y., Meurin, P., Beauvais, F., Weber, H., Renaud, N., Thabut, G., Cohen-Solal, A., Logeart, D. and Ben Driss, A. (2008): Absence of exercise capacity improvement after exercise training program: a strong prognostic factor in patients with chronic heart failure, *Circulation Heart Failure* **1**(4):220-226.

Tuffery, S. (2011): *Data mining and statistics for decision making*. Chichester, John Wiley and Sons.

Williams, G.J (2009): Rattle: a data mining GUI for R. *The R Journal* **1**(2):45-55.

Using Decision Tree for Diagnosing Heart Disease Patients

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

Abstract

Heart disease is the leading cause of death in the world over the past 10 years. Researchers have been using several data mining techniques to help health care professionals in the diagnosis of heart disease. Decision Tree is one of the successful data mining techniques used. However, most research has applied J4.8 Decision Tree, based on Gain Ratio and binary discretization. Gini Index and Information Gain are two other successful types of Decision Trees that are less used in the diagnosis of heart disease. Also other discretization techniques, voting method, and reduced error pruning are known to produce more accurate Decision Trees. This research investigates applying a range of techniques to different types of Decision Trees seeking better performance in heart disease diagnosis. A widely used benchmark data set is used in this research. To evaluate the performance of the alternative Decision Trees the sensitivity, specificity, and accuracy are calculated. The research proposes a model that outperforms J4.8 Decision Tree and Bagging algorithm in the diagnosis of heart disease patients.

Keywords: Data Mining, Decision Tree, Discretization, Heart Disease.

1 Introduction

Heart disease is the leading cause of death in the world over the past 10 years (World Health Organization 2007). The European Public Health Alliance reported that heart attacks, strokes and other circulatory diseases account for 41% of all deaths (European Public Health Alliance 2010). The Economical and Social Commission of Asia and the Pacific reported that in one fifth of Asian countries, most lives are lost to non-communicable diseases such as cardiovascular diseases, cancers, diabetes and chronic respiratory diseases (ESCAP 2010). The Australian Bureau of Statistics reported that heart and circulatory system diseases are the first leading cause of death in Australia, causing 33.7% all deaths (Australian Bureau of Statistics 2010).

Motivated by the world-wide increasing mortality of heart disease patients each year and the availability of huge amount of patients' data from which to extract useful knowledge, researchers have been using data mining techniques to help health care professionals in the diagnosis of heart disease (Helma, Gottmann et al. 2000; Podgorelec, Kokol et al. 2002). Data mining is the exploration of large datasets to extract hidden and previously unknown patterns, relationships and knowledge that are difficult to detect with traditional statistical methods (Lee, Liao et al. 2000; Thuraisingham 2000; Obenshain 2004; Han and Kamber 2006; Sandhya, P. Deepa Shenoy et al. 2010).

Data mining in healthcare is an emerging field of high importance for providing prognosis and a deeper understanding of medical data. Data mining applications in healthcare include analysis of health care centres for better health policy-making and prevention of hospital errors, early detection, prevention of diseases and preventable hospital deaths, more value for money and cost savings, and detection of fraudulent insurance claims (Ruben 2009). Researchers are using data mining techniques in the medical diagnosis of several diseases such as diabetes (Porter and Green 2009), stroke (Panzarasa, Quaglini et al. 2010), cancer (Li L 2004), and heart disease (Das, Turkoglu et al. 2009). Several data mining techniques are used in the diagnosis of heart disease such as Naïve Bayes, Decision Tree, neural network, kernel density, automatically defined groups, bagging algorithm, and support vector machine showing different levels of accuracies (Yan, Zheng et al. 2003; Andreeva 2006; Das, Turkoglu et al. 2009; Sitar-Taut, Zdrenghea et al. 2009; Rajkumar and Reena 2010; Srinivas, Rani et al. 2010)

This paper presents a new model that enhances the Decision Tree accuracy in identifying heart disease patients. The model integrates a multiple classifiers voting technique with different types of discretization methods and different types of Decision Trees. The rest of the paper is divided as follows: the background section investigates applying data mining techniques in the diagnosis of heart disease, the methodology section explains the proposed methodology for enhancing the Decision Tree accuracy in diagnosing heart disease, and the results section is followed by a summary section.

2 Background

Knowledge of the risk factors associated with heart disease helps health care professionals to identify patients at high risk of having heart disease. Statistical analysis has identified the risk factors associated with heart disease to be age, blood pressure, smoking habit (Heller, Chinn et al. 1984), total cholesterol (Wilson, D'Agostino et al. 1998), diabetes (Simons, Simons et al. 2003),

hypertension, family history of heart disease (Salahuddin and Rabbi 2006), obesity, and lack of physical activity (Shahwan-Akl 2010).

Researchers have been applying different data mining techniques to help health care professionals with improved accuracy in the diagnosis of heart disease. Neural network, Naïve Bayes, Genetic algorithm, Decision Tree, classification via clustering, and direct kernel self-organizing map are some techniques used in the diagnosis of heart disease (De Beule, Maesa et al. 2007; Tantimongcolwat, Naenna et al. 2008; Das, Turkoglu et al. 2009; Anbarasi, Anupriya et al. 2010; Kavitha, Ramakrishnan et al. 2010; Srinivas, Rani et al. 2010).

In particular, researchers have been investigating the application of the Decision Tree technique in the diagnosis of heart disease with considerable success. Andreeva used C4.5 Decision Tree in the diagnosis of heart disease showing accuracy of 75.73% (Andreeva 2006). Sitair-Taut et al. used the weka tool to investigate applying Naïve Bayes and J4.8 Decision Trees for the detection of coronary heart disease. The results showed that there is no significant difference between Naïve Bayes and Decision Trees in the ability to realize a correct prediction of coronary heart disease (Sitar-Taut, Zdrenghea et al. 2009). Tu et al. used the bagging algorithm in the weka tool and compared it with J4.8 Decision Tree in the diagnosis of heart disease. The bagging algorithm showed the better accuracy of 81.41% while the Decision Tree showed an accuracy of 78.91% (Tu, Shin et al. 2009).

Applying Decision Tree techniques has shown useful accuracy in the diagnosis of heart disease. But assisting health care professionals in the diagnosis of the world's biggest killer demands higher accuracy. Our research seeks to improve diagnosis accuracy to improve health outcomes. Most Decision Tree types used such as J4.8 and C4.5 Decision Trees are based on Gain Ratio in the extraction of Decision Tree rules. However there are other Decision Tree types such as Information Gain and Gini Index that have been less used in the diagnosis of heart disease.

Decision Tree is one of the data mining techniques that cannot handle continuous variables directly so the continuous attributes must be converted to discrete attributes, a process called discretization (Kotsiantis and Kanellopoulos 2006). J4.8 and C4.5 Decision Trees use binary discretization for continuous-valued features. However, multi-interval discretization methods are known to produce more accurate Decision Trees than binary discretization (Fayyad and Keki 1992; Perner and Trautzsch 1998), yet are less used in heart disease diagnosis research. Other important accuracy improving is applying multiple classifier voting and reduced error pruning to Decision Tree in the diagnosis of heart disease patients. Intuitively, more complex models might be expected to produce more accurate results, but which combination of techniques is best?

Seeking to thoroughly investigate options for accuracy improvements in heart disease diagnosis this paper systematically investigates applying multiple classifiers voting technique with different multi-interval discretization methods such as equal width, equal frequency, chi merge and entropy with different types of Decision Tree such as Information Gain, Gini Index, and Gain Ratio.

3 Methodology

There are two main issues that affect the performance of Decision Trees; the data discretization method used and the type of Decision Tree used. Reduced error pruning is shown to further improve decision tree performance. The proposed methodology involves systematically testing different discretization techniques, multiple classifiers voting technique and different Decision Trees type in the diagnosis of heart disease patients. Different combinations of discretization methods, decision tree types and voting are tested to identify which combination will provide the best performance in diagnosing heart disease patients. A test harness was implemented using Microsoft Visual Studio 2008.

3.1 Data Discretization

Discretization methods are categorised as supervised or unsupervised (Dougherty, Kohavi et al. 1995). The unsupervised discretization methods do not make use of class membership information during the discretization process. The supervised discretization methods use the class labels for carrying out discretization process such as chi-square based methods and entropy based methods (Kotsiantis and Kanellopoulos 2006). All the discretization methods are used as a pre-processing step to convert the continuous attributes in the data set to discrete attributes. The number of intervals used by the discretization techniques is five. Each method was used to pre-process the benchmark data set for trials of each decision tree type (hence left-most column of Tables 2 and 3).

3.1.1 Unsupervised Discretization

In unsupervised discretization, equal-width interval and equal-frequency methods are used. The equal-width discretization algorithm determines the minimum and maximum values of the discretized attribute and then divides the range into the user-defined number of equal-width discrete intervals. The equal-frequency algorithm determines the minimum and maximum values of the discretized attribute, sorts all values in ascending order, and divides the range into a user-defined number of intervals so that every interval contains the same number of sorted values (Dougherty, Kohavi et al. 1995).

3.1.2 Supervised Discretization

In supervised discretization chi merge and entropy are two of the most well-known discretization methods (Bramer 2007). The chi merge discretization uses $\chi 2$ statistic to determine the independence of the class from the two adjacent intervals, combining them if they are dependent, and allowing them to be separate otherwise (Kerber 1992). This algorithm merges the pair of intervals with the lowest value of $\chi 2$ as long as the number of intervals is more than predefined maximum number of intervals.

The entropy discretization used in this model is that developed by Fayyad and Keki (Fayyad and Keki 1992).

Entropy is an information-theoretic measure of the 'uncertainty' contained in a training set (Han and Kamber 2006). It evaluates candidate cut points through an entropy-based method to select boundaries for discretization. Instances are sorted into ascending numerical order and then the entropy for each candidate cut point is calculated. Cut points are recursively selected to minimize entropy until a stopping criterion is achieved. In this model the stop criterion is achieving five intervals of the attribute.

3.2 Voting

Multiple classifier voting involves dividing the training data into smaller equal subsets of data and building a Decision Tree classifier for each subset of data. Voting is based on plurality or majority voting; each individual classifier contributes a single vote (Hall, Bowyer et al. 2000). Applying voting to classification algorithms is showing successful improvement in the accuracy of these classifiers (Paris, Affendey et al. 2010). The research here tested voting subsets where the data was divided between three and eleven subsets for each discretization method for each decision tree type. The most successful division (nine subsets) is reported here.

3.3 Decision Tree Type

There are many types of Decision Trees. The difference between them is the mathematical model that is used in selecting the splitting attribute in extracting the Decision Tree rules. The research tests the three most commonly-used types: Information Gain, Gini Index, and Gain Ratio Decision Trees, each described below.

3.3.1 Information Gain

The entropy (Information Gain) approach selects the splitting attribute that minimizes the value of entropy, thus maximising the Information Gain. To identify the splitting attribute of the Decision Tree, one must calculate the Information Gain for each attribute and then select the attribute that maximizes the Information Gain. The Information Gain for each attribute is calculated using the following formula (Han and Kamber 2006; Bramer 2007):

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

Where k is the number of classes of the target attribute

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

3.3.2 Gini Index

The Gini Index measures the impurity of data. The Gini Index is calculated for each attribute in the data set. If there are k classes of the target attribute, with the probability of the ith class being Pi, the Gini Index is defined as (Bramer 2007):

$$\text{Gini Index} = 1 - \sum_{i=1}^{k} Pi^2 \quad (2)$$

The splitting attribute is the attribute with the largest reduction in the value of the Gini Index.

3.3.3 Gain Ratio

To reduce the effect of the bias resulting from the use of Information Gain, a variant known as Gain Ratio was introduced by the Australian academic Ross Quinlan (Bramer 2007). The Information Gain measure is biased toward tests with many outcomes. That is, it prefers to select attributes having a large number of values (Han and Kamber 2006). Gain Ratio adjusts the Information Gain for each attribute to allow for the breadth and uniformity of the attribute values.

Gain Ratio = Information Gain / Split Information (3)

Where the split information is a value based on the column sums of the frequency table (Bramer 2007).

3.4 Pruning

After extracting the decision tree rules, reduced error pruning was used to prune the extracted decision rules. Reduced error pruning is one of the fastest pruning methods and known to produce both accurate and small decision rules (Esposito, Malerba et al. 1997). Applying reduced error pruning provides more compact decision rules and reduces the number of extracted rules.

3.5 Performance Evaluation

To evaluate the performance of each combination the sensitivity, specificity, and accuracy were calculated. The sensitivity is proportion of positive instances that are correctly classified as positive (i.e. the proportion of sick people that are classified as sick). The specificity is the proportion of negative instances that are correctly classified as negative (i.e. the proportion of healthy people that are classified as healthy). The accuracy is the proportion of instances that are correctly classified (Bramer 2007). To measure the stability of the performance of the proposed model the data is divided into training and testing data with 10-fold cross validation.

Sensitivity = True Positive/Positive (4)

Specificity = True Negative/ Negative (5)

Accuracy = (True Positive + True Negative) / (Positive + Negative) (6)

3.6 Summary

The research process involves data discretization, data partitioning, Decision Tree type selection, and the application of reduced error pruning to produce pruned Decision Trees. The data discretization is divided into supervised and unsupervised methods. The unsupervised methods involve equal width and equal frequency while the supervised discretization methods involve chi merge and entropy. The data partitioning involves testing with and without voting. Three Decision Tree types are tested: Information Gain, Gini Index, and Gain Ratio. Finally, reduced error pruning is applied on all the Decision Tree rules extracted from the training data. Figure 1 summarizes the components of the research process.

The actual testing involved executing each variant of each element in combination against the whole data set. Twelve Decision Tree variants were created by mixing discretization approaches with different Decision Tree

types. Each variant was then tested on its own and through different voting partitioning schemes (three, five, seven, nine and eleven partitions). The result of each variant through each voting partition had reduced error pruning applied. Overall, more than 70 Decision Trees were executed over the one data set to compile the findings presented here

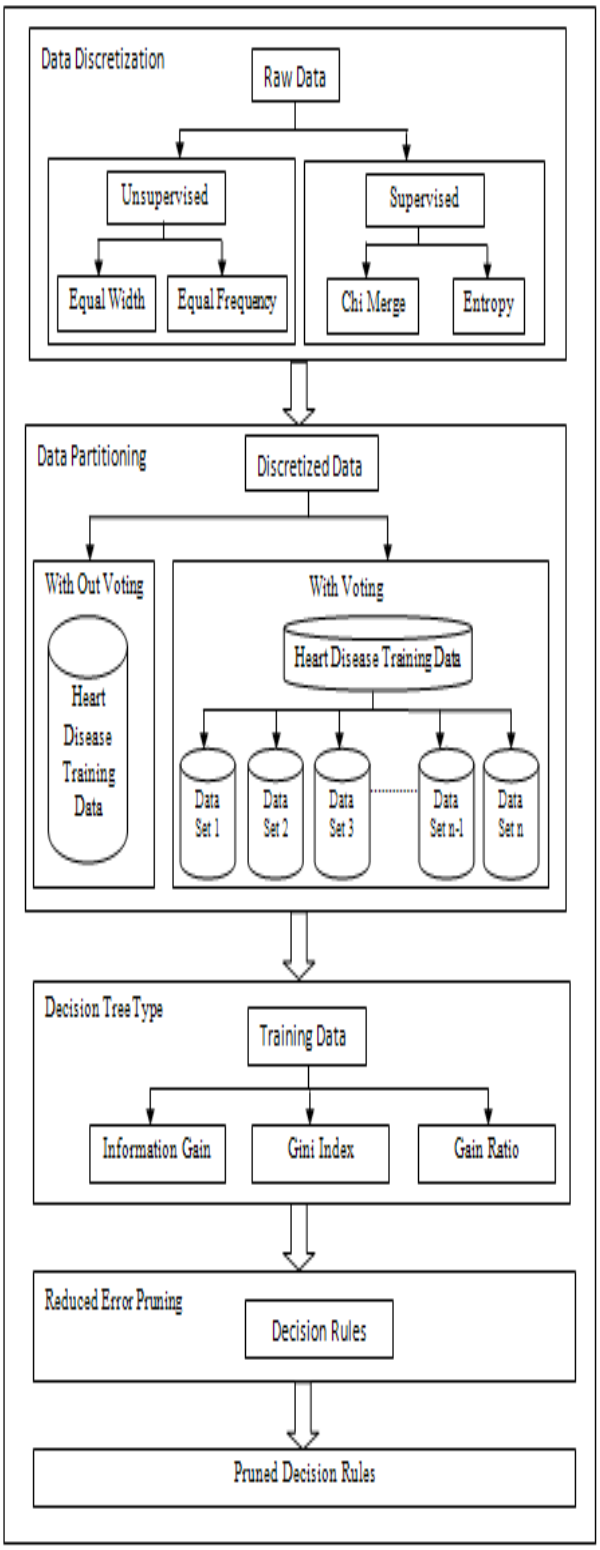


Figure 1: Research Process Used to Assess Alternative Decision Tree Techniques

4 Data

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

Table 1: Selected Cleveland Heart Disease Data Set Attributes

Name	Type	Description
Age	Continuous	Age in years
Sex	Discrete	1 = male 0 = female
Cp	Discrete	Chest pain type: 1 = typical angina 2 = atypical angina 3 = non-anginal pain 4 =asymptomatic
Trestbps	Continuous	Resting blood pressure (in mm Hg)
Chol	Continuous	Serum cholesterol in mg/dl
Fbs	Discrete	Fasting blood sugar > 120 mg/dl: 1 = true 0 = false
Restecg	Discrete	Resting electrocardiographic results: 0 = normal 1 = having ST-T wave abnormality 2 =showing probable or define left ventricular hypertrophy by Estes'criteria
Thalach	Continuous	Maximum heart rate achieved
Exang	Discrete	Exercise induced angina: 1 = yes 0 = no
Old peak ST	Continuous	Depression induced by exercise relative to rest
Slope	Discrete	The slope of the peak exercise segment : 1 = up sloping 2 = flat 3= down sloping
Ca	Discrete	Number of major vessels colored by fluoroscopy that ranged between 0 and 3.
Thal	Discrete	3 = normal 6= fixed defect 7= reversible defect
Diagnosis	Discrete	Diagnosis classes: 0 = healthy 1= patient who is subject to possible heart disease

5 Results

The results of sensitivity, specificity, and accuracy in the diagnosis of heart disease using equal width, equal frequency, chi merge, and entropy discretization with Information Gain, Gini Index, and Gain Ratio Decision Trees and reduced error pruning are shown in Table 2 and Table 3. Table 2 shows the results without applying voting to the Decision Tree. The highest accuracy achieved is 79.1% by the equal width discretization Information Gain Decision Tree. Different partitions of voting were applied to the data. The nine subsets voting showed the best performance and is the only iteration reported here (Table 3). The highest accuracy achieved is 84.1% by the equal frequency discretization Gain Ratio Decision Tree. Table 4 shows the difference in accuracy when applying the nine subsets voting scheme. The highest increase in the accuracy is achieved by the equal frequency discretization Gain Ratio Decision Tree and is 6.4%.

Table 2: Without Voting Decision Tree Results

		Sensitivity	Specificity	Accuracy
Equal Width	Info Gain	78.1%	79.4%	79.1%
	Gini Index	76.4%	83.4%	78.8%
	Gain Ratio	66.1%	80.5%	75.5%
Equal Frequency	Info Gain	76%	75.7%	76.3%
	Gini Index	75.5%	77.7%	76.9%
	Gain Ratio	71.4%	79.9%	77.7%
ChiMerge	Info Gain	71.7%	80.5%	77.1%
	Gini Index	73%	81.1%	78.2%
	Gain Ratio	63.3%	81.4%	75.1%
Entropy	Info Gain	78.1%	79.7%	78.1%
	Gini Index	77.1%	80.7%	78.4%
	Gain Ratio	68%	82.4%	76.5%

The chimerge and entropy supervised discretization methods do not show any enhancement in the Decision Tree accuracy either with or without voting. Applying the voting is showing an increase in the accuracy of different types of Decision Tree. When comparing the best results with the J4.8 Decision Tree and bagging algorithm that used the same data set (Tu, Shin et al. 2009), this research achieved higher sensitivity, specificity, and accuracy than J4.8 Decision Tree and achieved higher sensitivity, and accuracy than bagging algorithm as shown in Table 5.

Table 3: Nine Voting Decision Tree Results

		Sensitivity	Specificity	Accuracy
Equal Width	Info Gain	73%	89.7%	82.6%
	Gini Index	69%	89.6%	81.5%
	Gain Ratio	70.3%	90.6%	81%
Equal Frequency	Info Gain	69.3%	86.3%	82%
	Gini Index	68.4%	88.1%	81.4%
	Gain Ratio	77.9%	85.2%	84.1%
ChiMerge	Info Gain	71.6%	83.2%	78.3%
	Gini Index	72.7%	82.7%	79.4%
	Gain Ratio	66.2%	85.5%	79.1%
Entropy	Info Gain	69%	90.3%	80.9%
	Gini Index	73.4%	92.8%	83.9%
	Gain Ratio	67.5%	89.8%	79.9%

Table 4: Increased Accuracy after Applying the Voting

		Increase Accuracy
Equal Width	Info Gain	3.5%
	Gini Index	2.7%
	Gain Ratio	5.5%
Equal Frequency	Info Gain	5.7%
	Gini Index	4.5%
	Gain Ratio	6.4%
ChiMerge	Info Gain	1.2%
	Gini Index	1.2%
	Gain Ratio	4%
Entropy	Info Gain	2.8%
	Gini Index	5.5%
	Gain Ratio	3.4%

Table 5: Comparing Proposed Model with Previous Results

		Sensitivity	Specificity	Accuracy
Proposed Model (Nine Voting Equal Frequency Discretization Gain Ratio Decision Tree)		77.9%	85.2%	84.1%
Tu et al., 2009	**J4.8 Decision Tree**	72.01%	84.48%	78.9%
	Bagging Algorithm	74.93%	86.64%	81.41%

From these results it is concluded that although most researchers are using binary discretization with Gain Ratio Decision Tree in the diagnosis of heart disease, applying multi-interval equal frequency discretization with nine voting Gain Ratio Decision Tree provides better results in the diagnosis of heart disease patients. . We surmise that the improvement in accuracy arises from the increased granularity in splitting attributes offered by multi-interval discretization. Combined with Gain Ratio calculations, this likely increases the accuracy of the probability calculation for any given attribute value. Having that higher probability validated by the voting across multiple similar trees further enhances the selection of useful splitting attribute values. These results would benefit from further testing on much larger data sets.

6 Summary

Decision Tree is one of the successful data mining techniques used in the diagnosis of heart disease. Yet its accuracy is not perfect. Most research applies the J4.8 Decision Tree that is based on Gain Ratio and binary discretization. This research systematically tested combinations of discretization, decision tree type and voting to identify a more robust, more accurate method. The supervised discretization methods do not show any enhancement in the Decision Tree accuracy either with or without voting. Applying voting shows increase in the accuracy of different types of Decision Tree. Systematic testing against a widely-used benchmark data set shows that nine voting with equal frequency discretization and Gain Ratio Decision Tree can enhance the accuracy of the diagnosis of heart disease.

7 References

Anbarasi, M., E. Anupriya, et al. (2010). "Enhanced Prediction of Heart Disease with Feature Subset Selection using Genetic Algorithm." International Journal of Engineering Science and Technology Vol. 2(10).

Andreeva, P. (2006). "Data Modelling and Specific Rule Generation via Data Mining Techniques." International Conference on Computer Systems and Technologies - CompSysTech.

Australian Bureau of Statistics. (2010). Retrieved 7-February-2011, from http://www.ausstats.abs.gov.au/Ausstats/subscriber.nsf/0/E8510D1C8DC1AE1CCA2576F600139288/$File/33030_2008.pdf

Bramer, M. (2007). Principles of data mining, Springer.

Das, R., I. Turkoglu, et al. (2009). "Effective diagnosis of heart disease through neural networks ensembles." Expert Systems with Applications, Elsevier 36 (2009): 7675–7680.

De Beule, M., E. Maesa, et al. (2007). "Artificial neural networks and risk stratification: A promising combination." Mathematical and Computer Modelling, Elsevier.

Dougherty, J., R. Kohavi, et al. (1995). "Supervised and unsupervised discretization of continuous features." In: Proceedings of the 12th international conference on machine learning. San Francisco: Morgan Kaufmann: p. 194–202.

ESCAP. (2010). Retrieved 7-February-2011, from http://www.unescap.org/stat/data/syb2009/9.Health-risks-causes-of-death.asp.

Esposito, F., D. Malerba, et al. (1997). "A Comparative Analysis of Methods for Pruning Decision Trees." IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE VOL. 19, NO. 5.

European Public Health Alliance. (2010). Retrieved 7-February-2011, from http://www.epha.org/a/2352

Fayyad, U. M. and B. I. Keki (1992). "On the handing of Continuous-Valued Attributes in Decision Tree Generation." Machine Learning 8 (87-102).

Hall, L. O., K. W. Bowyer, et al. (2000). "Distributed Learning on Very Large Data Sets." In Workshop on Distributed and Parallel Knowledge Discover.

Han, j. and M. Kamber (2006). Data Mining Concepts and Techniques, Morgan Kaufmann Publishers.

Heller, R. F., S. Chinn, et al. (1984). "How well can we predict coronary heart disease? Findings in the United Kingdom Heart Disease Prevention Project." BRITISH MEDICAL JOURNAL.

Helma, C., E. Gottmann, et al. (2000). "Knowledge discovery and data mining in toxicology." Statistical Methods in Medical Research.

Kavitha, K. S., K. V. Ramakrishnan, et al. (2010). "Modeling and design of evolutionary neural network for heart disease detection." International Journal of Computer Science Issues (IJCSI) Vol. 7, Issue 5.

Kerber, R. (1992). "ChiMerge: Discretization of Numeric Attributes." In Proceedings of the Tenth National Conference on Articial Intelligence

Kotsiantis, S. and D. Kanellopoulos (2006). "Discretization Techniques: A recent survey." International Transactions on Computer Science and Engineering Vol.32 (1) pp. 47-58.

Lee, I.-N., S.-C. Liao, et al. (2000). "Data mining techniques applied to medical information." Med. inform.

Li L, T. H., Wu Z, Gong J, Gruidl M, Zou J, Tockman M, Clark RA (2004). "Data mining techniques for cancer

detection using serum proteomic profiling." Artificial Intelligence in Medicine, Elsevier.

Obenshain, M. K. (2004). "Application of Data Mining Techniques to Healthcare Data." Infection Control and Hospital Epidemiology.

Panzarasa, S., S. Quaglini, et al. (2010). "Data mining techniques for analyzing stroke care processes." Proceedings of the 13th World Congress on Medical Informatics.

Paris, I. H. M., L. S. Affendey, et al. (2010). "Improving Academic Performance Prediction using Voting Technique in Data Mining." World Academy of Science, Engineering and Technology 62.

Perner, P. and S. Trautzsch (1998). "Multi-Interval Discretization Methods for Decision Tree Learning." Advances in Pattern Recognition, A. Amin, D. Dori, P. Pudil, and H. Freeman (Eds.),LNCS 1451, Springer Verlag S. 475-482.

Podgorelec, V., P. Kokol, et al. (2002). "Decision Trees: An Overview and Their Use in Medicine." Journal of Medical Systems Vol. 26.

Porter, T. and B. Green (2009). "Identifying Diabetic Patients: A Data Mining Approach." Americas Conference on Information Systems.

Rajkumar, A. and G. S. Reena (2010). "Diagnosis Of Heart Disease Using Datamining Algorithm." Global Journal of Computer Science and Technology Vol. 10 (Issue 10).

Ruben, D. C. J. (2009). "Data Mining in Healthcare: Current Applications and Issues."

Salahuddin and F. Rabbi (2006). "Statistical Analysis of Risk Factors for Cardiovascular disease in Malakand Division." Pak. j. stat. oper. res. Vol.II: pp49-56.

Sandhya, J., P. Deepa Shenoy, et al. (2010). "Classification of Neurodegenerative Disorders Based on Major Risk Factors Employing Machine Learning Techniques." International Journal of Engineering and Technology Vol.2, No.4.

Shahwan-Akl, L. (2010). "Cardiovascular Disease Risk Factors among Adult Australian-Lebanese in Melbourne." International Journal of Research in Nursing 6 (1).

Simons, L. A., J. Simons, et al. (2003). "Risk functions for prediction of cardiovascular disease in elderly Australians: the Dubbo Study." Medical Journal of Australia 178.

Sitar-Taut, V. A., D. Zdrenghea, et al. (2009). "Using machine learning algorithms in cardiovascular disease risk evaluation." Journal of Applied Computer Science & Mathematics.

Srinivas, K., B. K. Rani, et al. (2010). "Applications of Data Mining Techniques in Healthcare and Prediction of Heart Attacks." International Journal on Computer Science and Engineering (IJCSE) Vol. 02, No. 02: 250-255.

Tantimongcolwat, T., T. Naenna, et al. (2008). "Identification of ischemic heart disease via machine learning analysis on magnetocardiograms." Computers in Biology and Medicine, Elsevier 38 (2008): 817 – 825.

Thuraisingham, B. (2000). "A Primer for Understanding and Applying Data Mining." IT Professional IEEE.

Tu, M. C., D. Shin, et al. (2009). "Effective Diagnosis of Heart Disease through Bagging Approach." Biomedical Engineering and Informatics, IEEE.

Wilson, P. W. F., R. B. D'Agostino, et al. (1998). "Prediction of Coronary Heart Disease Using Risk Factor Categories." American Heart Association Journal.

World Health Organization. (2007). Retrieved 7-Febuary 2011, from http://www.who.int/mediacentre/factsheets/fs310.pdf.

Yan, H., J. Zheng, et al. (2003). "Development of a decision support system for heart disease diagnosis using multilayer perceptron." Proceedings of the 2003 International Symposium on vol.5: pp. V-709- V-712.

Empirical Study of Bagging Predictors on Medical Data

Guohua Liang and Chengqi Zhang

The Centre for Quantum Computation & Intelligent Systems, FEIT, University of Technology, Sydney NSW 2007 Australia

gliang@it.uts.edu.au,chengqi@it.uts.edu.au

Abstract

This study investigates the performance of bagging in terms of learning from imbalanced medical data. It is important for data miners to achieve highly accurate prediction models, and this is especially true for imbalanced medical applications. In these situations, practitioners are more interested in the minority class than the majority class; however, it is hard for a traditional supervised learning algorithm to achieve a highly accurate prediction on the minority class, even though it might achieve better results according to the most commonly used evaluation metric, *Accuracy*. Bagging is a simple yet effective ensemble method which has been applied to many real-world applications. However, some questions have not been well answered, e.g., whether bagging outperforms single learners on medical data-sets; which learners are the best predictors for each medical data-set; and what is the best predictive performance achievable for each medical data-set when we apply sampling techniques. We perform an extensive empirical study on the performance of 12 learning algorithms on 8 medical data-sets based on four performance measures: True Positive Rate (*TPR*), True Negative Rate (*TNR*), Geometric Mean (*G-mean*) of the accuracy rate of the majority class and the minority class, and *Accuracy* as evaluation metrics. In addition, the statistical analyses performed instil confidence in the validity of the conclusions of this research.

Keywords: imbalanced class distribution, medical data, bagging predictors and binary classification.

1 Introduction

Finding effective learning methods and improving prediction accuracy are essential goals for most machine learning approaches (Quinlan 1996), and this is especially true for real-world medical applications. Bagging (Breiman 1996) is a simple and effective ensemble learning method. Due to its promising capabilities in improving accuracy of classification prediction over unstable single learners (Breiman 1996), it has been widely used in many applications. The effectiveness of bagging has been investigated empirically and it has been demonstrated that bagging is very effective for decision trees (Quinlan 1996, Breiman 1996, Bauer and Kohavi 1999, Dietterich 2000, Opitz and Maclin 1999), and Neural Networks (West et al. 2005, Opitz and Maclin 1999, Kim and Kang 2010). Even though the existing studies demonstrate the effectiveness of the bagging predictor, it is not clear whether bagging is superior to single learners in the context of imbalanced medical data-sets, nor which predictor is the best performing learning method on each imbalanced medical data-set.

Our previous works investigate the effectiveness of the bagging predictors in general terms (Liang et al. 2011a) and in imbalanced class distribution terms (Liang et al. 2011b, Liang and Zhang 2011). However, the previous conclusions are based on statistical tests that aggregate the data-sets and do not show which learners are the best prediction models for individual medical data-sets, as various prediction models might behave differently for different kinds of data-sets. They also do not show the best achievable predictive performance for each medical data-set using sampling technique.

In the literature, an empirical study of combined classifiers on medical data (Lopes et al. 2008) compared the performance of three classification methods, C4.5 (Quinlan 1986), bagging, and boosting on 16 medical data-sets and 16 generic data-sets. The evaluation was based on the accuracy of these learning methods as a performance measure; their research did not address the challenging issues of medical data-sets: imbalanced class distribution and the unequal costs of mis-classification errors in different classes. Moreover, accuracy is an inappropriate performance measure for evaluating imbalanced data-sets (Maimon et al. 2010, Chawla et al. 2002).

The majority of medical applications involve learning from imbalanced binary classification data-sets in which the proportion of the class distribution is skewed, the number of instances of the majority class is higher than those of the minority class, and practitioners are more interested in the minority class than the majority class, such as breast cancer early detection, in which the minority class is quite small with an unequal high cost associated with mis-classification errors in different classes. If a patient with breast cancer is mis-classified as normal, the patient will miss the opportunity for his/her earlier stage cancer detection and treatment; while if a patient without breast cancer is mis-classified as having cancer, it will cause unnecessary stress and treatment. Traditional supervised learning algorithms perform poorly in predictive accuracy over the minority class, even though they may produce high overall accuracy (Phua et al. 2004, Ng and Dash 2006, Maloof 2003, Su and Hsiao 2007, Chawla 2010). We therefore employ four measures, True Positive Rate (*TPR*), True Negative Rate (*TNR*), geometric mean (*G-mean*) of the accuracy rate of the majority class

and minority class, and *Accuracy* as evaluation metrics to assess the effectiveness of bagging in terms of learning from medical data-sets.

To solve the problem of imbalanced class distribution and increase the *Accuracy* of the prediction model, the most commonly used methods are sampling-oriented methods and algorithms-oriented methods (Liu and Chawla 2011).

In this study, we utilize under-sampling techniques to investigate the performance of bagging predictors at different levels of class distribution and report the best achieved performance of bagging by using sampling techniques based on the *G-mean* evaluation metrics.

The main objectives of this paper are threefold: we (1) determine whether bagging is superior to single learners in the context of imbalanced medical data-sets, (2) determine which learners give the best performance on each medical data-set with natural class distribution, and (3) report the best achieved performance of the bagging predictors on each medical data-set by using sampling techniques.

The paper is organized as follows. Section 2 presents details of the designed framework. Section 3 presents sampling techniques and Section 4 presents the evaluation metrics. Section 5 presents the experimental setting and Section 6 presents the experimental results analysis. Section 7 concludes the paper.

2 Designed Framework

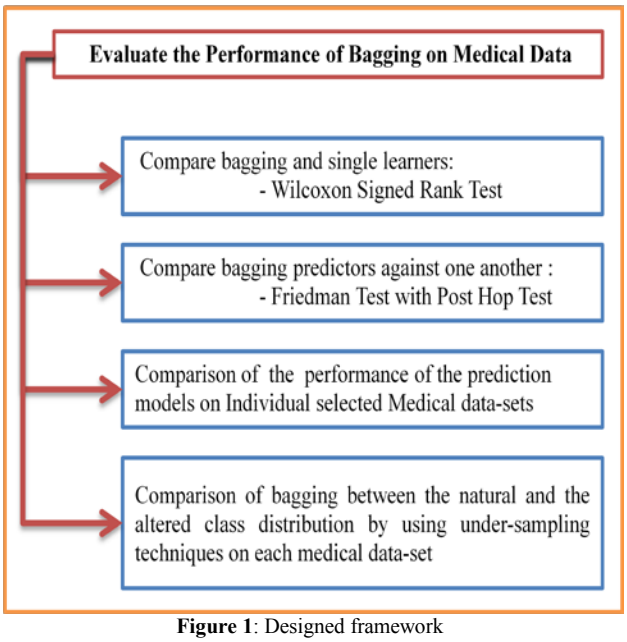


Figure 1: Designed framework

The designed framework and the evaluation of bagging predictors on Medical Data-Sets are broken down into four tasks as follows:

- Compare bagging predictors with single learners: the Wilcoxon Signed Ranks Test is used to compare two learners to determine whether bagging outperforms a single learner on medical data-sets.
- Compare the performance of bagging predictors against one another: the Friedman test with the corresponding Post-hoc Nemenyi test is used to compare multiple learners to determine which bagging predictors have the best performance over all 8 imbalanced data-sets,
- Compare the performance of the prediction and report the best performance models with the natural class distribution on each individual medical data-set based on four evaluation metrics: *G-mean, TPR, TNR* and *accuracy rate*.
- Compare the performance of bagging predictors between the natural class distribution and the altered levels of class distribution to determine the best performance of the bagging predictor on each medical data-set.

3 Sampling Techniques

Sampling techniques are commonly used to improve the performance of the prediction model for imbalanced data-sets (Chawla et al. 2002, Chawla et al. 2003, Weiss and Provost 2003), e.g., under-sampling and over-sampling SMOTE (Chawla et al. 2002), Borderline-SMOTE (Han et al. 2005), and Safe-Level-SMOTE (Bunkhumpornpat et al. 2009).

We utilize under-sampling techniques to vary the class distribution of the data to investigate the performance of bagging predictors over medical data-sets, i.e., to alter each original imbalanced data-set, D with sample size M into nine new data-sets, $D_1, D_2 \ldots D_9$ with new sample size $M_1, M_2 \ldots M_9$, respectively.

We consider the entire minority class samples as a positive class (P) and the proportions of P are as follows: $P = 10\%\ M_1 = 20\%\ M_2 = \ldots = 90\%\ M_9$, respectively. Then we select the majority class randomly without replacement as a negative class (sample size $N_1, N_2 \ldots N_9$), and the proportions of the negative class are as follows: $N_1 = 90\%\ M_1$; $N2 = 80\%\ M_2 \ldots N_9 = 10\%\ M_9$, respectively to form the new data-sets, $D_1, D_2 \ldots D_9$. Each original imbalanced data-set D is thereby altered into nine different levels of class distributions.

10 trials 10-fold cross-validation is performed on each of the new data-sets, $D_1, D_2 \ldots D_9$, so that the test-set has the same distributions as the training-set. We then compare the results of *G-mean* from nine different class distributions on each medical data-set and report the best results achieved on each data-set using sampling techniques.

4 Evaluation Metrics

Accuracy is a popular choice for evaluating the performance of a classifier; however, it might not be a good metric for measuring the performance of medical data-sets. The challenge issues of the most medical applications are imbalanced class distribution problem and unequal costs of the mis-classification errors in different classes. The minority class is more important than the majority class; normally a high prediction accuracy is required in a minority class and therefore a simple estimated accuracy has limitations in evaluating the performance of a classifier on a minority class (Fawcett 2006). We therefore adopt four measures, *Accuracy*, True Positive Rate (*TPR*), True Negative Rate (*TNR*), and *G-mean* as evaluation metrics.

In this paper, we consider the minority class as the positive class and the majority class as the negative class. Following this convention, *TP* refers to the number of positive instances correctly classified as the positive class; *TN* refers to the number of negative instances correctly classified as the negative class; *FP* refers to the number of negative instances incorrectly classified as the positive class; and *FN* refers to the number of positive instances incorrectly classified as the negative class (Chawla 2010, Guo et al. 2008).

Accuracy (*Acc*) is commonly used as a performance measure of a classifier for balanced learning. However, it has been considered an improper performance measure for evaluating learning from imbalanced data (He and Garcia 2009, Provost et al. 1998, Maloof 2003, Weiss and Provost 2003).

TPR and *TNR* evaluate the performance of a binary classification algorithm directly on the minority class and the majority class respectively. *TPR* refers to the proportion of the minority class that has been correctly classified as a positive class, while *TNR* refers to the proportion of the majority class that has been correctly classified as a negative class. The *G-mean* of the accuracy rate of the majority class and minority class was suggested as a performance measure to assess the effectiveness of learning methods for imbalanced learning (Ng and Dash 2006, He and Garcia 2009, Provost and Fawcett 2001). Table 1 presents the confusion matrix for a binary classification problem. Table 2 presents the formulas of both True Positive Rate and True Negative Rate in the first row, the formula of *G-mean* in the second row, and the formula of Accuracy *(Acc)* in the last row.

Table 1: Confusion matrix for a binary classification problem

	Predicted Positives	**Predicted Negatives**
Positive Instances *(P)*	*True Positive (TP)*	*False Negatives (FN)*
Negative Instances *(N)*	*False Positive (FP)*	*True Negatives (TN)*

Table 2: True Positive Rate, True Negative Rate and G-mean

$TP_{rate} = \frac{TP}{TP + FN}$	$TN_{rate} = \frac{TN}{TN + FP}$
$G - mean = \left(TP_{rate} * TN_{rate}\right)^{1/2}$	
$Acc = \frac{TP+TN}{TP+TN+FP+FN}$	

5 Brief Overview of Single Learner and Bagging

In this section, we briefly introduce two basic concepts: what constitutes a single learner of supervised learning and what is bagging.

Single learner refers to supervised learning using the labelled samples to form a classifier (called a single learner or prediction model) and having a function that can be used to predict new samples with pre-defined class labels. Figure 2 presents a prediction model of a single learner in supervised learning.

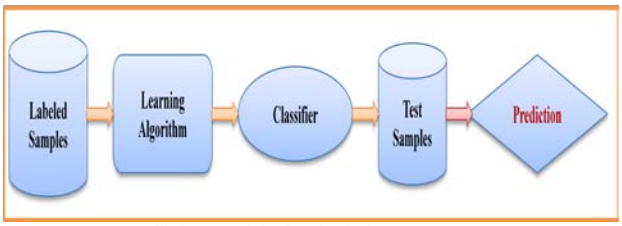


Figure 2: Prediction model of a single learner

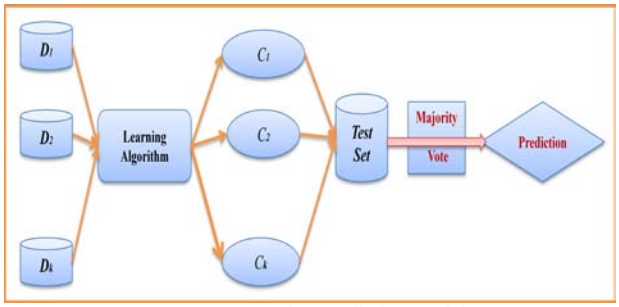


Figure 3: Bagging prediction model

Bagging represents a set of classifiers ($C_1, C2 \ldots C_k$) (called base learners) which are generated from a set of bootstrap samples ($D_1, D_2 \ldots D_k$) to form an ensemble method for prediction, and its function is to predict new samples by a set of classifiers; a final prediction is made by taking a majority vote .

Figure 3 illustrates the basic framework of a bagging prediction model by using bootstrap sampling and voting techniques to improve the performance of the bagging prediction model. Bagging is known as "bootstraps aggregating". Firstly, for each of the bootstrap samples ($D_1, D_2 \ldots D_k$), a new training set D_k is randomly drawn from the original training set D of m instances with replacement conducted by repeated drawing m times. Each bootstrap sample therefore contains the same number of m instances as the original training set D; some instances may appear many times, while some instances may not appear. Secondly, the k bootstrap samples of a training set with m instances will generate k classifiers ($C_1, C_2 \ldots C_k$). Finally, the unseen instance x of the test set will be predicted by applying each of the k classifiers C_i (i =1 to k) and a final decision C^* is made by majority vote of all classifiers ($C_1 \ldots C_k$). The algorithm for bagging is given in Figure 4.

Algorithm 1: Bagging

Input:
- D, a set of n training instances;
- k, the number of Boostrap samples;
- a Learning scheme (e.g. J48, decision tree algorithm)

Output: A composite model, C^*.

Method:

for i = 1 to k do
- Create bootstrap sample of size n, D_i by sampling D with replacement;
- Train a base classifier model C_i from D_i;

end

To use the composite model, C^* for Test set T on a instance, x and it's true class label is y:

$C^*(x) = \arg\max_y \sum_i \delta(C_i(x) = y)$

Delta function $\delta(\cdot) = 1$ if argument is true, else 0.

Figure 4: Algorithm of Bagging (Breiman 1996)

6 Experimental Setting

This section includes three subsections as follows: A. software and parameter settings, B. selection of base learners, and C. data-set selection.

6.1 Software and Parameter Settings

We performed 10-trial 10-fold cross-validations to evaluate bagging and single learners on 8 medical data-sets, which were collected from the UCI Machine Learning Repository (Merz and Murphy 2006). We used WEKA implementation of the 12 algorithms with their default parameter settings in this empirical study (Witten and Frank 2005). We implemented the bagging predictor in Java platform. In order to reduce uncertainty and obtain reliable experimental results, all the evaluations of bagging performance are assessed under the same test conditions by using the same randomly selected bootstrap samples with replacements in each fold of 10-trial 10-folds cross-validation on each data-set.

6.2 Selection of Base Learners

Twelve learning algorithms have been selected for this study. We first select the most commonly used learning algorithms in real-world applications: Support Vector Machines (SVM), Neural Network learner – Multi Layer Proceptron (MLP), Naïve Bayes learner (NB), and K-nearest-neighbours (KNN). We then select rule learners: PART, Decision Table (DTable), and OneR. We finally select tree family learners, C4.5 Decision Tree (J48), DecisionStump (DStump), RandomTree (RandTree), REPTree and Naïve-Bayes-Trees (NBTree).

6.3 Selection of Data-Sets

Table 3: Imbalanced Medical Data-Sets

		Information Data		Class Data		
ID	Name	attribut	instance	frequency	P%	clas
1	breastc	10	286	201,85	29%	2
2	diabetes	9	768	500,268	34%	2
3	heart-c	14	303	165,138	45%	2
4	sick	30	3772	3541,231	6%	2
5	heart-h	14	294	188,106	36%	2
6	stalogHe	14	270	120,150	44%	2
7	wbreastc	10	699	458,241	34%	2
8	WDBC	31	569	212,357	37%	2

A summary of the characteristics of the eight imbalanced medical data-sets is displayed in Table 3. The selected medical data-sets are binary classes. The selection of the eight data-sets covers the number of instances, which varies from small to large up to 3772, the number of attributes, which varies from 9 to 31, and the natural class distribution (*P%*), which indicates the percentage of the positive instances from the total instances of each data-set. The results vary from 6.1%, the extremely imbalanced data-set 'sick' to 45% the almost balanced data-sets 'heart-c' and 'stalogHeart'.

7 Experimental Results Analysis

This section presents the experimental results analysis including four sub-sections as follows: A. comparison of bagging with single learners, B. comparison of bagging predictors on medical data-sets, C. comparison of the performance of 24 prediction models and report the best prediction model on each individual data-set, and D. comparison of the performance of bagging predictors between natural class distribution and the altered class distribution by using under-sampling techniques on each medical data-set.

7.1 Comparison of Bagging and Single learners

This subsection compares bagging and single learners over multiple medical data-sets to determine whether bagging is superior to single learners based on two evaluation metrics, *Accuracy* and *G-mean*.

The Wilcoxon Signed Rank Test is used to compare two learners - bagging and a single learner over multiple data-sets - to determine whether bagging is superior to a single learner.
The Null Hypothesis is that the median of differences between bagging and a single learner equals 0.
Rule: Reject the Null Hypothesis if the p-value Test Statistic W is less than .05 at the 95% confidence level of significance.

Table 4: Compare bagging with each single learner based on Wilcoxon Signed Rank Test on *Accuracy*. The significance level is .05.

Wilcoxon Signed Rank Test on Accuracy						
Learners	J48	RepTree	Randtree	NB	SVM	Dstump
p-value	.025	.012	.012	**.207**	**.138**	**.128**
Learners	OneR	Dtable	PART	KNN	NBTree	MLP
p-value	.012	**.208**	.012	**0.092**	.017	.012

Tables 4 and 5 present the summarized results of the Wilcoxon Signed Rank Test on the evaluation metrics, *Accuracy* and *G-mean* for the comparison of the two learners: single learners versus their corresponding bagging predictors, i.e., we compare bagging J48 and single learner J48. If the p-value is greater than α value, .05, we accept the Null Hypothesis and the p-values are highlighted.

Table 4 indicates that bagging does not perform statistically significantly better than the single learners NB, SVM, Dstump, Dtable and KNN on eight Medical data-sets based on the evaluation metric, *Accuracy*. Table 5 indicates that bagging is statistically superior to the single learners J48, RandTree, OneR, PART and MLP on eight medical data-sets based on the *G-mean* evaluation metric.

Table 5: Compare bagging with each single learner based on Wilcoxon Signed Rank on *G-mean*. The significance level is .05.

Wilcoxon Signed Rank Test on Gmean.						
Learnr	J48	RepTree	Randtree	NB	SVM	Dstump
p-value	.036	**.161**	.036	**.069**	**.093**	**.866**
Learnr	OneR	Dtable	PART	KNN	NBTree	MLP
p-value	.017	**.779**	.036	**.327**	**.484**	.012

7.2 Comparison of the Performance of Bagging Predictors on Imbalanced Medical Data-Sets

Friedman Test and Post-hoc Nemenyi Test: Both tests are non-parametric for comparing multiple algorithms over multiple datasets. Firstly, all the algorithms are ranked on each data-set, giving the best performing algorithm the rank of 1, the second best rank 2, and so on. If there are ties, average values are assigned. Secondly, the average rank of the algorithm is calculated. Finally, the Friedman test compares the average ranks of algorithms and checks whether there is a significant difference between the mean ranks.

The Null Hypothesis of this test states that the performances of all algorithms are equivalent. If the Null Hypothesis is rejected, it does not determine which particular algorithms differ from one another. Because the test result does not show exactly where that significant difference occurs, a post-hoc Nemenyi test is needed for additional exploration of the differences between mean ranks to provide specific information on which mean ranks are significantly different from on another. The critical difference is calculated as:

$$CD = q_{\alpha}\sqrt{\frac{d(d+1)}{6N}}$$

Where d is the number of algorithms, N is the number of data-sets, and the critical values q_{α} are based on the Studentized range statistic divided by $\sqrt{2}$. If the mean ranks are different by at least the critical difference, the performance of learners is significantly different. Demšar has presented how to calculate the critical difference of the Nemenyi test in more detail (Demšar 2006).

Table 6: Ranking order of the performance of bagging based on *G-mean* and their Mean Ranks.

Gmean	MLP	NB	NBTree	SVM	PART	RdTree
breastc	2	1	7	6	5	8
diabetes	1	3	2	8	7	5
sick	10	9	5	12	3	7
heart-c	3	2	4	1	5	7
staHeart	3	1	4	2	5	7
heart-h	3	1	4	2	5	6
wdbc	2	10	4	1	3	6
wbreastc	3	1	2	4	6	5
Mean Rank	3.375	3.5	4	4.5	4.875	6.375
	J48	RepTree	Dstump	KNN	Dtable	OneT
breastc	9	11	3	4	12	10
diabetes	4	6	10	11	9	12
sick	1	4	2	11	8	6
heart-c	6	8	10	11	9	12
staHeart	10	6	12	9	8	11
heart-h	10	11	7	8	12	9
wdbc	7	8	11	5	9	12
wbreastc	7	9	12	8	10	11
Mean Rank	6.75	7.875	8.375	8.375	9.625	10.375

Table 6 presents the ranking order of the performance of bagging predictors on each imbalanced medical data-set based on the evaluation metric *G-mean*. Firstly, we divide Table 6 into two parts. In each part, the first row presents the ascending order of the bagging predictors according to their mean rank of the *G-mean* measure in the 10th row. Secondly, the second to ninth rows present the ranking order of the bagging predictors on each individual medical data-set, e.g., bagging MLP performs best on the diabetes data-set ranking as 1, followed by bagging NBTree ranking as 2, and bagging OneR ranked 12 is the worst bagging predictor on the same data-set. The last rows present the mean ranks of the performance of the bagging predictor over all eight medical data-sets. On the other hand, we observe that different bagging predictors behave differently for different medical data-sets, e.g., bagging MLP performs well on most of these medical data-sets, except for sick data-set which is an extremely imbalanced and high dimensional large data-set; bagging NB performs best (ranking as 1) on *four* medical data-sets, breastc, StatlogHeart, heart-h and wbreastc, but performs poorly on the other two data-sets, sick and WDBC, which are high dimensional attributes or extremely imbalanced class distribution data-sets; while bagging J48 and DStump perform well on the extremely imbalanced and high dimensional largest medical data-set, sick.

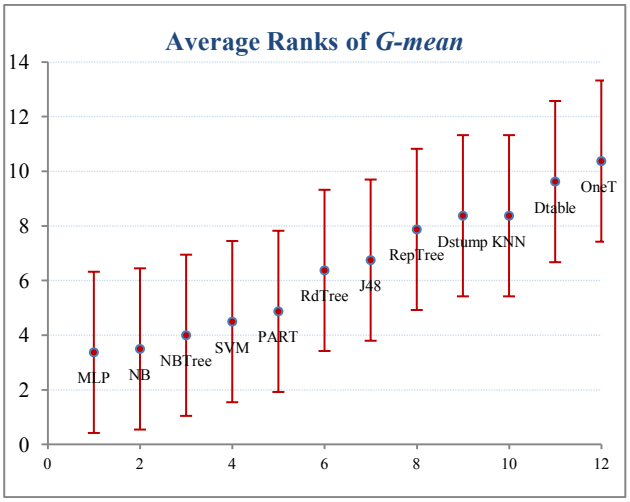


Figure 5: Comparison of all bagging predictors from the Friedman and Post-hoc Nemenyi test, where the x-axes indicate the mean rank of each bagging predictor, the y axes indicate the ascending ranking order of the Bagging predictors, and the horizontal error bars indicate the "critical difference".

Figure 5 presents the results of the mean ranking of the performance of bagging predictors over all eight medical data-sets based on the Friedman and Post-hoc Nemenyi tests. The results indicate that the group of bagging MLP and NB are the best bagging predictors, while bagging OneR is the worst bagging predictor. The performances of two bagging predictors are significantly different if the horizontal bars do not overlap; therefore, there is a statistically significant difference between the group of two best bagging predictors, MLP and NB and the worst bagging predictor OneR. However, there is not a statistically significant difference between remaining bagging predictors.

7.3 Comparison of the Performance of the Prediction Models on Individual Medical Data-Sets

In this subsection, we compare the performance of the prediction models, bagging predictors and single learners on eight selected medical data-sets. For the data-set selection, we first select breastc data-set which has 10 attributes and 286 instances, in which the proportion of the minority class is 29%; secondly, we select three moderately imbalanced data-sets, WDBC, heart-h, diabetes and wbreastc in which the proportions of the minority class are 37%, 36%, 34% and 34%, respectively; thirdly, we select an extremely imbalanced data-set, sick, which has 30 attributes and 3772 instances, in which the proportion of the minority class is 6%. Finally, we select two almost balanced data-sets, heart-c and stalogHeart data-sets, in which the proportions of the minority class are about 45%.

Figures 5 to 13 inclusive present a comparison of the performance of all the prediction models on eight medical data-sets, breastc, diabetes, sick, heart-h, WDBC, heart-c, wbreastc and statlogHeart. Each graph presents the summarization of the observed performance of the prediction models based on four measures, *G-mean, TPR, TNR* and *Accuracy* on each of the selected data-sets. For each plot, the horizontal axis indicates the ranking order of all the prediction models based on the descending order of the performance measure, *G-mean*, while the vertical axis indicates the value of the four performance measures.

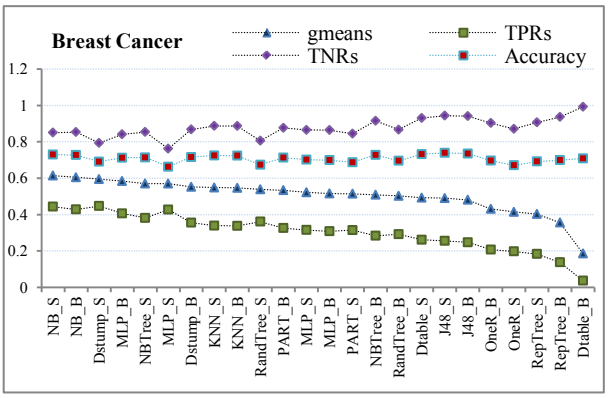


Figure 6: The performance of prediction models on breastc data-set.

Figure 6 shows that both single learner NB and bagging NB perform better than the other prediction models, followed by the simple learner DStrump and bagging MLP. The group of learners, bagging DTable, bagging RepTree, RepTree, OneR and bagging OneR are the worst prediction models for the breastc data-set based on the performance measure *G-mean* and *TPR*. Even though the performance of *Accuracy* seems reasonably good for all the prediction models, it does not present the accuracy of the minority class. Because the performance of accuracy is influenced by the *TNR*, this observation is consistent with the existing research.

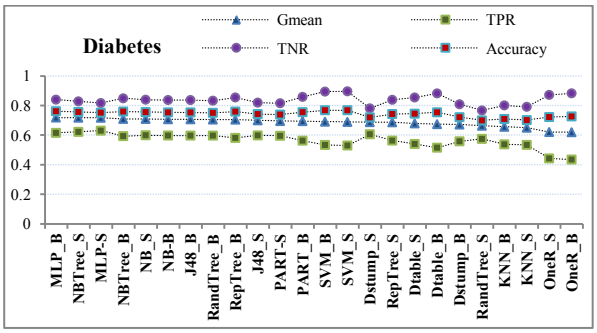


Figure 7: Comparison of the performance of prediction models on diabetes data-set.

Figure 7 presents the comparison of the performance of the prediction models on the diabetes data-set. The group of bagging MLP, NBTree, MLP and bagging NBTree are the best prediction models on this data-set, followed by NB and Bagging NB; while the group of learners, Bagging KNN, KNN, OneR and bagging OneR are the worst prediction models on this data-set.

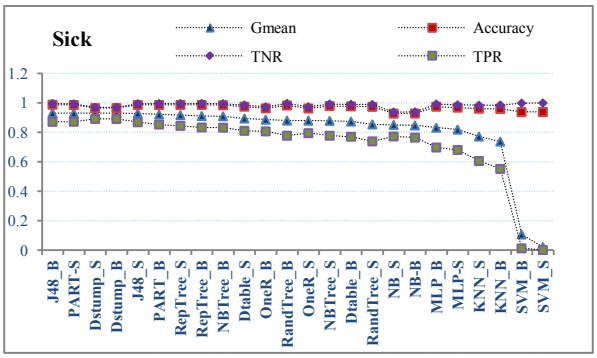


Figure8: Comparison of the performance of prediction models on sick data-set.

Figure 8 presents a comparison of the performance of the prediction models on the extremely imbalanced data-set, sick. We observe that *Accuracy* and *TNR* perform well for all the prediction models, because *Accuracy* is influenced by the *TNR* on this extremely imbalanced data-set. However, regarding the performance measures, *TPR* and *G-mean* of the accuracy of both the majority class and minority class, we observe that bagging J48 and PART perform best, followed by single DStump, bagging DStump, J48 and bagging PART, while the group learners, bagging KNN and SVM, and their single learners are the worst prediction models for this medical data-set.

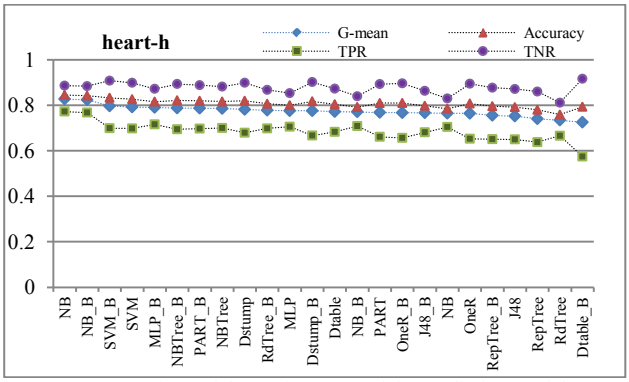


Figure 9: Comparison of the performance of the prediction models, bagging predictors and single learners on heart-h data-set.

Figure 9 presents a comparison of the performance of prediction models on the almost balanced heart-h data-set. Most prediction models perform well on this data-set, except the group of weak learners DStump and its bagging predictors. The group of learners, NB, bagging NB and SVMP, and bagging SVM are the best prediction models on this data-set.

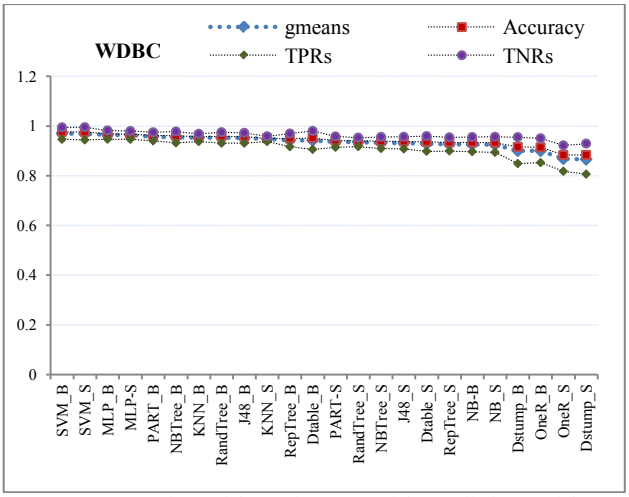


Figure 10: Comparison of the performance of the prediction models, bagging predictors and single learners on WDBC data-set.

Figure 10 presents a comparison of the performance of prediction models on the moderately imbalanced WDBC data-set. Most prediction models perform well on this data-set, except the group of weak learners, OneR, DStump, and their bagging predictors. The group of learners, bagging SVM, SVM, bagging MLP and MLP are the best prediction models on this data-set.

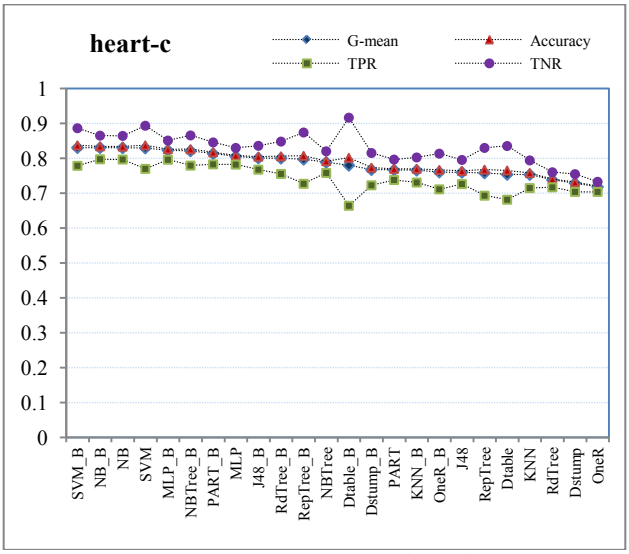


Figure 11: Comparison of the performance of the prediction models, bagging predictors and single learners on heart-c data-set.

Figure 11 presents a comparison of the performance of prediction models on the almost balanced heart-c data-set. The group of learners, bagging SVM, bagging NB, and NB are the best prediction models on this data-set, followed by SVM and bagging MLP; while the group of learners, KNN, Dstump and OneR are the worst prediction models on this data-set.

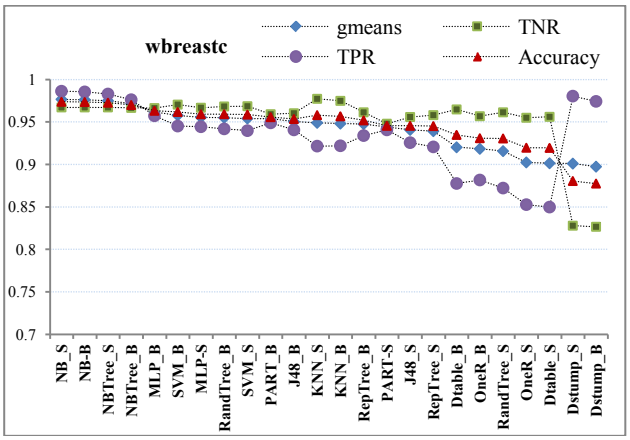


Figure 12: Comparison of the performance of the prediction models, bagging predictors and single learners on wbreastc data-set.

Figure 12 presents a comparison of the performance of the prediction models on the wbreastc data-set. The group of learners, NB, bagging NB, NBTree, and bagging NBTree are the best prediction models on this data-set, followed by bagging MLP and Bagging SVM; while the group of learners, Dstump and bagging Dstump are the worst prediction models on this data-set.

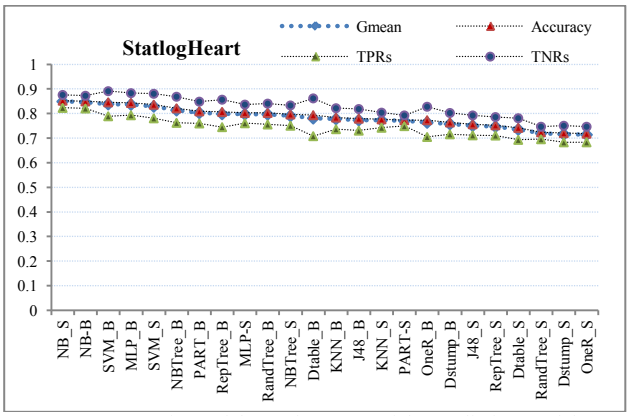


Figure 13: Comparison of the performance of the prediction models, bagging predictors and single learners on StatlogHert data-set.

Figure 13 presents a comparison of the performance of prediction models on the almost balanced StatlogHeart data-set. The group of learners NB and bagging NB are the best prediction models on this data-set, followed by bagging SVM and bagging SVM; while the group of learners, randTree, Dstump and OneR are the worst prediction models on this data-set.

Table 7: Best performance model for the natural class distribution on each individual data-sets.

	Best performance Model					P%
Name	G-mean	Err	TPR	TNR	Learners	
Heart-h	0.8239	0.1578	0.7679	0.8840	NB	0.36
Heart-c	0.831	0.1624	0.779	0.8867	SVM_B	0.45
stalogHeart	0.8492	0.1474	0.8233	0.876	NB	0.44
WDBC	0.97	0.0236	0.9462	0.9944	SVM_B	0.37
diabetes	0.7188	0.2384	0.6153	0.84	MLP_B	0.34
wbreastc	0.9767	0.0262	0.9672	0.9863	NB	0.34
breastc	0.6142	0.2703	0.4435	0.8507	NB	0.29
sick	0.932	0.0117	0.9959	0.8723	J48_B	0.06

Table 7 reports the best performance of prediction models for the natural class distribution on each individual medical data-set. Bagging predictors, SVM, MLP and J48 are the best prediction models for heart-c, WDBC, diabetes, and sick data-sets, respectively, while single learner NB is the best prediction models for heart-h, stalogHeart, wbreastc and breastc data-sets.

7.4 Comparison of the Performance of Bagging between the Natural Class Distribution and the Altered Class Distribution by Using Sampling Techniques on Each Medical Data-Set

In this subsection we report the performance of bagging predictors between the natural class distribution and the best achieved results by using sampling techniques on each medical data-set.

Tables 7 and 8 present the comparison of the performance of the bagging predictors between natural class distribution and the best achieved results by using sampling techniques on four medical data-sets. The first column indicates the name of a medical data-set and the bagging predictors; the second column presents the results from the natural class distribution which include *TPR, TNR and G-mean* of the accuracy rate on majority class and minority class; the third column presents the best achieved results based on *G-mean* by using sampling techniques which include *G-mean, TPR, TNR* and the proportion of the positive instances (*P%*) which refers to the level of the altered class distribution when bagging achieves the best performance on the *G-mean* measure. We also note that if the proportion of positive instances increases, the *TPR* will also increase but the *G-mean* may reduce.

The experimental results in the third column indicate the best achieved bagging performance based on the G-mean measure: the level of the class distribution is mostly about 50% on breastc, heart-c, and statlogHeart data-sets. This finding is consistent with previous research. However, the levels of class distribution are mostly 40% on WDBC and heart-h data-sets, and 30% on sick data-set, respectively, when the best bagging performance on the *G-mean* measure is achieved. In addition, there are interesting findings on both WDBC and sick data-sets in that when bagging NB achieves the best performance on the *G-mean* measure, the level of class distributions are 10% and 20% highlighted, respectively. This finding may be inconsistent with existing research, which assumes that traditional learning algorithms will perform better in a balanced situation than in an imbalanced situation.

The experimental results demonstrate that the sampling techniques can improve the performance of bagging predictors on the *G-mean* of the accuracy on the majority class and minority class over most medical data-sets, except for bagging OneR on the breastc data-set whose result is marked in red. The bagging performance on the TPR and TNR measures also improved at the same level of class distribution, except for NB on heart-h data-set with TNR measure marked as red.

.

breastc	**Natural Class Distribution**			**Sampling**			
Bagging	TPR	TNR	G-mean	G-mean	TPR	TNR	P%
J48	0.247	0.941	0.481	0.724	0.717	0.737	50%
RepTree	0.138	0.937	0.356	0.678	0.651	0.709	50%
RandTree	0.292	0.867	0.503	0.796	0.837	0.580	40%
NB	0.428	0.854	0.605	0.675	0.644	0.709	50%
SVM	0.308	0.865	0.516	0.690	0.695	0.685	50%
DStump	0.355	0.868	0.552	0.630	0.486	0.824	50%
OneR	0.207	0.904	0.431	0.619	0.505	0.769	50%
DTable	0.037	0.993	0.186	0.662	0.527	0.835	50%
PART	0.326	0.877	0.534	0.746	0.760	0.733	50%
KNN	0.338	0.887	0.546	0.802	0.782	0.822	50%
NBTree	0.284	0.915	0.509	0.731	0.732	0.732	50%
MLP	0.406	0.841	0.584	0.790	0.682	0.916	70%
heart-c	Natural			Sampling			
Bagging	TPR	TNR	G-mean	G-mean	TPR	TNR	P%
J48	0.7681	0.8364	0.8013	0.8872	0.8807	0.8941	50%
RepTree	0.7275	0.8745	0.7976	0.8496	0.82	0.8807	50%
RandTree	0.7558	0.8485	0.8007	0.9128	0.9052	0.9207	50%
NB	0.7978	0.8655	0.8309	0.8404	0.8019	0.8815	40%
SVM	0.779	0.8867	0.831	0.8461	0.8833	0.8109	60%
DStump	0.7232	0.8158	0.7679	0.7778	0.7454	0.8123	40%
OneR	0.7116	0.8139	0.761	0.7642	0.7407	0.7889	50%
DTable	0.6645	0.917	0.7804	0.8471	0.7904	0.9081	50%
PART	0.7833	0.8461	0.8139	0.9061	0.8956	0.917	50%
KNN	0.7312	0.803	0.7662	0.8983	0.9015	0.8956	50%
NBTree	0.7797	0.8661	0.8217	0.904	0.88	0.9289	50%
MLP	0.7964	0.8515	0.8234	0.9091	0.9074	0.9111	50%
Statlog Heart	Natural			Sampling			
Bagging	TPR	TNR	G-mean	G-mean	TPR	TNR	P%
J48	0.731	0.819	0.773	0.870	0.878	0.863	50%
RepTree	0.745	0.857	0.799	0.860	0.859	0.862	50%
RandTree	0.756	0.841	0.797	0.900	0.895	0.905	50%
NB	0.821	0.873	0.846	0.854	0.844	0.865	50%
SVM	0.789	0.891	0.839	0.865	0.853	0.877	50%
DStump	0.716	0.803	0.758	0.780	0.716	0.854	30%
OneR	0.705	0.828	0.764	**0.740**	0.726	0.756	50%
DTable	0.708	0.862	0.781	0.836	0.872	0.803	50%
PART	0.760	0.849	0.803	0.892	0.855	0.931	40%
KNN	0.737	0.822	0.778	0.891	0.863	0.919	40%
NBTree	0.763	0.869	0.814	0.900	0.856	0.946	40%
MLP	0.793	0.883	0.837	0.912	0.883	0.941	40%
heart-h	Natural			Sampling			
Bagging	TPR	TNR	G-mean	G-mean	TPR	TNR	P%
J48	0.6811	0.8633	0.7668	0.8548	0.8543	0.8562	50%
RepTree	0.6509	0.8771	0.7555	0.8282	0.8814	0.7794	60%
RandTree	0.6981	0.8676	0.7781	0.9035	0.8731	0.9353	40%
NB	0.7679	0.884	0.8239	0.8369	0.8076	0.8676	50%
SVM	0.6991	0.908	0.7966	0.8354	0.8794	0.7941	60%
DStump	0.667	0.9021	0.7757	0.7963	0.7412	0.8588	60%
OneR	0.6566	0.8963	0.7671	0.7944	0.801	0.7897	60%
DTable	0.5745	0.9165	0.7254	0.8297	0.8048	0.8571	50%
PART	0.6972	0.8883	0.7869	0.8665	0.8279	0.9077	40%
KNN	0.7085	0.8394	0.7711	0.8951	0.8731	0.9179	40%
NBTree	0.6943	0.8936	0.7876	0.8728	0.8346	0.9135	40%
MLP	0.716	0.8723	0.7903	0.8927	0.8596	0.9276	40%

Table 7: Compare the performance of bagging predictors on the G-mean measure between the natural class distribution and the altered class distribution by using sampling techniques on four data-sets: breastc, heart-c, statlogHeart and heart-h.

diabetes	Natural			Sampling			
Bagging	TPR	TNR	G-mean	G-mean	TPR	TNR	P
J48	0.597	0.837	0.707	0.861	0.834	0.890	40%
RepTree	0.581	0.855	0.705	0.824	0.780	0.871	40%
RandTree	0.597	0.833	0.705	0.876	0.844	0.909	40%
NB	0.597	0.837	0.707	0.726	0.740	0.712	60%
SVM	0.534	0.894	0.691	0.741	0.700	0.785	50%
DStump	0.558	0.809	0.672	0.696	0.620	0.792	40%
OneR	0.435	0.883	0.620	0.719	0.723	0.716	50%
DTable	0.515	0.882	0.674	0.776	0.778	0.774	50%
PART	0.563	0.859	0.695	0.852	0.824	0.881	40%
KNN	0.538	0.801	0.656	0.848	0.816	0.881	40%
NBTree	0.593	0.849	0.710	0.840	0.803	0.879	40%
MLP	0.615	0.840	0.719	0.812	0.833	0.793	50%
sick	Natural			Sampling			
Bagging	TPR	TNR	G-mean	G-mean	TPR	TNR	P
J48	0.872	0.996	0.932	0.973	0.967	0.979	30%
RepTree	0.834	0.997	0.912	0.965	0.954	0.976	30%
RandTree	0.778	0.997	0.881	0.972	0.964	0.980	40%
NB	0.765	0.939	0.848	0.880	0.864	0.898	**20%**
SVM	0.013	0.999	0.107	0.892	0.857	0.930	30%
DStump	0.892	0.970	0.930	0.934	0.896	0.974	70%
OneR	0.807	0.974	0.887	0.934	0.898	0.971	30%
DTable	0.771	0.991	0.874	0.941	0.902	0.982	30%
PART	0.854	0.995	0.922	0.973	0.967	0.979	30%
KNN	0.552	0.986	0.738	0.912	0.908	0.915	40%
NBTree	0.833	0.995	0.910	0.974	0.964	0.984	30%
MLP	0.698	0.993	0.832	0.951	0.964	0.938	50%
WDBC	Natural			Sampling			
Bagging	TPR	TNR	G-mean	G-mean	TPR	TNR	P
J48	0.931	0.972	0.951	0.974	0.968	0.981	40%
RepTree	0.917	0.970	0.943	0.968	0.959	0.978	40%
RandTree	0.931	0.975	0.953	0.979	0.971	0.986	40%
NB	0.897	0.956	0.926	0.937	0.905	0.970	**10%**
SVM	0.946	0.994	0.970	0.977	0.966	0.987	50%
DStump	0.849	0.955	0.900	0.925	0.940	0.910	70%
OneR	0.852	0.950	0.900	0.929	0.900	0.959	40%
DTable	0.906	0.980	0.942	0.958	0.961	0.954	40%
PART	0.940	0.975	0.957	0.979	0.977	0.981	40%
KNN	0.937	0.969	0.953	0.980	0.980	0.981	50%
NBTree	0.932	0.978	0.955	0.979	0.976	0.981	50%
MLP	0.947	0.982	0.964	0.979	0.972	0.985	50%
wbreastc	Natural			Sampling			
Bagging	TPR	TNR	G-mean	G-mean	TPR	TNR	P
J48	0.941	0.960	0.950	0.964	0.967	0.961	40%
RepTree	0.934	0.961	0.948	0.961	0.964	0.958	50%
RandTree	0.942	0.968	0.955	0.982	0.983	0.981	40%
NB	0.986	0.967	0.976	0.981	0.985	0.976	60%
SVM	0.945	0.971	0.958	0.979	0.988	0.970	80%
DStump	0.974	0.827	0.897	0.908	0.981	0.840	30%
OneR	0.882	0.957	0.918	0.935	0.962	0.908	60%
DTable	0.878	0.965	0.920	0.960	0.981	0.940	60%
PART	0.949	0.959	0.954	0.964	0.969	0.958	40%
KNN	0.922	0.975	0.948	0.981	0.982	0.981	60%
NBTree	0.976	0.967	0.972	0.983	0.983	0.984	30%
MLP	0.957	0.966	0.962	0.979	0.990	0.968	60%

Table 8: Compare the performance of bagging predictors on *G-mean* measure between the natural class distribution and the altered class distribution by using sampling techniques on four data-sets: diabetes, sick, WDBC and wbreastc.

8 Conclusions

This research investigates the performance of bagging predictors with respect to 12 different learning algorithms on 8 medical data-sets. We address the imbalance class distribution and unequal cost of mis-classification errors issues on medical data which may have high accuracy but poor performance on the TPR of minority class. We report the best performance prediction model for the natural class distribution on each individual medical data-set by comparing 12 single learners and 12 bagging predictors. In addition, we utilize sampling techniques to alter the class distribution at different imbalanced levels, and report the comparison of the bagging performance between the natural class distribution and the best achieved performance based on the *G-mean* measure at a certain level of class distribution. We note that by using sampling techniques to improve the performance of the bagging predictors, the level of the class distribution is mostly at 50% balanced level for three data-sets, breastc, heart-c, and statlogHeart; however, it is mostly at 40% for the diabetes, WDBC and heart-h data-set, and at 30% for the sick data-set. In addition, we also observe that the levels of class distribution for bagging NB to achieve the best performance on the *G-mean* measure are at 10% for the WDBC data-set and 20% for the sick data-set.

We investigated the effectiveness of bagging by using statistical tests. We also compared the performance of 12 bagging predictors on each of the medical data-sets; we observed that different bagging predictors behave differently for different medical data-sets. Bagging MLP performs well on most of these medical data-sets, except for the extremely imbalanced class distribution and high dimensional attributes large data-set 'sick'; Bagging NB has the best performance on 4 out of 8 medical data-sets but performs poorly on two medical data-sets: sick and WDBC; Bagging J48 and Dstump perform well on the extremely imbalanced and high dimensional large data-set, sick. The full comparison of the performance of bagging predictors would allow data mining practitioners to choose proper learners and to understand what to expect when using bagging predictors for medical imbalanced applications.

References:

Bauer, E. & Kohavi, R. (1999) An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning,* 36(1): 105-139.

Breiman, L. (1996) Bagging predictors. *Machine Learning,* 24(2): 123-140.

Bunkhumpornpat, C., Sinapiromsaran, K. & Lursinsap, C. (2009): Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. *Proc.* PAKDD 2009 Conference, 475-482,

Chawla, N. V. (2010) Data mining for imbalanced datasets: An overview. *Data Mining and Knowledge Discovery Handbook,* 875-886.

Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. (2002) Smote: Synthetic minority

over-sampling technique. *Journal of Artificial Intelligence Research,* 16(1): 321-357.

Chawla, N. V., Lazarevic, A., Hall, L. O. & Bowyer, K. W. (2003) Smoteboost: Improving prediction of the minority class in boosting. *Knowledge Discovery in Databases: PKDD 2003,* 107-119.

Demšar, J. (2006) Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research,* 7(1-30.

Dietterich, T. (2000) An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning,* 40(2): 139-157.

Fawcett, T. (2006) An introduction to roc analysis. *Pattern Recognition Letters,* 27(8): 861-874.

Guo, X., Yin, Y., Dong, C., Yang, G. & Zhou, G. (2008) On the class imbalance problem. *Fourth International Conference on Natural Computation.* IEEE.

Han, H., Wang, W. Y. & Mao, B. H. (2005) Borderline-smote: A new over-sampling method in imbalanced data sets learning. *Advances in Intelligent Computing,* 878-887.

He, H. & Garcia, A. E. (2009) Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering,* 21(9): 1263-1284.

Kim, M.-J. & Kang, D.-K. (2010) Ensemble with neural networks for bankruptcy prediction. *Expert Systems with Applications,* 37(4): 3373-3379.

Liang, G. & Zhang, C. (2011): An empirical evalutation of bagging with different learning algorithms on imbalanced data. *Proc.* 7th International Conference on Advanced Data Mining and Applications Conference, 17th-19th December, Beijing, China, Springer.

Liang, G., Zhu, X. & Zhang, C. (2011a): An empirical study of bagging predictors for different learning algorithms. *Proc.* 25th AAAI Conference on Artificial Intelligence, AAAI 2011 Conference, 7-11 August, San Francisco, USA, 1802-1803, AAAI Press.

Liang, G., Zhu, X. & Zhang, C. (2011b): An empirical study of bagging predictors for imbalanced data with different levels of class distribution. *Proc.* 24th Australasian Joint Conference on Artificial Intelligence, AI 2011 Conference, 5th-8th December, Perth, Australia, Springer.

Liu, W. & Chawla, S. (2011): Class confidence weighted knn algorithms for imbalanced data sets. *Proc.* PAKDD 2011 Conference, 6635:345-356, Springer Berlin / Heidelberg.

Lopes, L., Scalabrin, E. & Fernandes, P. (2008) An empirical study of combined classifiers for knowledge discovery on medical data bases. *Advanced Web and NetworkTechnologies, and Applications,* 110-121.

Maimon, O., Rokach, L. & Chawla, N. V. (2010) Data mining for imbalanced datasets: An overview. *Data mining and knowledge discovery handbook.* Springer US.

Maloof, M. (2003) Learning when data sets are imbalanced and when costs are unequal and unknown. *ICML-2003 workshop on learning from imbalanced data sets II.* Washington, DC.

Merz, C. & Murphy, P. (2006) *Uci repository of machine learning databases*.

Ng, W. & Dash, M. (2006) An evaluation of progressive sampling for imbalanced data sets. *Sixth IEEE International Conference on Data Mining Workshops.* IEEE.

Opitz, D. & Maclin, R. (1999) Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research,* 11(1): 169-198.

Phua, C., Alahakoon, D. & Lee, V. (2004) Minority report in fraud detection: Classification of skewed data. *ACM SIGKDD Explorations Newsletter,* 6(1): 50-59.

Provost, F. & Fawcett, T. (2001) Robust classification for imprecise environments. *Machine Learning,* 42(3): 203-231.

Provost, F., Fawcett, T. & Kohavi, R. (1998) The case against accuracy estimation for comparing induction algorithms. Citeseer.

Quinlan, J. (1996) Bagging, boosting, and c4.5. *Proceedings of the National Conference on Artificial Intelligence.*

Quinlan, J. R. (1986) Induction of decision trees. *Machine learning,* 1(1): 81-106.

Su, C. T. & Hsiao, Y. H. (2007) An evaluation of the robustness of mts for imbalanced data. *IEEE Transactions on Knowledge and Data Engineering,* 1321-1332.

Weiss, G. M. & Provost, F. (2003) Learning when training data are costly: The effect of class distribution on tree induction. *Journal of Artificial Intelligence Research,* 19(1): 315-354.

West, D., Dellana, S. & Qian, J. (2005) Neural network ensemble strategies for financial decision applications. *Computers & Operations Research,* 32(10): 2543-2559.

Witten, I. H. & Frank, E. (2005) *Data mining: Practical machine learning tools and techniques,* San Francisco, Morgan Kaufmann.

A Decision Tree-based Missing Value Imputation Technique for Data Pre-processing

Md. Geaur Rahman and Md. Zahidul Islam
Centre for Research in Complex System, School of Computing and Mathematics, Charles Sturt University, Locked Bag 588, Boorooma Street, Wagga Wagga, NSW 2678, Australia
{grahman,zislam}@csu.edu.au

Abstract

Data pre-processing plays a vital role in data mining for ensuring good quality of data. In general data pre-processing tasks include imputation of missing values, identification of outliers, smoothening out of noisy data and correction of inconsistent data. In this paper, we present an efficient missing value imputation technique called DMI, which makes use of a decision tree and expectation maximization (*EM*) algorithm. We argue that the correlations among attributes within a horizontal partition of a data set can be higher than the correlations over the whole data set. For some existing algorithms such as EM based imputation (EMI) accuracy of imputation is expected to be better for a data set having higher correlations than a data set having lower correlations. Therefore, our technique (DMI) applies EMI on various horizontal segments (of a data set) where correlations among attributes are high. We evaluate DMI on two publicly available natural data sets by comparing its performance with the performance of EMI. We use various patterns of missing values each having different missing ratios up to 10%. Several evaluation criteria such as coefficient of determination (R^2), Index of agreement (d_2) and root mean squared error (*RMSE*) are used. Our initial experimental results indicate that DMI performs significantly better than EMI.

Keywords: Data pre-processing, Data cleansing, Missing value imputation, Decision tree algorithm, EM algorithm.

1 Introduction

Organisations are extremely dependant nowadays on data collection, storage and analysis for various decision-making processes. Data are collected in various ways such as paper based and online surveys, interviews, and sensors (Chapman 2005, Cheng, Chen and Xie 2008, Apiletti, Bruno, Ficarra and Baralis 2006). For example, temperature, humidity, and wind speed data in a habitat monitoring system (HMS) are often acquired through different sensors. Due to various reasons including human error and misunderstanding, equipment malfunctioning, and introduction of noise during transformation and propagation data can often be lost or perturbed. For example, data in the HMS can be lost due to limited bandwidth problem in a wireless network, insufficient battery power of sensing devices, and other electro-mechanical problems of sensors.

Data anomalies and impurities can cause inefficient data analyses, inaccurate decisions and user inconveniencies. Careless use of erroneous data can be misleading and damaging making it useless for the users (Muller and Freytag 2003, Abbas and Aggarwal 2010, Han and Kamber 2006). For instance, poor quality data in genomic databases can have serious impact on end users. Errors in genome data can result in inaccurate outcomes from biological and pharmaceutical experiments costing billions of dollars for the pharmaceutical companies for developing only a few useful drugs. Hence, it is of great importance to have high quality data for safety critical data analyses (Muller, Naumann and Freytag 2003, Hensley 2002).

Therefore, a data pre-processing framework is crucial to deal with inaccurate and incomplete data for ensuring high data quality through effective data cleansing. One important task in data pre-processing is the imputation of missing values as accurately as possible. In the last few years a number of imputation methods have been proposed (Tseng, Wang and Lee 2003, Zhang, Qin, Zhu, Zhang and Zhang 2006, Junninen, Niska, Tuppurainen, Ruuskanen and Kolehminen 2004, Schneider 2001, Dempster, Laird and Rubin 1977, Dellaert 2002, Li, Zhang and Jiang 2005, Pyle 1999, Little and Rubin 1987).

Generally imputation performance heavily depends on the selection of a suitable technique (Zhang *et al*. 2006). Different imputation techniques perform well on different types of data sets and missing values. Existing imputation techniques therefore have room for further improvement.

We argue that since EMI algorithm relies on the correlations among the attributes while imputing missing values, it performs better on a data set having high correlations among the attributes. Correlations among the attributes are natural properties of a data set and they cannot be improved or modified. However, we realise that it is often possible to have horizontal segments within a data set where there are higher correlations than the correlations over the whole data set.

Record	Age (year)	Height (cm)
R1	4	70
R2	17	175
R3	12	150
R4	40	165
R5	8	120
R6	50	170
R7	30	165
R8	16	160
R9	60	168

(a) Correlations between age and height attributes over the whole sample data set are low.

	Record	Age (year)	Height (cm)
Partition -1	R1	4	70
	R5	8	120
	R3	12	150
	R8	16	160
	R2	17	175
Partition-2	R7	30	165
	R4	40	165
	R6	50	170
	R9	60	168

(b) Correlations between age and height attributes within a partition (Partition-1) are high.

Figure 1: A sample data set contains age and height information of different persons of a city in Australia.

This work was supported by the second author's CRiCS seed grant.

For example, consider a sample data set containing two attributes: age and height (Figure 1). For the whole data set (Figure 1a) the correlation among the attributes is very low since height of a person always does not increase with age. However, age and height are highly correlated for people younger than say 17 years. Therefore, if we partition the whole example data set into two segments where one segment contains records for all people who are younger than 17 years and the other segment contains the remaining records then we have a higher correlation among age and height in Partition 1 of Figure 1b. Therefore, the identification of the horizontal segments having high correlations and application of EMI algorithm within the segments is expected to produce a better imputation result.

In this paper, we propose a novel hybrid imputation technique called DMI that makes use of an existing decision tree algorithm such as *C4.5* (Quinlan 1993, Quinlan 1996, Kotsiantis 2007), and an Expectation Maximisation (EM) based imputation technique called EMI (Junninen *et al*. 2004, Schneider 2001, Dempster *et al*. 1977, Dellaert 2002) for data sets having both numerical and categorical attributes. In order to impute categorical missing values our technique uses a decision tree algorithm. However, for numerical missing values our technique with the help of a decision tree algorithm first identifies horizontal segments of records having high correlations among the attributes and then applies the EM algorithm within various horizontal segments.

We evaluate our technique (DMI) on two publicly available natural data sets by comparing its performance with the performance of EMI. We first prepare a data set having no natural missing values. We then generate data sets with artificial missing values (the original values for which are known to us) using various patterns of missing values such as simple, medium, complex and blended. In a simple pattern a record can have at most one missing value, whereas in a medium pattern if a record has any missing values then it has minimum 2 attributes with missing values and maximum 50% of the attributes with missing values. Similarly a record having missing values in a complex pattern has min 50% and maximum 80% attributes with missing values. In a blended pattern we have a mixture of records from all three other patterns. A blended pattern contains 25% records having missing values in simple pattern, 50% in medium pattern and 25% in complex pattern (Junninen *et al*. 2004).

In this paper, we also use different missing ratios ranging from 1% to 10% of total attribute values of a data set. Moreover, we use two missing categories/models called uniformly distributed (UD) missing values and overall missing values. In UD each attribute has the same number of missing values, whereas in an overall category an attribute can have higher number of missing values than the number of missing values in another attribute. Moreover, several well known evaluation criteria such as coefficient of determination (R^2), Index of agreement (d_2) and root mean squared error (*RMSE*) are used. Our experimental results indicate that DMI performs significantly better than EMI.

The organization of the paper is as follows. Section 2 presents a literature review. Our technique (DMI) is presented in Section 3. Section 4 presents experimental results and Section 5 gives concluding remarks.

2 Background Study

Many missing value imputation methods have been proposed recently (Tseng *et al*. 2003, Zhang *et al*. 2006, Junninen *et al*. 2004, Schneider 2001, Dempster *et al*. 1977, Dellaert 2002, Li *et al*. 2005, Pyle 1999, Little and Rubin 1987). However, most of the existing techniques are not suitable for a data set having both numerical and categorical attributes (Tseng *et al*. 2003).

A simple technique is to impute a missing value of an attribute by the mean of all values of the attribute (Schneider 2001). Several missing value imputation techniques such as Nearest Neighbour (NN), Linear Interpolation (LIN), Cubic Spline Interpolation, Regression based Expectation Maximization (REGEM) imputation, Self-organizing Map (SOP) and Multilayer Perceptron (MLP) have been proposed in the literature (Junninen *et al*. 2004). Hybrid models such as LIN+MLP and LIN+SOP have also been presented by Junninen *et al*. (2004) in order to improve the imputation accuracy. Their experimental results show a slight improvement due to the use of hybrid methods. Moreover, it was pointed out that a single imputation method has several limitations while a combination of a few imputation techniques can improve accuracy significantly.

Two interesting imputation methods, among many existing techniques, are EM algorithm (Dempster *et al*. 1977) and Kernel Function (Zhang *et al*. 2006). We discuss the techniques as follows.

For imputing numerical missing values of a data set EM algorithm relies on mean and covariance matrix of the data set. First the mean and covariance matrix are estimated from a data set having some missing values. Based on the mean and covariance matrix missing values are then imputed. For each record missing values (if any) are estimated as follows based on the relationship between attributes (Schneider 2001).

$$\boldsymbol{x}_m = \mu_m + (\boldsymbol{x}_a - \mu_a)\boldsymbol{B} + \boldsymbol{e} \qquad (1)$$

where $\boldsymbol{x}_m$ and $\boldsymbol{x}_a$ are vectors of missing values and available values of a record $\boldsymbol{x}$, respectively. Moreover, μ_m and μ_a are the mean vectors of missing values and available values, respectively. $\boldsymbol{B} = \boldsymbol{\Sigma}_{aa}^{-1}\boldsymbol{\Sigma}_{am}$ is a regression coefficient matrix, which is the product of the inverse of covariance matrix of available attribute values ($\boldsymbol{\Sigma}_{aa}^{-1}$), and cross covariance matrix of available and missing values ($\boldsymbol{\Sigma}_{am}$). Besides, $\boldsymbol{e}$ is a residual error with mean zero and unknown covariance matrix (Schneider 2001).

Using the imputed data set EM algorithm again estimates the mean and covariance matrix. The process of imputing missing values, and estimating mean and covariance matrix continues recursively until we get a mean and covariance matrix having difference with the previous mean and covariance matrix under user defined thresholds.

Kernel imputation method for numerical missing values was originally proposed by Wang and Rao (2002) and later on discussed by Zhang *et al*. (2006). In the technique the mean, covariance matrix and Gaussian kernel function of a data set are used to impute missing

values. Initially all missing values belonging to an attribute are imputed by the mean value of the attribute. Let d be the number of attributes, n be the number of records of a data set D and m be the total number of missing values in the whole data set. An attribute A_j $(1 \leq j \leq d)$ may have missing values for more than one record. All missing values belonging to an attribute $A_j, \forall j$ are first imputed using its average value.

At this stage the originally missing values are again imputed one by one considering only one imputed value as missing at a time. All other values including the remaining (*m-1*) imputed values are considered as non-missing. A missing value A_{ij} of an attribute A_j and record R_i is calculated as follows (Zhang et al. 2006).

$$\mathrm{A_{ij}} = \mathrm{m}\left(\mathrm{A_{i1}}, \mathrm{A_{i2}}, \dots, \mathrm{A_{id}}; \forall \mathrm{A_{ik}} \neq \mathrm{A_{ij}}\right) + \varepsilon_{\mathrm{i}} \qquad (2)$$

where he function $m\left(A_{i1}, A_{i2}, \dots, A_{id}; \forall A_{ik} \neq A_{ij}\right)$ is computed as follows.

$$\mathrm{m}\left(\mathrm{A_{i1}}, \mathrm{A_{i2}}, \dots, \mathrm{A_{id}}; \forall \mathrm{A_{ik}} \neq \mathrm{A_{ij}}\right) = \frac{\sum_{a=1}^{n} \delta_a \mathrm{A_{aj}} \prod_{s=1}^{d} \mathrm{K}\left(\frac{\mathrm{A_{is}} - \mathrm{A_{as}}}{h}\right)}{\sum_{a=1}^{n} \delta_a \prod_{s=1}^{d} \mathrm{K}\left(\frac{\mathrm{A_{is}} - \mathrm{A_{as}}}{h}\right) + \mathrm{n}^{-2}} \qquad (3)$$

where δ_a is either 0 or 1. If A_{aj} is missing then δ_a=*0*, and otherwise δ_a=*1*. $K(x) = \frac{1}{\sqrt{2\pi}} exp\left(\frac{x^{-2}}{2}\right)$ is the Gaussian kernel function, and h is typically considered as a constant value say 5 (Zhang et al. 2006).

3 Our Technique

We present a novel "Decision tree based Missing value Imputation technique" (DMI) which makes use of an EM algorithm and a decision tree (DT) algorithm. We first introduce the main ideas of the proposed technique as follows before we introduce it in detail.

EM based imputation techniques rely on the correlations among the attributes in a data set. We realise that the imputation accuracy is supposed to be high for a data set having high correlations among the attributes. Besides correlations among the attributes are natural properties of a data set and they cannot be improved or modified for the sake of achieving better imputation accuracy. However, we argue that correlations among the attributes can be higher within a horizontal partition of a data set than within a whole data set as shown in Figure 1.

Figure 5a gives an example of a decision tree obtained from a toy data set shown in Figure 4b. The nodes of the tree are shown as the rectangles and the leaves are shown as the ovals (Figure 5a). In each node a tree tests an attribute which we call the "test attribute" of the node. A decision tree divides the records of a data set into a number of leaves, where ideally all records belonging to a leaf have the same class value. However, in reality a leaf may contain records where majority of them have the same class value, but a few of them can have a different class value/s. Such a leaf is generally known as heterogeneous leaf. The class values in a heterogeneous leaf are considered to be similar to each other (Estivil-Castro and Brankovic 1999, Islam and Brankovic 2003). Moreover, the records belonging to a leaf are considered to be a cluster as they are shown to be similar to each other as well (Islam and Brankovic 2011, Islam 2008). The main justifications for considering the records similar to each other are that they share the same or similar class value/s, same values for all categorical test attributes for the leaf and similar values for all numerical test attributes for the leaf. For example, the records in the *Leaf 8* of Figure 5c have the same class value ("*7-10*") and the same test attribute value ("*a12*"). Another justification for considering the records to be similar was the fact that the entropy of the class values for the records within a leaf is the minimum (Islam and Brankovic 2011, Islam 2008).

Therefore, we argue that attribute correlations within the records belonging to a leaf are likely to be higher than attribute correlations within a whole data set. We test attribute correlations for the records within a leaf and for all records of "*Credit Approval*" data set (UCI Repository). Applying *C4.5* algorithm (Quinlan 1993, 1996) on Credit Approval data set we build a decision tree that has seven leaves. We then prepare seven correlation matrices, each for the records within a leaf. We also prepare a correlation matrix for all records. We observe that correlations among attributes within a leaf are generally higher than within the whole data set. Considering all seven leaves, on an average 66% of the correlations among the attributes have higher values within the records of a leaf than the correlation values that are calculated for all records. Figure 2 shows two correlation matrices for the six numerical attributes of Credit Approval data set. For six numerical attributes there are 15 correlations among the attributes. Only 3 out of 15 correlations have lower values in Figure 2b than the corresponding correlation values in Figure 2a.

1.00	0.23	0.41	0.19	-0.09	0.02
0.23	1.00	0.30	0.27	-0.21	0.12
0.41	0.30	1.00	0.33	-0.06	0.05
0.19	0.27	0.33	1.00	-0.12	0.06
-0.09	-0.21	-0.06	-0.12	1.00	0.07
0.02	0.12	0.05	0.06	0.07	1.00

(a) Full data set.

1.00	0.37	0.51	0.23	-0.13	0.10
0.37	1.00	0.39	0.30	-0.26	-0.01
0.51	0.39	1.00	0.32	-0.15	0.08
0.23	0.30	0.32	1.00	-0.17	0.04
-0.13	-0.26	-0.15	-0.17	1.00	0.08
0.10	-0.01	0.08	0.04	0.08	1.00

(b) Within a leaf.

Figure 2: Correlation matrix for the six numerical attributes of Credit Approval data set.

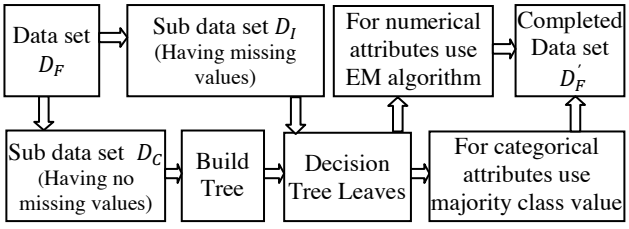


Figure 3: The overall block diagram of our DMI.

We first mention the main steps of DMI as follows and then explain each of them in detail.

Step-1: DMI divides a full data set (D_F) into two sub data sets D_C (having only records without missing values) and D_I (having only records with missing values).

Step-2: Build a set of decision trees on D_C considering the attributes, having missing values in D_I, as the class attributes.

Step-3: Assign each record of D_I to the leaf where it falls in for the tree that considers the attribute, which has a missing value for the record, as the class attribute. If the record has more than one attributes with missing values it will be assigned to more than one leaves.

Step-4: Impute numerical missing values using *EM* algorithm and categorical missing values using majority class values within the leaves.

Step-5: Combine records to form a completed data set ($D_F^{'}$) without any missing values.

The overall block diagram of DMI is shown in Figure 3.

Step-1: DMI divides a full data set (D_F) into two sub data sets D_C (having only records without missing values) and D_I (having only records with missing values).

To impute missing values in a data set, we first divide the data set D_F into two sub data sets D_C and D_I, where D_C contains records having no missing values and D_I contains records having missing values as shown in the DMI algorithm (Step-1 of Figure 8). For example, Figure 4 shows an example data set D_F, sub data set D_C, and sub data set D_I. The data set D_F has 9 records and 4 attributes out of which two are numerical and two are categorical. Two records (R3 and R5) have missing values for attributes C3 and C4, respectively. Therefore, we first move records R3 and R5 from D_F to sub data set D_I (Figure 4c) and the remaining records into sub data set D_C (Figure 4b).

Record	C1	C2	C3	C4
R1	a11	5	a31	10
R2	a13	7	a31	5
R3	a11	7	?	7
R4	a12	5	a31	10
R5	a13	3	a32	?
R6	a12	9	a31	10
R7	a11	5	a32	3
R8	a13	6	a32	7
R9	a12	6	a32	10

(a) A sample data set D_F.

Record	C1	C2	C3	C4
R1	a11	5	a31	10
R2	a13	7	a31	5
R4	a12	5	a31	10
R6	a12	9	a31	10
R7	a11	5	a32	3
R8	a13	6	a32	7
R9	a12	6	a32	10

(b) Sub data set D_C.

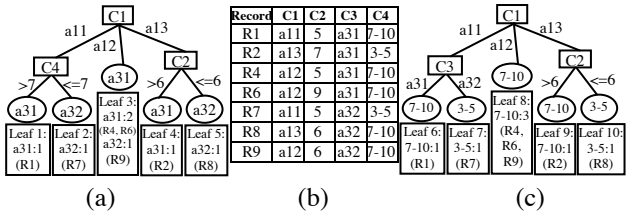
Record	C1	C2	C3	C4
R3	a11	7	?	7
R5	a13	3	a32	?

(c) Sub data set D_I.

Figure 4: A sample data set D_F with sub data sets D_C and D_I.

Step-2: Build a set of decision trees on D_C considering the attributes, having missing values in D_I, as the class attributes.

In this step we first identify attributes A_i $(1 \leq i \leq M)$ where M is the total number of attributes, in D_I, having missing values. We make a temporary copy of D_C as $D_C^{'}$. For each attribute A_i we build a tree using the *C4.5* decision tree algorithm for the sub data set $D_C^{'}$ considering A_i as class attribute. If A_i is a numerical attribute, we first generalize A_i of $D_C^{'}$ into N_C categories, where N_C is the squared root of the domain size of A_i (Step-2 of Figure 8).



(a) (b) (c)

Figure 5: (a) DT for sub data set shown in Figure 4b considering C3 as class attribute, (b) generalised sub data set of Figure 4b with numerical attribute C4 having 2 categories, and (c) DT for sub data set of Figure 5b considering C4 as class attribute.

We also use a boolean variable I_j with initial value "*FALSE*" for a data set d_j (Step 2, Figure 8) belonging to a leaf; for each leaf of all trees. The boolean variable is used in Step 4 to avoid multiple imputation on a data set and thereby reduce complexity.

In our example we build two DTs for attributes C3 and C4. Figure 5a shows a DT for data set of Figure 4b considering C3 as class attribute. It has five leaves. In Figure 5a, "*a31: 1*" for *Leaf 1* indicates the class value (*a31*) and number of records having the class value in the leaf. Moreover, (R1) indicates that record R1 falls in this leaf. Figure 5b shows the same sub data set as shown in Figure 4b. However, the C4 attribute in the data set shown in Figure 5b has been generalized into 2 categories. We then build a DT considering C4 as class attribute using the generalized sub data set (Figure 5c).

Step-3: Assign each record of D_I to the leaf where it falls in for the tree that considers the attribute, which has a missing value for the record, as the class attribute. If the record has more than one attributes with missing values it will be assigned to more than one leaves.

Each record r_k from the sub data set D_I has missing values. If r_k has a missing value for the attribute A_i then we use the tree T_i (that considers A_i as the class attribute) in order to indentify the leaf R_j, where r_k falls in. Note that r_k falls in R_j if the test attribute values of R_j match the attribute values of r_k. We add r_k in the data set d_j representing R_j *of* T_i (Step 3 of Figure 8). If r_k has more than one attributes with missing values then it is added in more than one data sets.

In this step, d_j represents a sub data set containing all records of the j^{th} leaf/logic rule. In our example, attribute C3 of record R3 is missing (Figure 4c). In order to assign the record R3 into a leaf, the tree shown in Figure 5a is the target tree since it considers C3 as the class attribute. Matching the test attribute values with the attribute values of R3 we assign R3 in *Leaf 2* of Figure 5a. The resultant leaf records are shown in Figure 6a. Similarly, we assign R5 into *Leaf 10* of Figure 5c which is shown in Figure 6b.

Record	C1	C2	C3	C4
R7	a11	5	a32	3
R3	a11	7	?	7

(a) Leaf 2.

Record	C1	C2	C3	C4
R8	a13	6	a32	7
R5	a13	3	a32	?

(b) Leaf 10.

Figure 6: Assign records of D_I into leaves.

Step-4: Impute numerical missing values using *EM* algorithm and categorical missing values using majority class values within the leaves.

We impute the missing values for all records in D_I one by one. For a record r_k we identify an attribute A_i having a missing value. We also identify the data set d_j where r_k has been added (in Step 3) for imputation of missing value in the attribute A_i. If d_j has not been imputed before (i.e. if I_j is *FALSE*) then we apply either *EM* algorithm or DT based imputation depending on the type (numerical or categorical) of the attribute A_i and thereby impute the values of A_i in d_j. Finally we update the attribute value for A_i of r_k in D_I from the imputed attribute value for A_i of r_k in d_j. We continue the process for all r_k in D_I and thereby impute all missing values for all r_k in D_I (Step-4 of Figure 8).

In our example, we impute the attribute C3 of record R3 (Figure 6a) using the class value "*a32*" of *Leaf 2* (Figure 5a). We apply EM algorithm on the data set belonging to *Leaf 10* (Figure 5c and Figure 6b) to impute the value of attribute C4 of record R5. Figure 7 shows the data sets after imputation.

Record	C1	C2	C3	C4
R3	a11	7	a32	7
R5	a13	3	a32	7

(a) Imputed data set D_I.

Record	C1	C2	C3	C4
R1	a11	5	a31	10
R2	a13	7	a31	5
R3	a11	7	**a32**	7
R4	a12	5	a31	10
R5	a13	3	a32	**7**
R6	a12	9	a31	10
R7	a11	5	a32	3
R8	a13	6	a32	7
R9	a12	6	a32	10

(b) Completed data set $D_F^{'}$.

Figure 7: Impute missing values and combine records to form a completed data set $D_F^{'}$.

There are two problematic cases where EM algorithm needs to be used carefully in order to get a proper imputation result. First, EM algorithm does not work if all records have the same value for a numerical attribute. Second, EM algorithm is also not useful when we have all numerical values missing in a record. DMI initially ignores the attribute having same value for all records. It also ignores the records having all numerical values missing. DMI then imputes all others values as usual. Finally, it uses the mean value of an attribute to impute a numerical missing value for a record having all numerical values missing. It also uses the mean value of an attribute to impute a missing value belonging to an attribute having the same value for all records.

Step-5: Combine records to form a completed data set ($D_F^{'}$).

We finally combine D_C and D_I in order to form $D_F^{'}$ which is the imputed data set. In our example, we integrate sub data sets of Figure 4b and Figure 7a to represent the complete data set $D_F^{'}$ shown in Figure 7b.

4 Experimental Result

We implement both our novel technique DMI and an existing technique EMI (Junninen *et al*. 2004), which is an EM based imputation over all records of a data set. We apply the techniques on two real life data sets, namely Adult and Credit Approval data set. The data sets are publicly available in UCI Machine Learning Repository (UCI Repository).

The Adult data set contains census information of United States and has 32561 records with 15 attributes including the natural class attribute. There are all together 9 categorical attributes, and 6 numerical attributes. We remove all records having missing values, and thereby work on 30162 records with no missing values. On the other hand, the Credit Approval data set contains information about credit card applications and has 690 records with 16 attributes including the class attribute. There are 10 categorical attributes, and 6 numerical attributes. There are also a number of records having missing values. We remove all records with missing values, and thereby produce 653 records with no missing values.

Note that the missing values that naturally exist in the data sets are first removed to prepare a data set without any missing values. We then artificially create missing values which are imputed by the different techniques. Since the original values of the artificially created missing data are known to us we can easily evaluate the efficiencies of the imputation techniques.

Algorithm: DMI
Input: A data set D_F having missing values
Output: A data set $D_F^{'}$ with all missing values imputed

Method:
$D_C \leftarrow \emptyset$. //Sub data set having records without missing values
$D_I \leftarrow \emptyset$. //Sub data set having records with missing values
L←0. //Total no. of leaves of all trees
Step-1: Divide data set D_F as follows
$D_I \leftarrow$ all missing-valued records of D_F.
$D_C \leftarrow D_F - D_I$.
Step-2: Find $A_i, i = 1, \ldots, M$, where M is the total no. of attributes, in D_I, having missing values.
For all attributes $A_i \in A = \{A_1, A_2, \ldots A_M\}$ do
  Set $D_C^{'} \leftarrow D_C$.
  If A_i is numerical then
    Find no. of categories $N_C \leftarrow \sqrt{|A_i|}$ where $|A_i|$ is the domain size of A_i.
    Generalize values of A_i for all records in $D_C^{'}$ into N_C categories.
  End if
  Call *C4.5* to build a tree T_i from $D_C^{'}$ considering A_i as class attribute.
  Set $S_i \leftarrow$ No. of leaves of T_i.
  For $j = L$ to $L + S_i$ do
    Define logic rule R_j from T_i.
    Generate data set d_j having all records belonging to R_j.
    Set $I_j \leftarrow FALSE$.
  End for
  $L \leftarrow L + S_i$.
End for
Step-3: For each record $r_k \in D_I$ do
  For attributes $A_i \in A$ do
    If A_i is numerical and missing then
      Find the data set d_j for the logic rule R_j belonging to the tree T_i where the record r_k falls in R_j.
      $d_j \leftarrow d_j \cup r_k$.
    End if
  End for
End for
Step-4: For each record $r_k \in D_I$ do
  For attributes $A_i \in A$ do
    If A_i is missing then
      Find d_j.
      If I_j *is* $FALSE$ then
        If A_i is numerical then
          Impute A_i using EM algorithm on d_j.
        Else if A_i is categorical then
          Impute A_i from R_j considering the majority class value of d_j as the imputed value.
        End if
        $I_j \leftarrow TRUE$.
      End if.
      Update A_i of r_k in D_I from A_i of r_k in d_j.
    End if
  End for
End for
Step-5: Completed data set $D_F^{'} \leftarrow D_C \cup D_I$.
  Return $D_F^{'}$.

Figure 8: The DMI missing values imputation algorithm.

An imputation performance does not only depend on the amount of missing data, but also depends on the characteristics of missing data patterns (Junninen *et al.* 2004). For example, in one scenario (pattern) we may have a data set where a record has at most one missing value, and in another scenario we may have records with multiple missing values. Note that both data sets may have the same number of total missing values.

Typically the probability of a value being missing does not depend on the missing value itself (Rubin 1976, Schneider 2001) and hence missing values often can have a random nature which can be difficult to formulate. Therefore, we use various patterns of missing values such as simple, medium, complex and blended. In a simple pattern a record can have at most one missing value, whereas in a medium pattern a record can have missing values for number of attributes ranging from 2 to 50% of the total number of attributes. Similarly a record having missing values in a complex pattern has minimum 50% and maximum 80% attributes having missing values. In a blended pattern we have a mixture of records from all three other patterns. A blended pattern contains 25% records having missing values in simple pattern, 50% in medium pattern and 25% in complex pattern. Blended pattern simulates a natural scenario where we may expect a combination of all three missing patterns. For each of the patterns, we use different missing ratios (1%, 3%, 5% and 10% of total attribute values of a data set) as shown in Table. 1.

We also use two types of missing models namely Overall and Uniformly Distributed (UD). In the overall distribution missing values are not equally spread out among the attributes, and in the worst case scenario all missing values can belong to a single attribute. However, in the UD model each attribute has equal number of missing values.

For each combination of Missing Pattern, Missing Ratio and Missing Model we create 10 data sets with missing values. For example, for the combination having "simple" missing pattern, "1%" missing values and "overall" model (see Table 2) we generate 10 data sets with missing values. We therefore create all together 320 data sets (32 combinations x 10 data sets/combination) for each natural data set namely Adult and Credit Approval.

Imputation accuracy is evaluated using a few well known evaluation criteria namely co-efficient of determination (R^2), Index of agreement (d_2) and root mean squared error (RMSE).

Missing Pattern	No. of attributes having missing values		Missing Ratios	Missing Model	No. of simulations for each pattern combination
	Min	Max			
Simple	1	1	1%, 3%, 5% and 10%	Overall and Uniformly distributed (UD)	10
Medium	2	50%			
Complex	50%	80%			
Blended	Simple 25%, Medium 50% and Complex 25%				

Table 1: Settings of missing data simulation.

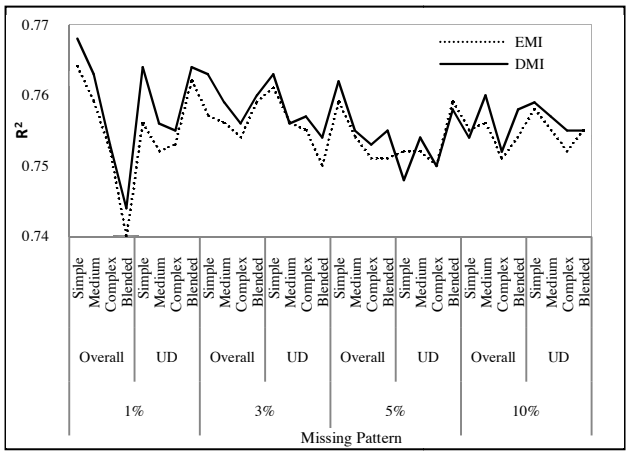
We now define the evaluation criteria briefly. Let *N* be the number of artificially created missing values, O_i be the actual value of the *i*th artificially created missing value ($1 \leq i \leq N$), P_i be the imputed value of the *i*th missing value, $\bar{O}$ be the average of actual values $O_i, \forall i \in N$. Let $\bar{P}$ be the average of the imputed values, σ_O be the standard deviation of the actual values and σ_P be the standard deviation of the imputed values.

Missing Pattern			R^2 (Higher value is better)		d_2 (Higher value is better)		RMSE (Lower value is better)	
			EMI	DMI	EMI	DMI	EMI	DMI
1%	Overall	Simple	0.759	**0.763**	0.893	**0.927**	0.123	**0.122**
		Medium	0.754	**0.758**	0.888	**0.923**	0.126	**0.125**
		Complex	0.747	**0.748**	0.886	**0.922**	0.126	0.126
		Blended	0.735	**0.739**	0.880	**0.919**	0.129	**0.128**
	UD	Simple	0.751	**0.759**	0.888	**0.925**	0.126	**0.124**
		Medium	0.747	**0.751**	0.886	**0.922**	0.127	**0.126**
		Complex	0.748	**0.750**	0.884	**0.923**	0.126	**0.125**
		Blended	0.757	**0.759**	0.893	**0.924**	0.123	0.123
3%	Overall	Simple	0.752	**0.758**	0.886	**0.923**	0.126	**0.125**
		Medium	0.751	**0.754**	0.888	**0.922**	0.125	**0.124**
		Complex	0.749	**0.751**	0.886	**0.922**	0.126	0.126
		Blended	0.754	**0.755**	0.888	**0.922**	0.125	0.125
	UD	Simple	0.756	**0.758**	0.889	**0.923**	0.125	**0.124**
		Medium	0.751	0.751	0.886	**0.919**	**0.126**	0.127
		Complex	0.750	**0.752**	0.888	**0.922**	0.126	0.126
		Blended	0.745	**0.749**	0.885	**0.920**	0.127	**0.126**
5%	Overall	Simple	0.754	**0.757**	0.886	**0.921**	0.126	**0.125**
		Medium	0.749	**0.750**	0.886	**0.919**	**0.126**	0.127
		Complex	0.746	**0.748**	0.885	**0.921**	0.127	**0.126**
		Blended	0.746	**0.750**	0.884	**0.919**	0.128	**0.127**
	UD	Simple	**0.747**	0.743	0.885	**0.915**	**0.127**	0.129
		Medium	0.747	**0.749**	0.884	**0.919**	0.127	0.127
		Complex	0.745	0.745	0.885	**0.918**	**0.127**	0.128
		Blended	**0.754**	0.753	0.888	**0.918**	**0.125**	0.127
10%	Overall	Simple	**0.750**	0.749	0.886	**0.915**	**0.127**	0.129
		Medium	0.751	**0.755**	0.885	**0.920**	0.127	**0.126**
		Complex	0.746	**0.747**	0.884	**0.919**	0.127	0.127
		Blended	0.749	**0.753**	0.884	**0.920**	0.127	**0.126**
	UD	Simple	0.753	**0.754**	0.888	**0.917**	**0.125**	0.127
		Medium	0.750	**0.752**	0.885	**0.918**	0.127	**0.126**
		Complex	0.747	**0.750**	0.886	**0.921**	0.127	**0.126**
		Blended	0.750	0.750	0.885	**0.917**	0.127	0.127

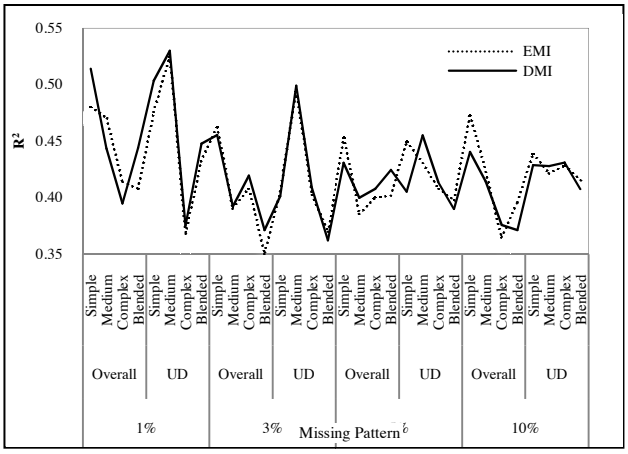
Table 2: Average performance of DMI and EMI on Adult data set.

Missing Pattern			R^2 (Higher value is better)		d_2 (Higher value is better)		RMSE (Lower value is better)	
			EMI	DMI	EMI	DMI	EMI	DMI
1%	Overall	Simple	0.480	**0.514**	0.704	**0.796**	0.111	**0.109**
		Medium	**0.470**	0.444	0.707	**0.773**	**0.109**	0.119
		Complex	**0.414**	0.395	0.668	**0.723**	**0.116**	0.118
		Blended	0.408	**0.444**	0.687	**0.766**	0.106	**0.103**
	UD	Simple	0.475	**0.504**	0.721	**0.799**	0.105	**0.102**
		Medium	0.524	**0.530**	0.739	**0.793**	0.113	**0.112**
		Complex	0.368	**0.375**	0.648	**0.723**	0.120	**0.119**
		Blended	0.432	**0.448**	0.697	**0.755**	0.119	**0.118**
3%	Overall	Simple	**0.462**	0.455	0.707	**0.779**	**0.114**	0.116
		Medium	0.390	**0.392**	0.665	**0.753**	0.118	0.118
		Complex	0.408	**0.420**	0.676	**0.757**	0.123	**0.122**
		Blended	0.351	**0.372**	0.642	**0.729**	**0.125**	0.126
	UD	Simple	**0.405**	0.402	0.685	**0.761**	**0.116**	0.119
		Medium	0.494	**0.499**	0.738	**0.804**	0.112	0.112
		Complex	0.402	**0.408**	0.678	**0.748**	0.121	0.121
		Blended	**0.370**	0.363	0.657	**0.723**	**0.131**	0.133
5%	Overall	Simple	**0.453**	0.431	0.718	**0.783**	**0.113**	0.118
		Medium	0.385	**0.400**	0.664	**0.752**	0.129	0.129
		Complex	0.400	**0.408**	0.668	**0.747**	0.124	**0.123**
		Blended	0.401	**0.425**	0.685	**0.768**	0.120	**0.117**
	UD	Simple	**0.449**	0.406	0.712	**0.761**	**0.118**	0.125
		Medium	0.430	**0.455**	0.697	**0.779**	0.120	**0.117**
		Complex	0.407	**0.414**	0.674	**0.750**	0.123	**0.122**
		Blended	**0.398**	0.390	0.671	**0.743**	**0.123**	0.125
10%	Overall	Simple	**0.472**	0.440	0.714	**0.779**	**0.118**	0.127
		Medium	**0.422**	0.415	0.691	**0.764**	**0.118**	0.120
		Complex	0.365	**0.376**	0.648	**0.736**	0.125	**0.124**
		Blended	**0.396**	0.372	0.674	**0.740**	**0.123**	0.128
	UD	Simple	**0.438**	0.429	0.703	**0.774**	**0.117**	0.120
		Medium	0.421	**0.428**	0.690	**0.769**	0.124	**0.123**
		Complex	0.428	**0.431**	0.688	**0.763**	0.120	0.120
		Blended	**0.415**	0.408	0.687	**0.759**	**0.119**	0.121

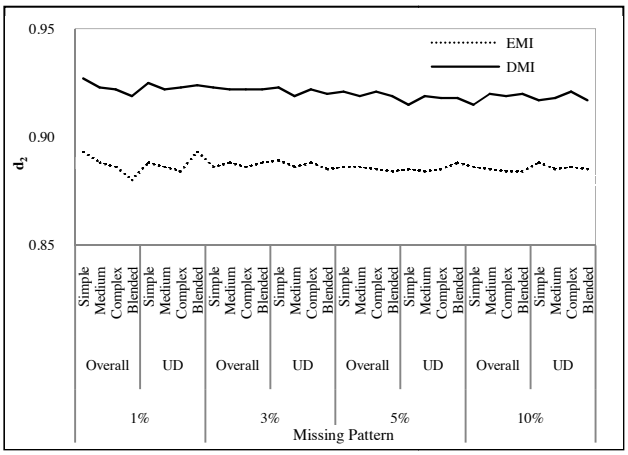
Table 3: Average performance of DMI and EMI on Credit Approval data set.



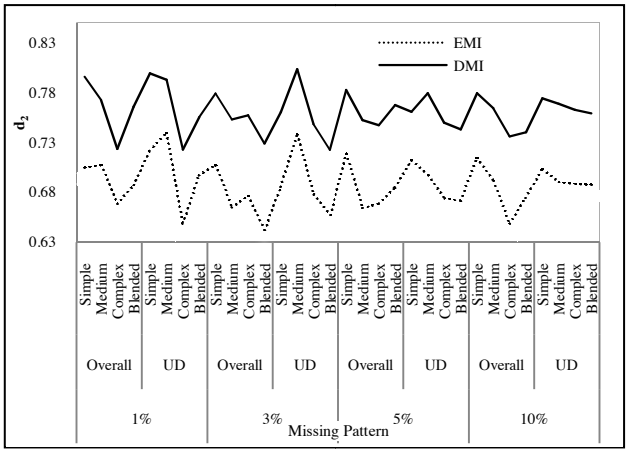
Graph 1: Average performance based on R^2 of DMI and EMI on Adult data set.



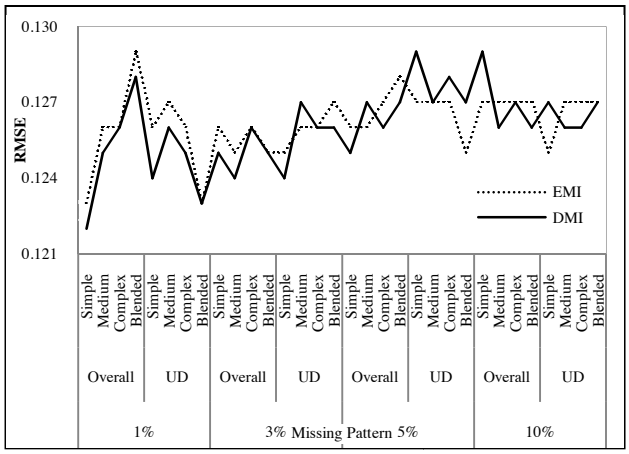
Graph 4: Average performance based on R^2 of DMI and EMI on Credit Approval data set.



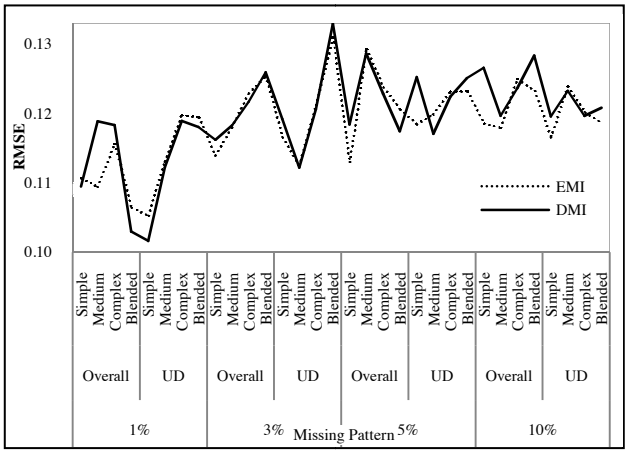
Graph 2: Average performance based on d_2 of DMI and EMI on Adult data set.



Graph 5: Average performance based on d_2 of DMI and EMI on Credit Approval data set.



Graph 3: Average performance based on RMSE of DMI and EMI on Adult data set.



Graph 6: Average performance based on RMSE of DMI and EMI on Credit Approval data set.

The most commonly used imputation performance indicator is the coefficient of determination (R^2), which determines the degree of correlations between actual and imputed values. It varies between the range 0 and 1, where 1 indicates a perfect fit.

$$R^2 = \left[\frac{\frac{1}{N}\sum_{i=1}^{N}[(P_i-\bar{P})(O_i-\bar{O})]}{\sigma_P \sigma_O}\right]^2 \quad (4)$$

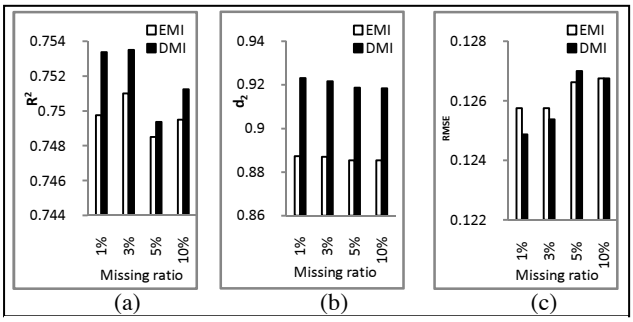
The index of agreement (Willmott 1982) determines the degree of agreement between actual and imputed values. Its value ranges between 0 and 1. Higher value indicates a better fit.

$$d = 1 - \left[\frac{\sum_{i=1}^{N}(P_i-O_i)^k}{\sum_{i=1}^{N}(|P_i-\bar{O}|+|O_i-\bar{O}|)^k}\right] \quad (5)$$

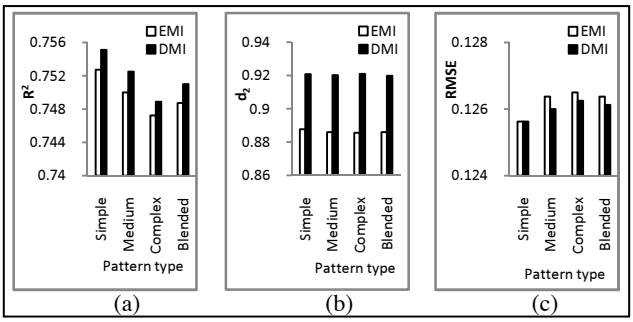
where k is either 1 or 2. The index (d_2) has been used throughout this experiment with k equal to 2.

The root mean squared error (*RMSE*) aims to explore the average difference of actual values with the imputed values as shown in Equation 6. Its value ranges from 0 to ∞, where a lower value indicates a better matching.

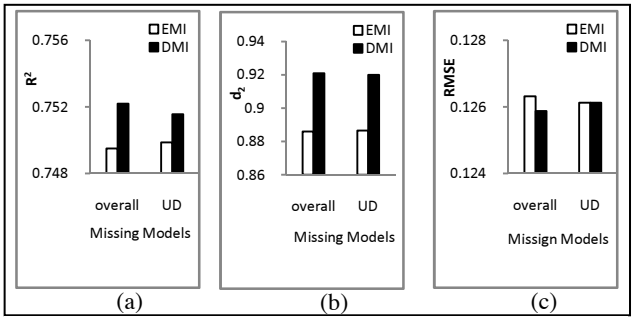
$$RMSE = \left(\frac{1}{N}\sum_{i=1}^{N}[P_i - O_i]^2\right)^{\frac{1}{2}} \quad (6)$$



Graph 7: Average performances of DMI and EMI based on missing ratios for Adult data set.



Graph 8: Average performances of DMI and EMI based on pattern types for Adult data set.



Graph 9: Average performances of DMI and EMI based on missing models for Adult data set.

The residual error $\boldsymbol{e}$ in Equation 1 is calculated in this experiment as follows (Muralidhar, Parsa and Sarathy 1999).

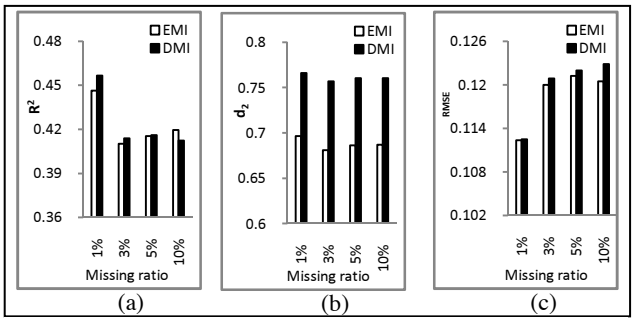
$$e = [\mu_0 + H.Z^T]^T \quad (7)$$

where μ_0 is a mean vector having zero values, H is a cholesky decomposition of covariance matrix C (Equation 8), and Z is a vector having Gaussian random values.C is calculated as follows (Schneider 2001).

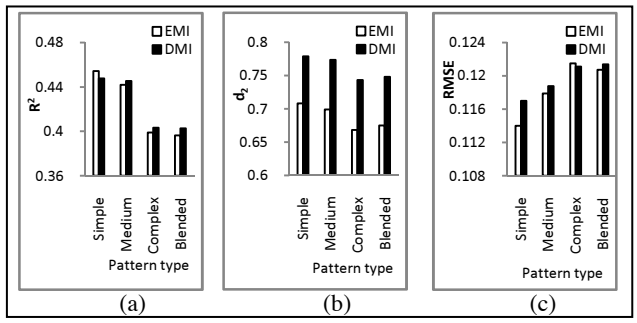
$$C = \Sigma_{mm} - \Sigma_{ma}\Sigma_{aa}^{-1}\Sigma_{am} \quad (8)$$

where Σ_{mm} is the covariance matrix of missing variables, Σ_{ma} is a covariance of missing and available variables, and Σ_{aa}^{-1} is the inverse of a covariance of available variables.

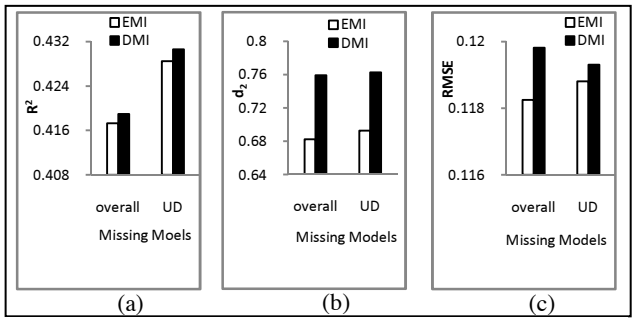
Moreover, in order to evaluate the changes in covariance matrices of the consecutive iterations (in EMI and DMI) we calculate the sample covariance matrices from the data sets generated by the iterations. The termination threshold used in EM algorithm is considered to be 10^{-10} in the experiments. If the difference between the determinants of two consecutive covariance matrices is below the threshold, and the difference between the average-values of two consecutive mean vectors are below the threshold then the termination condition used in the experiments is considered to be satisfied.



Graph 10: Average performances of DMI and EMI based on missing ratios for Credit Approval data set.



Graph 11: Average performances of DMI and EMI based on pattern types for Credit Approval data set.



Graph 12: Average performances of DMI and EMI based on missing models for Credit Approval data set.

We present performance of DMI and EMI based on R^2, d_2 and RMSE for both Adult and Credit approval data set in Table 2 and Table 3, respectively.

The average values of the performance indicators on 10 data sets having missing values for each combination of missing pattern, missing ratio and missing model is presented in the tables. For example, there are 10 data sets having missing values with the combination Com_1 of "Simple" missing pattern, "1%" missing ratio and "Overall" missing model. The average of R^2 for the data sets having Com_1 is 0.763 for DMI as reported in Table 2. Bold values in the tables indicate better results between the two techniques. DMI performs significantly better than EMI on both data sets.

We now discuss the experimental results on Adult data set presented in Table 2. For R^2 DMI performs better than EMI in 26 out of 32 missing pattern combinations. In 3 occasions it has the same performance as the performance of EMI (Table 2, and Graph 1). In terms of d_2 DMI performs better than EMI for all 32 missing pattern combinations (Graph 2). Moreover, for RMSE it performs better in 17 combinations and similar in 8 combinations (Graph 3).

Similarly for Credit Approval data set (Table 3) DMI performs better than EMI for 19 out of 32 combinations in terms of R^2(Graph 4). DMI out performs EMI for all 32 combinations based on d_2 (Graph 5). However, for RMSE it performs better in 13 combinations and similar in 5 combinations (Graph 6).

The average performances (for all 32 missing pattern combinations) of DMI and EMI on both data sets are shown in Graph 7 and Graph 10.

It is inspiring to observe that DMI performs significantly better than EMI on Adult data set. Although DMI achieves overall better result than EMI on Credit Approval data set as well, it does not perform as good as its performance on Adult data set. One possible explanation can be the size (30162 records) of Adult data set, which is larger than the size (653 records) of Credit Approval data set. Since in DMI we apply EM based imputation on the records belonging to a leaf we often may end up having insufficient number of records for getting a good result from the EM algorithm. Therefore, it is understandable that DMI's performance on a larger data set is better than its performance on a smaller data set.

Graph 8 and Graph 11 show a performance comparison of the techniques on both data sets based on four missing patterns namely simple, medium, complex and blended. Both graphs show that the overall performances of both techniques are better for simple pattern type than blended pattern type. We also present a comparison of performances based on missing models in Graph 9 and Graph 12. DMI performs clearly better than EMI in all different analyses, especially for Adult data set.

We now assign a score to an algorithm for each combination of Table 1 and Table 2. If an algorithm performs better than the other then we assign 1, otherwise 0. If the performances of both algorithms are equal then both of them score 0.5. The overall scores of DMI and EMI are presented in Table 4. For both Adult data set and Credit Approval data set in terms of d_2DMI scores 32 while EMI scores zero.

Data set	Evaluation Criteria	EMI	DMI
Adult	R^2	4.5	**27.5**
	d_2	0	**32**
	RMSE	11	**21**
Credit Approval	R^2	13	**19**
	d_2	0	**32**
	RMSE	**16.5**	15.5
	Total	45	**147**

Table 4: Overall performance (score comparison).

5 Conclusion

In this paper we present a novel missing value imputation technique called DMI that makes use of an entropy based decision tree algorithm and expectation maximisation based imputation technique. The main contributions of the paper are as follows. We realise that the EM algorithm produces better imputation result on data sets having higher correlations among the attributes. Besides correlations among the attributes are natural properties of a data set. A data set should not be modified in order to improve the correlations for the sake of achieving better imputation accuracy. However, correlations among the attributes can be higher within a horizontal partition of a data set than within the whole data set. We propose the use of an entropy based decision tree algorithm in order to identify the horizontal segments having higher correlations.

On each horizontal segment DMI algorithm applies an existing imputation technique, which heavily relies on the correlations among the attributes, in order to take advantage of higher correlations within the segments. Thus, DMI is expected to impute numerical missing values with higher accuracy. Moreover, for categorical missing values DMI applies a decision tree based imputation approach within each horizontal segment separately. It applies the decision tree algorithm to build a tree for each attribute having missing value/s. Therefore, it uses an attribute-specific horizontal segments that results in better imputation accuracy. DMI is capable of imputing both numerical and categorical missing values.

DMI also handles the two problematic cases where EMI algorithm may not provide reasonable results. The cases are, all records having the same value for a numerical attribute and all numerical values are missing in a record.

We evaluate DMI on two publicly available natural data sets by comparing its performance with the performance of EMI. We use various missing patterns such as simple, medium, complex and blended each having different missing ratios ranging from 1% to 10%. We also use two missing models/categories namely Uniformly Distributed (all attributes have equal number of missing values) and Overall. Several evaluation criteria such as coefficient of determination (R^2), Index of agreement (d_2) and root mean squared error (*RMSE*) are also used. Our initial experimental results indicate that DMI performs significantly better than EMI.

DMI performs clearly better on Adult data set for all evaluation criteria, whereas for Credit Approval data set DMI performs better for R^2 and d_2 (Table 4). Moreover, we compare DMI with EMI based on missing ratios (Graph 7 and Graph 10), missing patterns (Graph 8 and Graph 11) and missing models (Graph 9 and Graph 12). For Adult data set DMI performs better than EMI for all comparisons. It also performs better than EMI in most cases for Credit Approval data set.

Although DMI performs significantly better than EMI on both data sets, its performance on Adult data set is clearly better than its performance on Credit Approval data set. It is worth mentioning that Adult data set has 32561 records which is approximately 47 times larger than the Credit Approval (690 records) data set. It appears that DMI performs better on larger data sets. Since DMI applies EM based imputation on the records belonging to a leaf separately, for a small data set we often may end up having insufficient number of records for getting a good result from the EM algorithm. However, in our experiments DMI still performs better than EMI even on Credit Approval data set in most of the cases.

Our future work plans include further improvement of the DMI algorithm, and extensive experiments to compare DMI with many other existing techniques such as kernel function, regression analysis and a number of existing hybrid models.

6 References

Abbas, Q. S. and Aggarwal, A. (2010): Development of a Structured Framework To Achieve Quality Data. *International Journal of Advanced Engineering & Application*, 193-196.

Apiletti, D., Bruno, G., Ficarra, E. and Baralis, E. (2006): Data Cleaning and Semantic Improvement in Biological Databases. *Journal of Integrative Bioinformatics*.

Chapman, A. D. (2005): Principles of data quality, version 1.0. Report for the global biodiversity information facility, Copenhagen.

Cheng, R., Chen, J. and Xie, X. (2008): Cleaning Uncertain Data with Quality Guarantees. *VLDB Endowment*, ACM.

Data Cleansing: Data Cleansing, Navindex Pty Ltd, http://www.navindex.com.au/index.php?option=com_content&view=article&id=38:data-cleansing&catid=32:data-analysis&Itemid=53. Accessed 15 June 2011.

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977): Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*. Series B (Methodological), **39(1)**:1-38.

Dellaert, F. (2002): The Expectation Maximization Algorithm. College of Computing, Georgia Institute of Technology, Technical Report number GIT-GVU-02-20.

Estivil-Castro, V. and Brankovic, L. (1999): Data Swapping: Balancing Privacy Against Precision in Mining for Logic Rules. *Data Warehousing and Knowledge Discovery* (DaWaK'99), LNCS, **1676**: 389-398.

Han, J. and Kamber, M. (2006): *Data mining concepts and techniques*. Morgan Kaufmann Publishers, San Mateo, California, USA.

Hensley, S. (2002): Death of Pfizer's 'Youth Pill' Illustrates Drugmakers Woes. *The Wall Street Journal online*.

Islam, M. Z. and Brankovic, L. (2003): Noise Addition for Protecting Privacy in Data Mining. *Proceedings of the 6th Engineering Mathematics and Applications Conference (EMAC 2003)*.

Islam, M. Z. and Brankovic, L. (2005): DETECTIVE: A Decision Tree Based Categorical Value Clustering and Perturbation Technique in Privacy Preserving Data Mining. *Proc. of the 3rd International IEEE Conference on Industrial Informatics*.

Islam, M. Z. (2008): Privacy Preservation in Data Mining through Noise Addition. PhD thesis. The University of Newcastle, Australia.

Islam, M. Z. (2010): Explore: A Novel Decision Tree Classification Algorithm. *Proc. of 27th International Information Systems Conference*, British National Conference on Databases (BNCOD 2010), LNCS, Dundee, Scotland, vol. **6121**, Springer press.

Islam, M. Z. and Brankovic, L. (2011): Privacy Preserving Data Mining: A Noise Addition Framework Using a Novel Clustering Technique. *Knowledge-Based Systems*, in press.

Junninen, H., Niska, H., Tuppurainen, K., Ruuskanen, J. and Kolehminen, M. (2004): Methods for imputation of missing values in air quality data sets. *Atmospheric Environment*, Elsevier, **38**: 2895-2807.

Kotsiantis, S. B. (2007): Supervised Machine Learning: A Review of Classification Techniques. *Informatica,* **31**:249-268.

Li, H., Zhang, K. and Jiang, T. (2005): The Regularized EM Algorithm. *American Association for Artificial Intelligence*.

Little, R. J. A., and Rubin, D. B. (1987): *Statistical Analysis with Missing Data*. John Wileyand Sons Publishers, New York.

Muralidhar, k., Parsa, R. and Sarathy, R. (1999): A General Additive Data Perturbation Method for Database Security. *Management Science*, **45**: 1399 – 1415.

Muller, H., Naumann, F. and Freytag, J. (2003): Data Quality in Genome Databases. Humboldt-Universitt zu Berlin, Institut fr Informatik, Berlin.

Muller, H. and Freytag, J. (2003): Problems, methods and challenges in comprehensive data cleansing. Humboldt-Universitt zu Berlin, Institut fr Informatik, Berlin.

Pyle, D. (1999): *Data Preparation for Data Mining*. Morgan Kaufmann Publishers, San Mateo, California, USA.

Quinlan, J. R. (1986): Induction of Decision Trees. *Maching Learning*, **1**: 81-106.

Quinlan, J. R. (1993): *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, California, USA.

Quinlan, J. R. (1996): Improved Use of Continuous Attributes in C4.5. *Artificial Intelligence Research*, **4**:77-90.

Rubin, D. B. (1976): Inference and missing data. *Biometrika,* **65**:581-592.

Schneider, T. (2001): Analysis of incomplete climate data: Estimation of mean values and covariance matrices and imputation of missing values. *Journal of Climate,* American Meteorological Society, 853–871.

Tseng, S. M., Wang, K. H. and Lee, C. I. (2003): A pre-processing method to deal with missing values by integrating clustering and regression techniques. *Applied Artificial Intelligence*, **17**:535–544.

UCI Repository: UCI Machine Learning Repository. http://archive.ics.uci.edu/ml/. Accessed 15 June 2011.

Wang, Q. and Rao, J. N. K.(2002): Emperical likelihood-based inference in linear models with missing data. Journal of Statistics, **29**:563-576.

Willmott, C. J. (1982): Some comments on the evaluation of model performance. *Bulletin of the American Meteorological Society* **63**:1309-1313.

Zhang, C., Qin, Y., Zhu, X., Zhang, J. and Zhang, S. (2006): Clustering-based Missing Value Imputation for Data Preprocessing. *Industrial Informatics IEEE International Conference*, Singapore, 1081-1086.

Feature Selection using Misclassification Counts

Adil Bagirov[1] **Andrew Yatsko**[1] **Andrew Stranieri**[1] **Herbert Jelinek**[1,2]

[1] School of Science, Information Technology and Engineering,
University of Ballarat,
Ballarat, Victoria, 3353, Australia
E-mail: a.stranieri@ballarat.edu.au, a.bagirov@ballarat.edu.au,
andrewyatsko@students.ballarat.edu.au

[2] School of Community Health
Charles Sturt University,
Albury, New South Wales, 2640, Australia
Email: hjelinek@csu.edu.au

Abstract

Dimensionality reduction of the problem space through detection and removal of variables, contributing little or not at all to classification, is able to relieve the computational load and instance acquisition effort, considering all the data attributes accessed each time around. The approach to feature selection in this paper is based on the concept of coherent accumulation of data about class centers with respect to coordinates of informative features. Ranking is done on the degree to which different variables exhibit random characteristics. The results are being verified using the Nearest Neighbor classifier. This also helps to address the feature irrelevance and redundancy, what ranking does not immediately decide. Additionally, feature ranking methods from different independent sources are called in for the direct comparison.

Keywords: classification, feature ranking, feature selection, dimensionality reduction, optimization.

1 Introduction

Supervised Classification implies that unique association of instances with classes of data is known on the training stage for a data sample. This mapping is then used to develop an algorithm by which any new instance can be assigned to a correct class based on the data. A classification algorithm has to be able to deal with computational complexity commonly caused by the magnitude of instances often driven by the multitude of data attributes. This problem is huge in text categorization, every word expanding the attribute space to a whole new dimension. This area received much attention in the past, but continues to be in the focus despite the processing power of computers has increased dramatically. Some terminology has settled over the time. (Saeys et al. 2007) give a contemporary view of *feature selection* methods in bioinformatics.

Without knowing better, we can certainly assume that disengaging of variables, assumed all contributing, will cause reduction of the classification accuracy. We can stage experiments to ascertain influence of different variables, referred commonly to as *features*, indirectly, via responses we get from a classifier. Various models of feature-set are entered sequentially into the classifier, no matter what kind, and the best response is learned. This generic technique of feature selection is called *wrapping*. In this work we use accuracy of the k-NN classifier as an indirect measure of *fitness* of feature-set. Where a pre-selection of features is possible, it is termed *filtering*. Devices of different sorts are in employ, and if they can provide answers to feature *irrelevance* and *redundancy* - whether features align with no class or their input is equivalent to others - the better. Filtering, which can be rather elaborate, is independent from the method of classification, although it inevitably uses the class information. Information Gain and Relief are two filtering techniques considered widely a standard, each coming from a different perspective: probabilistic - the former, deterministic - the latter. Ultimately, there are methods of classification, selecting features to best suit the class distribution for the tune-up. This is referred to as *embedding*. SVM is an example of classifier where feature selection is embedded. We discuss these and other methods when comparing them to those introduced in this paper.

Only wrapping offers a universal approach for feature-set selection. A chosen set has to be *consistent* with the agenda of classification, that is, be sufficient for class discrimination. The enumeration of different subsets of features is computationally challenging. If *monotonicity* holds, so that any addition of a feature can only improve fitness of the current set, the exhaustive search can be escaped via *branch-and-bound* arrangement setting a qualifying level for fitness (Narendra and Fukunaga 1977). While same features may add differently to fitness of different sets, knowing fitness of individual features can be useful. Embedding may or may not produce a shortlist of features best describing data as a whole. In SVM it does. In Decision Trees, instead, one best feature is selected for spawning at different stages of tree growing (Quinlan 1993) with Information Gain often used as the criterion. The feature is different for different subsets of data included in subsequent branches of the tree. *Globally* or *locally*, it helps knowing how to rank features by *relevance*.

While *ranking* of features can be obtained as a byproduct of feature subset selection by a wrapper or embedded method, often ranking is an element of design of filter methods. Conversely, having features ranked is attractive for quick assembly of a desired feature-set. For example, (Huda et al. 2010) pair a Neural Network wrapper with a filter, akin to Information Gain, to facilitate selection of sufficient quality a feature-set. Feature ranking is also the

element of design of the proposed method, although we do not systematically explore the aspect of best feature-set, if only for verification. The ranking is done on the degree to which different variables exhibit random characteristics. Similar methods are known as filters. It may be interpreted as a method of classification adapted for feature ranking. So, it may be seen also as a wrapper or embedded technique. In fact, a number of methods, particularly those used for comparison in this paper, are exposed to such interpretations. These methods share the idea of misclassification counts. We pay Relief (Kira and Rendell 1992, Kononenko et al. 2008) a special attention for conceptual likeness to our method. The algorithm is given a remake to fit the new agenda.

2 Feature Ranking Algorithm

Introduction of a measure of similarity is a founding step in any approach to classification. If a class can be described as a cluster of data points in the problem space, then its center may be defined as a point most similar to them all, that is, containing the class information signature. One measure, commonly used, is distance in the problem space. Any new data can then be class assigned running the affinity check for different class centers. We imply sufficiency of the information signature for class identification.

2.1 Formulation

In this section we lay out an approach, whereby features are selected step-by-step, more informative / relevant first. In the formulation t stands for the iteration number and $I_t \subset \{1, \ \ldots, \ n\}$ are indices of the reduced set of features contending at time t.

Consider data A consisting of $m \geq 2$ classes (finite sets) $A_j \subset \mathbb{R}^n$, $j = 1 \ldots m$, so that: $A_j \neq \emptyset$; $A_{j1} \bigcap A_{j2} = \emptyset$, $\forall j^1, j^2, \ j^1 \neq j^2$; and $A = \bigcup_{j=1}^{m} A_j$. Let a^{ij} be elements of the sets, $i = 1 \ldots |A_j|$, where $|\cdot|$ is the notation for set cardinality.

Let $\|\cdot\|$ defines the metric for $\mathbb{R}^n$ space as follows:

$$\|a-x\| \ = \ (\sum_{l \in I_t} |a_l - x_l|^2)^{1/2} \ , \ \ \forall \, a, x \ \in \ \mathbb{R}^n \ , \ \ n > 1 \ .$$

$\mathbb{R}^n$ is a space of variable dimensionality from 1 to n.

Algorithm 1 (Forward Selection)

Step 1. (Initialization). Set $t = 0, \ I_t = \{1, \ \ldots, \ n\}$.

Step 2. (Class Centers). Determine class centers x^j assuming that sets A_j each form a unique cluster. Compute the centers by solving the following problem of convex programming:

$$\text{minimize} \ \ 1/|A_j| \cdot \sum_i \|a^{ij} - x^j\|^2 \ . \qquad (1)$$

(See Theoretical Aspects.)

Step 3. (Misclassified Points). Find points of sets A_j, which are closer to class centers of other sets coordinate-wise. Let x^j_* be solutions to the Problem 1.

Evaluate sets:

$$N^j = \left\{ a^{ij} \ : \ \min_{s \neq j} |a^{ij} - x^s_*|^2 \leq |a^{ij} - x^j_*|^2 \right\} \ ,$$

where $s = 1 \ldots m$ is the class index.

Get the resulting set:

$$N = \bigcup_{j=1}^{m} N^j \ .$$

The coordinate index $l \in \ I_t$ is implied in the above.

Step 4. (Relevant Attribute). To determine the coordinate of highest relevance find

$$l_* = \arg \min_{l \in I_t} (|N_l|/|A|) \ .$$

If ties exist choose arbitrary.

Step 5. (Contending Features). Make $t = t + 1$, and construct the new set of contributing factors:

$$I_t = I_{t-1} \setminus \ \{l_*\} \ .$$

If $|I_t| \ = 1$ then stop, else go to Step 2.

The $\|\cdot\|$, way we define it, is the radial, or Euclidian, distance. Square omission throughout the algorithm gives rise to formulation in so called Manhattan (the city block), or Hamming distances.

Plainly, the algorithm finds class centers and enumerates elements that belong to a class, but considering a particular feature, are closer to centers of other classes. The total of these counts, normalized by the number of elements in the whole set, establishes the feature rating. The higher the rating, the less relevant is the feature. The best performing feature out, search is repeated again to select a next one. The idea is stemming from the approach suggested in (Bagirov et al. 2003). However, the technique there engages subset selection directly, without having features ranked.

A variable in this method has the higher relevance, the more distant are values taken at class centers. If, instead, a variable has close readings for different classes, this results in the number of misclassified points growth. Obviously, a variable with the same value for all classes is irrelevant given data. Irrelevance correlates with rating close to unity obtained for a feature. However, it is theoretically impossible to judge irrelevance given only data. No matter how big is the set, it is merely a sample revealing the data concept, largely unknown, only partially.

At the same time, the algorithm is adaptable for other search tactics (Saeys et al. 2007). Generally, if the selection criterion is sensitive enough and consistency of the feature-set for classification is not violated, dismissal of insignificant has advantage over selection of significant - despite not the shortest, the complete reliable subset of features is immediately known after each elimination. Certain time saving is achievable on a big set of features if forward selection and backward elimination is done concurrently, that is, one best and one worst feature are taken out in a single swoop, steps 4 and 5 of Algorithm 1 adjusted accordingly for this mixed scheme.

Even with the full set of features, all being relevant, no classifier can guarantee the absolute precision. Harnessing minor features can not help overcoming this inherent classifier limitation. Instead, it may cause overfitting: a classifier gets perfectly trained, but performs poorly on a test data. Yet a classifier can have less overhead if fewer performance boosting features are used, explaining preference that we give to forward selection.

The one cluster per class representation holds by the slim assumption that classes may be described as "connected" and "convex" sets. This can be improved, if classes are subdivided into clusters, although the complexity of the algorithm increases significantly. The Incremental Global Search featuring k-Means by (Bagirov 2008) makes this possible, but other clustering methods are also available. (Kaufman and Rousseeuw 1987) partition data around medoids, so they call their algorithm PAM. They refer to the structural model assumed for data as k-Medoid. Confusion in the literature exists about origins of the k-Medoids algorithm. In fact, k-Medoids is a featured component of PAM. Both k-Means and k-Medoids find only local solutions for k clusters, for the Euclidian or Manhattan metric respectively, and corresponding to the formulation with or without squares. The algorithms do the same by redistributing data between clusters based on proximity of cluster centers. The difference is only in how cluster centers are obtained (Theoretical Aspects). The hard, unconstrained objective applies. That is, each element of data belongs to one and only one natural cluster. The algorithm by (Kaufman and Rousseeuw 1987) is reconfigurable for k-Means. Likewise, the algorithm by (Bagirov 2008) can be recast for k-Medoids. Both algorithms strive to find a near global solution for k clusters.

Algorithm 1 can be generalized as follows.

Algorithm 2 (Class Overlay Counts)

Step 1. (Initialization). Set $t = 0,\ I_t = \{1,\ \ldots,\ n\}$.

Step 2. (Class Centers). Compute centers $x^{jk} \in \mathbb{R}^n$ of clusters A_{jk} making class A_j by solving the following problem of convex programming:

$$\text{minimize}\ \ 1/|A_{jk}| \cdot \sum_i \|a^{ijk} - x^{jk}\|^2\ , \qquad (2)$$

where $a^{ijk} \in A_{jk}$ are the cluster elements, $i = 1 \ldots |A_{jk}|$, $j = 1 \ldots m$, $k = 1 \ldots p_j$. Subdivision of classes into clusters is assumed known.

Step 3. (Misclassified Points). Find points of sets A_{jk}, which are closer to cluster centers of other classes coordinate-wise.

Let x_*^{jk} be solutions to Problem 2. Evaluate sets:

$$N^{jk} = \left\{ a^{ijk} : \min_{s \neq j} \min_r |a^{ijk} - x_*^{sr}|^2 \leq |a^{ijk} - x_*^{jk}|^2 \right\},$$

where $s = 1 \ldots m$ and $r = 1 \ldots p_s$ are the class and the cluster within class indices respectively.

This results in the set:

$$N = \bigcup_{j=1}^{m} \bigcup_{k=1}^{p_j} N^{jk}\ .$$

The coordinate index $l \in\ I_t$ is implied.

Step 4. (Relevant Attribute). To determine the most relevant coordinate find

$$l_* = \arg\min_{l \in I_t} (|N_l|/|A|)\ .$$

If ties exist make an arbitrary choice.

Step 5. (Contending Features). Make $t = t + 1$, and construct the new set of contributing factors:

$$I_t = I_{t-1} \backslash \{l_*\}\ .$$

If $|I_t|\ = 1$ then stop, else go to Step 2.

If Algorithm 2 is reconfigured for backward elimination, it makes sense reclustering data after each cycle. Irrelevant features may cause misrepresentation of the instance space structure by significantly changing distances in concerned directions. This can make subsequent feature deselection less certain.

Even though the unconstrained clustering condition may be fulfilled by class, in the united set it is not guaranteed to hold. Conversely, partitioning of the superset leaves no tension between clusters. The tension between classes helps achieving the algorithm goal. However, the class interaction weakens with number of clusters increasing. Also, smaller clusters are rounder in shape, their eccentricity less expressed.

The circumstance of Algorithm 1 not taking parameters is attractive. In Algorithm 2 the number of clusters per class has to be selected. Generally, more clusters per class should be improving the model description. However, if this number is not small enough, the centers become close to each other and number of instances per cluster small, rendering less reliable counts. Clearly, having statistically sound data props precision of the algorithm.

This does not answer the question, though, of how to choose an appropriate number of clusters for each class - after all, classes may vary in size and have simple or complex mapping. In this regard we propose the following approach: the data is clustered first as a whole with a set number of clusters. A label is assigned to each cluster based on the leading class membership. The classes are then clustered independently using the information obtained. So, we search the data undivided by class for clusters to the best isolation as in (Bagirov 2008). After class labels are assigned to clusters we initiate the standard k-Means procedure (MacQueen 1967) to make clusters conform to the topology of individual classes.

Choice of parameters in Algorithm 2 involves preprocessing and this poses a significant setback. However, if the number of clusters is increased to the number of elements, each point becomes a cluster of its own. This seems to be solving the problem of parameter setting, neither clustering needs to be performed. At the same time, this removes tension between classes. Unless attribute values repeat, no instance can possibly cross to a different class because now the center of a cluster coincides precisely with its only element. All features important - makes the selection a futile exercise. This, however, inspires the idea of following approximation to Algorithm 2.

Algorithm 3 (Estimated Overlay)

Step 1. (Initialization). Set overlay encounter by feature to none: $N_l = \emptyset,\ l = 1 \ldots n$. Iterate by coordinate (index l is implied) with the following.

Step 2. (Closest Points). Find two points for each point a^{ij}: indices of $a1$ are $i_1 \neq i,\ j_1 = j$, and indices of $a2$ satisfy $j_2 \neq j$; that is, points belonging to the same and a different class, but not a^{ij}, so that

$$|a1 - a^{ij}|^2 \ = \ \min_{a \ \in \ A_j \setminus \ \{a^{ij}\}} |a - a^{ij}|^2,$$

and

$$|a2 - a^{ij}|^2 \ = \ \min_{a \ \in \ A \ \setminus \ A_j} |a - a^{ij}|^2.$$

Step 3. (Misclassification). Add point a^{ij} to set N_l if

$$|a1 - a^{ij}|^2 \geq |a2 - a^{ij}|^2.$$

Keep iterating from Step 2 until point and coordinate cycles are exhausted.

Step 4. (Attribute Relevance). To determine relevance of coordinates find $|N_l|/|A|,\ l = 1 \ldots n$. Ordering on this ratio prioritizes the feature relevance.

In words, Algorithm 3 examines each attribute to establish whether it is good a class separator, as if for discretization purposes, that is, whether single class layers of data characterize the variable, or it is inundated by the class mix. This can be improved if k values of each class are drawn in the vicinity of current value and their averaged distances to the instance projection are compared to establish whether the instance is in the midst of its own class. It is clear that ranking obtained by this algorithm is independent from feature selection tactics. It is why usual steps articulating the tactics are not included. The ranking is also independent from the metric of problem space. We include squares for outward compatibility with other algorithms only.

Followed so far is the global feature weighting approach. It can be seen in the light of overlaying distributions for different classes along individual dimensions. Any overlapping of multivariate distributions, the class noise, adds uncertainty, but is not a problem nor the clue to feature weighting. It is the potential overlapping in respect of coordinates that matters. At the same time, data is borderless, represented by a finite set. Without fitting a structural model it is not possible to infer from the position of knowing the data concept. Nonetheless, it is possible to weight features locally by examining immediate neighborhoods of known instances. The results then can be generalized for the whole space hosting the data. This is the idea of Relief (Kira and Rendell 1992). In Relief feature-wise distance differences establish the rating. We use misclassification counts, and so let us call the version ReliefC.

Algorithm 4 (ReliefC)

Step 1. (Initialization). Set the encounter of class mix by feature to none: $N_l = \emptyset,\ l = 1 \ldots n$.

Step 2. (Closest Points). Find two points for each point a^{ij}: indices of $a1$ are $i_1 \neq i,\ j_1 = j$, and indices of $a2$ satisfy $j_2 \neq j$; that is, points belonging to the same and a different class, but not a^{ij}, so that

$$\|a1 - a^{ij}\|^2 \ = \ \min_{a \ \in \ A_j \setminus \ \{a^{ij}\}} \|a - a^{ij}\|^2,$$

and

$$\|a2 - a^{ij}\|^2 \ = \ \min_{a \ \in \ A \ \setminus \ A_j} \|a - a^{ij}\|^2.$$

Step 3. (Misclassification). Update coordinate sets N_l by including points a^{ij} if

$$|a1_l - a_l^{ij}|^2 \geq |a2_l - a_l^{ij}|^2.$$

Reiterate from Step 2 to cover all data.

Step 4. (Attribute Relevance). To determine relevance of coordinates find $|N_l|/|A|,\ l = 1 \ldots n$. Ordering on this ratio prioritizes the feature relevance.

Ranking obtained by this algorithm is space metric dependent. We can achieve refinement of the result if we adopt the tactics of backward elimination. Irrelevant dimensions may cause a significant distortion of the perceived data distribution. We can get a better understanding of other coordinate significance if we run Algorithm 4 again with the confusing attribute withheld, which, of course, can be repeated until each feature rank is adjusted. As in ReliefF (Kononenko et al. 2008) we can draw $k > 1$ nearest neighbors to the current instance for each present class to rely more on the distance statistics.

In Algorithm 4 each point is treated as a self-contained cluster, but having no other cluster elements required in Algorithm 2, we find a closest same class point to the instance of choice, which is to play the role of its cluster center. This makes the approximation. The data mapping used by Algorithm 2 has to be scaled down to fulfill the local feature weighting approach, which is impossible. We can notice semblance of Algorithms 4 and 3. Nevertheless, Algorithm 3 follows global, not the local generalized approach. It is value-wise, but not instance-wise.

2.2 Evaluation

We tested Algorithms 2, 3, 4 and ran a comparison with some other methods of feature ranking from this study and outside sources on data from (UCI Machine Learning Repository) and a proprietary data-set. The data space was assumed Euclidian in all metric dependent algorithms. Characteristics of data examples appear in Table 1. The numbers in columns reflect any transformations data required. "Clusters" applies to the data undivided by class. "k-NN" is the number of nearest neighbors in classification by the k-NN method to verify the results. Other columns are self-explanatory.

Data	Features	Instances	Classes
Housing Prices	13	336	7
Congressional Voting	16	435	2
Wall Following	24	5456	4
Diabetes Diagnostic	63	291	2
Data	Clusters	k-NN	
Housing Prices	21	1	
Congressional Voting	6	3	
Wall Following	20	5	
Diabetes Diagnostic	10	3	

Table 1: Data-sets used in experiments.

Housing Prices in suburbs and their defining factors is a snapshot of state of affairs in Boston, USA some time ago (Harrison and Rubinfeld 1978). The Housing Price is a continuous variable, and so the problem is of regression type. To represent it as a classification problem, we cluster the class variable by the same method we apply to data generally (Bagirov 2008). Whenever a conversion like this takes place, certain amount of noise is inevitably created, as data in the middle, between any two values defining adjacent classes, can not be successfully assigned to either of them. Therefore, the data was denoised after conversion using a technique we developed in (Stranieri and Yatsko 2009). Also, attributes of the data were re-scaled / standardized to zero mean and unit deviation (the absolute mean deviation).

Next two examples are exact classification tasks; neither the data requires a standardization. The Congressional Voting records on selected issues from a particular period in the past for each of the U.S. House of Representatives congressmen, either democrats or republicans, were interpreted by (Schlimmer 1987). The data can be treated as three-value numeric. In the Wall Following case a robot navigates around a room using ultrasonic sensors. This has to be seen as a time series; however, the moves are elementary and replicating: the robot either "follows and follows" directly or it "turns and turns". So, this is approached as a classification problem by (Freire et al. 2009), creators of the data. All the measurements are uniform, also having same upper limit defined by the sensor reachability.

The Diabetes Diagnostics data is a collection of medical records of various signs of presence or absence of this condition in patients and the expert opinion. This data array is available to the University of Ballarat Centre for Informatics and Applied Optimization Health Informatics Laboratory through collaboration with Charles Sturt University under provisions of DiScRi screening research initiative. Although this is a classification problem, it has specifics pertaining to diagnostic applications, with all focus given to a single small subset of data. A mix of attribute types required to be dealt with. So, where appropriate, ordinal attributes were converted to numeric. Otherwise, individual values of discrete attributes were turned into binary attributes. All the attributes except the class, numeric by the end throughout, were standardized to zero mean and unit deviation. Additional preprocessing relieved the data of several attributes inundated by missing values and involved generalization of the class for rare conditions.

Binary attributes qualify as numeric. At the same time, value combinations of binary attributes naturally subdivide the data, creating structures varying in detail, depending on how specific is the combination. This was used for setting missing values of numeric attributes, based on average. Unknown values of binary attributes themselves were entered using the same technique, but based on mode. Only selected binary attributes were used to narrow down the search, their number reduced step-by-step, until values left missing were set from all available data. The class attribute represents a special case. These missing values were set before any others from a predictor earlier identified as the best.

Note, the proposed algorithm of feature ranking can be adapted for missing values given cluster centers, and theoretically even the clustering algorithm can. However, this is not granted in respect of any other technique, and we do require a number of them for comparison. Generally, absence of certain values does not hurt predictability as this may seem - the data structure may make them redundant.

Results of application of feature ranking Algorithm 2 to different data-sets are shown in Tables 2, 3, 4, 5 and 6. In these tables: "Order" is the feature informativeness from highest to lowest - the rank; and "Rating" is the actual value corresponding to the rank as obtained by the algorithm after the first cycle.

2.2.1 Housing Prices

The factors affecting Housing Prices are listed in Table 2, their actual meaning can be found at the source. Representation of factors and specific circumstances have bearing on the ranking. Standing of several aspects of housing generally may be different.

Feature	Order	Rating
Rooms	1	0.6905
Income	2	0.7381
Employment	3	0.7768
Crime	4	0.7857
Pollution	5	0.8006
Industrial Area	6	0.8095
Education	7	0.8274
Building Age	8	0.8452
Black Culture	9	0.8631
Transport Access	10	0.8720
Tax	11	0.9315
Residential Area	12	0.9583
Natural Reserves	13	1.0000

Table 2: Housing Prices: factor significance.

The listing order corresponds to the result of forward selection. However, no shift of position of residual factors occurs through the factor set reduction. This is a characteristic of the formulation used and applies to all data-sets.

First impression of the ranking is that it does not betray the common sense, especially the two factors at the top. Indeed, housing price is higher for more room and with less population on low income. (Harrison and Rubinfeld 1978) also note the clean environment as a factor gaining in significance.

For the Housing Prices data-set results by other authors are also available (Bi et al. 2003), where feature weighting is a byproduct of classification using Support Vectors. Results of numerical experiments are presented graphically as star plots. We estimated feature ranks for comparison out of this representation. The authors specifically mention the number of rooms as the leading factor, influencing positively the housing price. Interestingly, ranks are positively or negatively charged. The next important factor appears to be the income, and it is charged negatively.

2.2.2 Congressional Voting

The Congressional Voting example is good that it refracts feature significance as heat of the debate, whether due to issue controversy or its actuality, and this is what Table 3 is meant to reflect. The topical context has to be examined carefully to fully understand significance of different issues and also be seen in the historical frame, whether they were routine or new matters at that time.

Feature	Order	Rating
Physicians	1	0.0552
Budget	2	0.1356
Education	3	0.1931
Crime	4	0.2299
Nicaragua	5	0.2391
El − Salvador	6	0.2506
Missiles	7	0.2897
Superfunds	8	0.3448
Synfuels	9	0.3586
Exports	10	0.3862
Satellites	11	0.3931
Handicapped	12	0.4069
Religious	13	0.4115
Immigration	14	0.6529
South − Africa	15	0.7678
Water	16	0.7954

Table 3: Congressional Voting: issue controversy.

Although given identifiers do not reveal the full story, it is clear that some up-to-date or pressing issues do occupy leading positions on the list and, instead, some issues of consensus appear down the list. However, there is no clear divide between parties only on two issues of physicians and budget at the top.

2.2.3 Wall Following

This data-set mirrors the Wall Following Robot moves. Table 4 shows significance of one sensor readings above others as obtained by the proposed feature ranking algorithm. The robot has 24 ultrasonic sensors around its "waist", but it is clear that the robot can get away with only two sensors: one tracking the wall, and another the obstacle ahead - the orthogonal wall, what the robot is actually programmed for. At the same time, it is obvious that in a small room or narrow space all or some readings interpret the same information. Represented appropriately, velocity of the robot and / or radius inverse of the turn could capture it all.

Feature	Order	Rating	Feature	Order	Rating
*US*15	1	0.6171	*US*24	13	0.8048
*US*19	2	0.7392	*US*05	14	0.8070
*US*06	3	0.7546	*US*13	15	0.8116
*US*18	4	0.7680	*US*14	16	0.8141
*US*08	5	0.7835	*US*02	17	0.8286
*US*20	6	0.7887	*US*11	18	0.8380
*US*17	7	0.7896	*US*16	19	0.8455
*US*22	8	0.7927	*US*21	20	0.8475
*US*01	9	0.7953	*US*10	21	0.8510
*US*23	10	0.7997	*US*03	22	0.8563
*US*07	11	0.8013	*US*04	23	0.8563
*US*12	12	0.8024	*US*09	24	0.8671

Table 4: Wall Following: sensor informativeness.

According to how the robot circumnavigates the room (clockwise), half of its sensors is on its side nearer to the wall, and other half is sounding the outer space. The sensor numbers (not the rank) closer to the wall are between 13 and 24 with US13 pointing exactly in the opposite direction of the robot and US19 exactly towards the wall. Indeed, we find US19 the second leading feature. Also, among the eight leading features we encounter six sensors next to the wall and, vise versa, among the eight trailing features there are six sensors further from the wall. It is reasonable to assume the sensor range is insufficient to cover the space of the room, which makes sensors next to the wall more valuable predictors. However, US01 pointing directly ahead is not in the leading third. Actually, only one of many comparison methods in the next section places US01 at the top, and none the adjacent sensors. In this regard we have to clarify that shape of the room in (Freire et al. 2009) is not simply rectangular, but has a rectangular concession in one corner, which coerces the robot to make turns not only to the same side (right) but also to the other (left).

Instead of the sensor pointing directly ahead, we have US15 as the leading factor, pointing almost backwards, which is sensible to rely on when making a turn without arriving at the obstacle. US15 actually sounds parallel to the wall in places, because for whatever reason trajectory of the robot is turned by about the same angle as the misdirection of US15 against the robot opposite, as appears on images in (Freire et al. 2009), which also makes the sensor sounding distance shorter. In the limited space of the room many sensors provide reasonable whereabouts, which explains appearance of versions of the data-set with four or even two features, although they are not the readings from sensors pointing in "compass" directions. Indeed, sensors rate close, due likely to their mutual redundancy. Yet the situation comprehensiveness can be improved via sensor combination.

2.2.4 Diabetes Diagnostics

The Diabetes Diagnostics is a medical data-set and without specialist knowledge it is difficult to comment on significance of different symptoms and results of tests. At the same time, because the publicity acknowledged burden of the spread condition on health funds and its link to the obesity, some general awareness exists.

Feature	Order	Rating
DM Diagnostic	1	0.0584
Screening Glucose	2	0.1478
Glucose	3	0.2062
LDL	4	0.2887
HT Diagnostic	5	0.3058
TC	6	0.3127
HbA1c	7	0.4467
HT Status	8	0.4708
LSBP	9	0.4777
DM Family History	10	0.4880
Ewing − Early	11	0.4880
Ewing Score	12	0.5292
DBHR	13	0.5395
BMI	14	0.5533
VAHR	15	0.5704
TC/HDL ratio	16	0.5704
Age	17	0.5876
Ewing − Normal	18	0.6186
DBHR result	19	0.6426
Grade 10 *sec*	20	0.6598
LSHR	21	0.6667
Lying DBP	22	0.6976
HDL	23	0.7320
Triglyceride	24	0.7388
Lying SBP	25	0.7801
PQ 10 *sec*	26	0.7938
Waist Circumference	27	0.8007
QRS 10 *sec*	28	0.8419
HGBP	29	0.8522
QRS Axis 10 *sec*	30	0.8832
QTc 10 *sec*	31	0.9003
Ewing − Atypical	32	0.9141

Table 5: Diabetes Diagnostics: symptom significance.

From Tables 5 and 6 we notice that some top ranking factors do imply the high content of sugars in specimens and, consulting the dictionary, the leading factor, Diabetes Mellitus (DM) diagnostic, appears to be a very specific carbohydrate metabolism disorder, besides reoccurring. While the DM diagnostic may be lacking analytic qualities as a forgone conclusion,

Feature	Order	Rating
QTd 10 sec	33	0.9313
Atrial Fibrillation	34	0.9588
Ewing − Definite	35	0.9588
Hearth Attack	36	0.9656
Pain in Left Arm	37	0.9794
CVD Diagnostic	38	0.9863
Palpitations	39	0.9863
Smoking	40	0.9897
Stroke	41	0.9931
Nausea	42	0.9931
Vomiting	43	0.9931
LSHR result	44	0.9931
VAHR result	45	0.9931
QTc 10 sec $> 1/2$	46	0.9931
Gender	47	0.9966
CVD Status	48	0.9966
Angina	49	0.9966
Hearth Failure	50	0.9966
Chest Pain	51	0.9966
CA Neuropathy	52	0.9966
Bloating	53	0.9966
Abdominal Pain	54	0.9966
Alcohol	55	0.9966
CVD Family History	56	0.9966
HGBP result	57	0.9966
Ewing − Severe	58	0.9966
QTc 5 min $> 1/2$	59	0.9966
Dizziness	60	1.0000
Pacemaker	61	1.0000
LSBP negative	62	1.0000
LSBP result	63	1.0000

Table 6: Diabetes Diagnostics: symptom significance (continued).

a number of factors immediately after it show a very strong predictive ability of the condition according to the rating. General awareness factors have an advanced position on the list, but can not be a match for specialist testing. The DM family history and the age appear in the first third of the list. Perhaps the waist circumference in the first half of the list by itself does not offer a measure sensitive enough without linking to other sizes, like height. It is likely, though, that BMI (the body mass index) in the first third of the list does take this into account. The host of factors towards the end of the list is either general complicities or those having a circumstantial effect. However, misplacement of two last features may have had occurred because the data scarcity and substitution of missing values. A number of features in this data-set are present in both numerical and categorical forms, one implying the other.

At the end of this section let us restate that no additional passes through Algorithm 2 are required to fine-tune the ranking, unless Algorithm 2 is run in the mode of backward elimination and the data-set is restructured. The observation that the order of significance of features does not change in the reduced set is a characteristic of the formulation used. Following result holds (applicable also to Algorithm 2):

Proposition 1 *Rating of features as obtained after application of Algorithm 1 does not change after a feature is removed from the set.*

This follows directly from the lemma proved in Theoretical Aspects. △

3 Comparison

Classification opens a way for indirect comparison, as previously explained, and there are different methods of feature ranking that can be compared directly with.

3.1 Classification

The results were undergone a verification using a classification method. The purpose was to ascertain that performance of the classifier is predictable. This is exactly the wrapping technique, except the ranking is known beforehand and only designated combinations of features need testing. Specifically, we are interested to find how the accuracy changes when features are subtracted from the end or beginning of the ranked list. The accuracy is found via the leave-one-out procedure, always fetching unwavering results, a special case of multi-fold cross-validation whereby credibility of each instance class is tested in turn against the whole instance base excepting that instance. The result will fluctuate if folds of cross-validation contain more than one instance.

The Nearest Neighbor classifier (k-NN) is easy implementable and suits our approach that it deals in distances. A good survey of instance-based methods, those with k-NN in their core, is contained in (Wilson and Martinez 2000). Precisely, the crisp version of fuzzy k-NN algorithm described in (Keller et al. 1987) is used. The only difference we introduce is that all neighbors are included in the radius given by the farthest of k initially selected nearest neighbors of the instance to be classed from the reference base. This allows to capture all repeating instances. Where data was treated for noise, a single nearest neighbor is often the best. Where it was not, a bigger k may be more optimal, as appears in Table 1, although in the Wall Following example $k = 1$ is still the best.

Computation results appearing next as accuracy percentile charts represent classification series driven by leading (or trailing) feature-sets, that is, with features below (or above) any given line on the ranked list removed to obtain each series element.

3.1.1 Housing Prices

The k-NN classification accuracy on the Housing Prices data-set in the forward run, first series on Figure 1, exhibits a slow decreasing trend at the beginning, getting more intense as factors are discarded one-by-one from the end of ranked list, that is, less informative first. The slide of accuracy at the run end reflects its accumulated loss as weighed against significance of the topmost factors. Conversely, the backward run, second series on Figure 1, where more informative factors are discarded first, is notable for abrupt, going less dramatic fall of accuracy; respective of the order but with some grouping; although unable to contain the accumulated loss at the end, having no factors of significance left. These two observations thus support the result of ranking.

(Bi et al. 2003), while applying Support Vector Machines, compare responses from the classifier with and without some unrelated variables mixed in, so that features performing no better than the artificial ones could be discarded as irrelevant. They could not find any not contributing, however, in the Housing Prices example. This corresponds to our finding, despite the other authors scheme is for regression, not for classification, which required us to discretize the class variable. We should expect a temporary increase of accuracy when discarding irrelevant features and this does not happen.

However, irrelevance can be full or it can be partial. We may find a number of features being a burden on a classifier overall, while contributing for

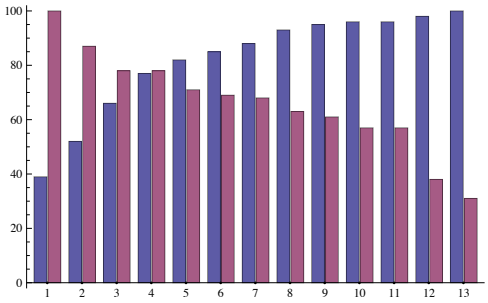


Figure 1: Housing Prices classification accuracy change with leading features removed first (2nd series, left-to-right) or last (1st series, right-to-left).

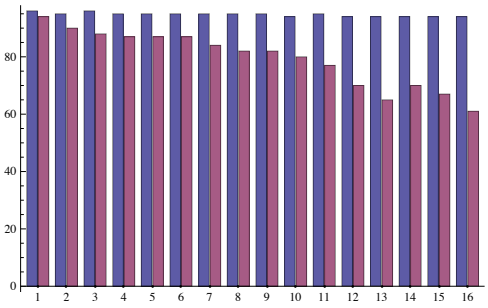


Figure 2: Congressional Voting classification accuracy change with leading features removed first (2nd series, left-to-right) or last (1st series, right-to-left).

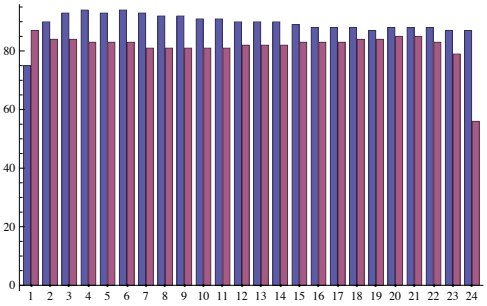


Figure 3: Wall Following classification accuracy change with leading features removed first (2nd series, left-to-right) or last (1st series, right-to-left).

Figure 4: Diabetes Diagnostics classification accuracy change with leading features removed first (2nd series, left-to-right) or last (1st series, right-to-left).

a small subset of data. If no such subset exists then, of course, these features are fully irrelevant, as long as the data truly represents the underlying concept. The subtle differences may be lost in data conversion.

3.1.2 Congressional Voting

It is remarkable that the Congressional Voting data shows no visible loss of classification accuracy throughout in the forward run, represented by the first series on Figure 2. At the same time, the backward run exhibits a profile suggesting significance of leading factors when removed - the second series on Figure 2. The increase of accuracy at the end of backward run belies the insignificance of left features as k-NN switches to the implicit mode of predicting the biggest class anyway with all irrelevant features, which is an issue with imbalanced data-sets. Closer, behind the chart, result examination reveals that parity of class prediction deteriorates abruptly, reducing to zero by the end. Note, irrelevance does not mean the question on agenda is unimportant, simply parties both agree or disagree. One could be interested to see the feature list from this perspective.

At the same time, the accuracy in the forward run does not grow noticeably. This can be simply because the accuracy is already high at the beginning, and there is a limit of achievable with k-NN. Although, there are may be a background connection between debated topics. For instance, despite all of them may seem independent, there is a monetary component to any of them, which the budget attribute, emerging at the top of significance list, fully embraces.

3.1.3 Wall Following

In predicting Wall Following Robot moves the features are likely much related. Nonetheless, the forward run, first series on Figure 3, is characterized by a notable growth of accuracy, continuing up to the moment when only four top features is left. This can not be explained by irrelevance. The second series on Figure 3 certainly does not confirm this. These features are all redundant, and the improvement is purely due to reduction of overhead - distance calculations for k-NN classification become simpler. Although the backward run has a weak profile, it supports correctness of the feature ranking. Despite the accuracy elevates slightly again by the end, it is still below the level the forward run even takes off.

This leads to quite a different idea of how a feature list may be shortened, not only from the position of little relevance. If groupings of similar features are known, then keeping top features of each lot is sufficient. Yet, if it is not for the knowledge of domain, then how to tell that features are redundant?

3.1.4 Diabetes Diagnostics

The Diabetes Diagnostics is certainly an example of feature little relevance or even irrelevance. At the same time, features in the first half of the list are much different in relevance, spanning rating from very small to very high, see Table 5. We observe from the forward run, first series on Figure 4, that the classification accuracy is only increasing, allowed some fluctuation. This chart shows features in groups of three counted off the end of ranked relevance list.

The backward run, depicted as the second series on Figure 4, arranged similarly, portrays the

increasing insignificance of left features. Again, because this set is imbalanced, which is usual for medical diagnostics, it has the problem previously explained, causing the accuracy to increase at the end of run, while the relevance of features left is vanishing.

3.2 Ranking

The end result of feature rating is the order of informativeness, to know what features to keep and which to discard. It is only by luck that ratings calculated by different algorithms are compatible. There must be a way to compare methods based only on ranking they produce, and this is what we deal with in this section. A number of independent methods of feature ranking is brought in for comparison, (Weka Data Mining Tools) being the main source, find guidance where required from the book by (Witten and Frank 2005). We identify all these methods in Table 7.

ID	Name	Design	Origin
NNS	Single Feature k-NN	Wrapper	Proposed
NNX	Feature Excepted k-NN	Wrapper	Proposed
EO	Estimated Overlay	Filter	Proposed
RC	ReliefC	Filter	Proposed
RF	ReliefF	Filter	Weka
$H2$	Chi Square	Filter	Weka
IG	Information Gain	Filter	Weka
$1R$	One Attribute Rule / One R	Embed.	Weka
SVM	Support Vector Machine	Embed.	Weka

Table 7: Feature ranking methods for comparison.

3.2.1 Alternative Methods of Ranking

Let us recount methods sourced from (Weka Data Mining Tools) first, although this does not explain their exact implementation.

Chi Square statistic and Information Gain are probabilistic filters, which expect nominal data, but this can be arranged through discretization.

Chi Square statistic is the mean quadratic deviation of observed against expected frequencies, approaching one of the classic types of probability distribution introduced by Pearson when the number of data points increase. Each attribute produces a different result depending on how closely it follows the expected frequency for each class. No difference means that the feature is contributing nothing special to classification. So, smaller values of the statistic correspond to higher independence of the class from a given attribute, and feature ranks are assigned accordingly. (Liu and Setiono 1997) adapt a Chi Square discretization method by (Kerber 1992) to rank features on the number of intervals, different for different attributes, but obtained with the same significance level for the statistic.

Information Gain is often a choice among methods using probabilities. It is based on calculation of Entropy, a measure of uncertainty of a particular outcome. Entropy is calculated for each class and the total is found. It is then reduced by entropy calculated on class posterior probabilities for each value of a variable. The difference is the Information Gain. The less uncertain outcome from using a feature, the more is the information gain, and this makes the basis for ranking. The technique is widely used in Decision Trees (Quinlan 1993). (Fayyad and Irani 1993) extend the splitting mechanism of decision trees to discretization of features. Because they utilize the Minimum Description Length principle, put into theory by (Rissanen 1978), in the stopping criteria, potentially, this also can be used for ranking of features by the number of intervals.

Methods not using probabilities but having a statistical interpretation are as follows.

One Attribute Rule, used for classification, compares different attributes and relies solely on the attribute giving the least error (Holte 1993). The error is how features are rated. This is an embedded technique that could be identified as a filter, the wrapping clad kind, if not the design hierarchy.

The idea of Relief by (Kira and Rendell 1992) is that a feature should have distinct readings for different classes about same locality. Therefore, we can find two closest instances of data to the instance acting as probe, of a different and the same class, and subtract coordinate distances to these points. Positive differences characterize inner points of a class. The feature-wise differences are then averaged for a random selection of instances or all data to obtain weights for ranking. Larger weights identify features of better class separation. The technique is a filter: its design has a connection to, but does not include classification by k-NN. ReliefF is a multi-class implementation of Relief, taking care of noisy and incomplete data by (Kononenko et al. 2008).

The simplification One Attribute Rule implies makes it more of a filter than embedded type, and so is ReliefF. Both have certain design similarities with our method. The algorithm of Class Overlay Counts is of filter type, although a substantial preparatory phase is involved if it is not run in the simple mode.

Support Vector Machine (SVM), as a method of classification, finds separation hyperplanes maximizing the margin between classes, for which purpose it locates base points called support vectors. This results in weighting of variables, establishing their ranks. Clearly, this is an example of embedded method. Other authors result of using SVM on the Housing data (Bi et al. 2003) is also available.

Also included are: two k-NN wrapper estimators and two alternative ranking schemes of own making. The wrapping technique for obtaining ranking from accuracy of classification by nearest neighbors is using either single features or sets found by exception of single feature. No forward selection or backward elimination was pursued on this occasion to enhance the selection. Estimated Overlay and ReliefC are the two alternative schemes suggested in this study to circumvent necessity to cluster the data. ReliefC is an interpretation of Relief, counting occurrences of class overlap instead of summing up the standard feature-wise distance differences.

3.2.2 Method of Ranking Comparison

One approach to ranking goodness evaluation is extraction of longest sequence of preserved order of features of a scheme taken for a standard. The discrepancy with total number of features is then relative error, or variation. This method of comparison, while clear for understanding, on implementation side is not trivial. Also, it is not taking into account local changes of the position, tending to overestimation

of error, especially on long feature-sets. Sequences of different length may be compared instead. By this method features of the alternative ranking, the principal sequence includes up to a given position, are counted. This turns to be similar to a method of Kendall discussed in (Bhamidipati and Pal 2006) if rankings are compared instead of ratings.

However, our way of implementing it gave no different result than simply summing up absolute displacements of the rank for each feature, which is attributed to Spearman (Bhamidipati and Pal 2006). Therefore, we use this simple and recognized method to represent results of comparison. Thus found totals are rated by a like result, obtained for opposite ordering of the principal sequence, to measure how "wrong" the contending ranking is. Approaches of forward selection and backward elimination explain interest to validity of leading or trailing features ranked by significance. Therefore, comparison of ranking as produced by Algorithm 2 with other methods is conducted not only for all features but also for leading and trailing thirds of the list.

3.2.3 Ranking Comparison Results

Table 8 summarizes results of feature ranking using different methods, identified in Table 7, for each of examined sets. The comparison elements are: the absolute rank displacement for all features (1st-), for top (-2nd) and bottom (-3rd) portions of the ranked list. Columns denote alternative ranking schemes stated in Table 7. Because ranking differences on short data-sets can be rather imprecise, and to get a qualitative rather than quantitative evaluation of different methods, we use tenths rather than hundredths (percents) parts of the whole.

Dataset	NNS	NNX	EO
Housing Prices	6-4-4	5-1-3	2-2-1
Congressional Voting	2-1-1	6-5-3	0-0-0
Wall Following	4-3-3	5-2-3	7-6-5
Diabetes Diagnostic	5-4-4	4-3-3	2-3-0
Dataset	H2	IG	1R
Housing Prices	3-3-1	4-4-2	3-3-3
Congressional Voting	2-1-1	1-1-1	2-1-1
Wall Following	4-2-3	4-3-3	4-3-3
Diabetes Diagnostic	1-1-0	1-1-0	5-3-4
Dataset	RC	RF	SVM
Housing Prices	2-0-1	6-6-4	6-6-3 4-1-4
Congressional Voting	5-3-4	5-3-3	6-5-5
Wall Following	5-4-4	7-6-5	5-3-5
Diabetes Diagnostic	3-2-2	4-2-5	6-5-4

Table 8: Ranking method comparison summary on variation scale of 0 to 10.

Of all represented methods Chi Square and Information Gain give the best support for the proposed method of Class Overlay Counts. This is not a surprise because Chi Square and Information Gain are based on the same idea in the guise of probabilities. Estimated Overlay is of the same type, but is much dependent on data. On one occasion we see a significant departure from the principal method, and on a different occasion we obtain fully indifferent ranking, thus making the comparison trivial. Estimated Overlay offers, otherwise, a very undemanding alternative to the main method.

One Attribute Rule and Single Feature k-NN give the proposed method a more cautious support than Chi Square or Information Gain. Single Feature k-NN is a wrapper and One Attribute Rule can be interpreted as a wrapper, because it calculates the prediction error. Single Feature k-NN has specifics that can make it insensitive to irrelevant features, as k-NN switches into the mode that simply predicts the biggest class, and on imbalanced data-sets this accuracy can be high. This, however, affects only the end of ranked list. Wrapper methods are thus potentially exposed to a loss of detective ability on features of little relevance, and so the backward elimination makes a wrong design of feature selection algorithms relying on wrapping.

While ReliefF is not a wrapper and Feature Excepted k-NN is, these two methods use the same principle of k-NN and do produce similar results but are less supportive of the proposed ranking, even to the point of disagreement. Interestingly, ReliefC, despite being akin to ReliefF, gives much closer results overall. Feature-wise distance differences in ReliefF do appear more ambiguous than class overlap in ReliefC. As to Feature Excepted k-NN, it has the limitation that the impact on classification accuracy of a single feature missing from the set can be very small, resulting in features rating close assigned same rank. Also, the method can mistake redundant features for those with little expression.

A Support Vector Machine (SVM) has a very different design than the rest of methods, although it is not a fact that SVM has much a different idea about what the correct ranking should be like, because the independent results by other authors for the Housing data, appearing as the second line, are encouraging. We found though that SVM can be computationally very demanding and unpredictable even on small feature-sets. The necessity to output a unique ranking possibly makes the algorithm loop.

Overall, probabilistic schemes used for comparison are in a good alliance with the feature ranking method we propose, better than techniques not using probabilities, while the proposed two alternatives to the main method are competitive.

4 Conclusion

In this paper an approach to dimensionality reduction of the problem space through feature selection is proposed. It is based on the concept of coherent accumulation of data about class centers for informative features. Those in accord with this property can be short-listed to represent the data or, alternatively, discordant features can be removed, allowing for faster classification and data acquisition. The conclusive rating of features becomes known after the first cycle of the algorithm, making it possible to do without selection refinement of residual feature-sets.

Comparison with other methods of feature ranking shows a good correlation in many cases. However, assumptions the proposed algorithm relies on must be upheld. Firstly, the model should allow interpretation of classes as unique, rotund in shape sets, or classes can be subdivided into such clusters. Secondly, better results can be expected on statistically abundant data. The former poses a dilemma between getting quick results and getting the data model right first. Quicker results may be desirable in some circumstances, so alternatives in the spirit of main algorithm are considered, although the clustering does not bear hugely on performance. The latter is rather broad. In this regard the method shares assumptions of many other algorithms using

misclassification counts, whether in the guise of probability or accuracy of classification.

The algorithm outputs a ranked list, where it is only possible to say that a feature up the list is more relevant than a feature down the list. It is impossible to brand a feature irrelevant, although wrapping, supplementing results of ranking, can help to make the deselection. It appears that removal of less informative features from the end of the list may result in a temporary increase of classification accuracy before its starts to fall. This indicates that features removed may be irrelevant. The classification on results of ranking may also show that some, listed one after another features are similar by their action, that the top feature in a lot carries essentially the same information as the rest. The accuracy plateaus when these features get removed. However, ranking by itself cannot answer the question of redundancy.

5 Theoretical Aspects

Assume, without loss of generality, that there is just one set A. The objective function in Problem 1 then can be expressed as follows:

$$1/|A| \cdot \sum_i \|x - a^i\|^2 \; = \; \sum_l (1/|A| \cdot \sum_i |x_l - a_l^i|^2). \qquad (3)$$

Proposition 1 is consequent from following.

Lemma 1 *Minimizer of Problem 1 obtained on Step 2 of Algorithm 1 is the by-coordinate minimizer.*

Proof. The objective function is representable in a form of sum of non-negative continuous functions of their arguments, according to Expression 3. Because a global minimum exists for each of the components it exists for the compound function. The exact location of the minimum is governed by interaction between components. In this case components are independent of each other. Therefore, the compound minimum is sum of minimums of the components. Besides, each component is represented by a single variable and all variables are included in the total. Thus, minimizers of Problems 1 in respect of coordinates together make the minimizer of Problem 1. △

The above applies to the Euclidian metric. However, it is easy to see that Lemma 1 also holds for the Manhattan metric, all what is required is omission of squares in Expression 3.

Lemma 1 is essential for Algorithm 1. Nevertheless, even a stronger result holds.

Proposition 2 *Solution to Problem 1 is the centroid of elements making the class in the Euclidian metric.*

Proof. Taking partial derivatives from Expression 3 for the objective function by each of coordinates l and equating them to zero we obtain:

$$\sum_i (2 \cdot x_l - 2 \cdot a_l^i) = 0.$$

It immediately follows that

$$x_l = 1/|A| \cdot \sum_i a_l^i,$$

which is exactly the by-coordinate expression for the centroid vector. The solution delivers a minimum, because second derivatives are all greater than zero, and so the Hessian matrix of the objective function at the point is positive definite. It is also the only minimum as no constraints apply. △

To do the same in the Manhattan metric we do require Lemma 1 though.

Proposition 3 *Solution to Problem 1 is the medoid of elements making the class in the Manhattan metric.*

Proof. Expression 3 for the objective function by coordinate, unsquared, with index l omitted for clarity, and scaling factor $1/|A|$, a positive constant, dropped for convenience, can be rewritten as:

$$E \; = \; \sum_i |x - a^i| \; = \; E_1 \; + \; \Delta E \; + \; E_2 \; ,$$

where

$$E_1 \; = \; \sum_i (a^p - a^i) \;\; , \;\; i \; \leq \; p \; ,$$

$$E_2 \; = \; \sum_i (a^i - a^p) \;\; , \;\; i \; > \; p \; ,$$

$$\forall \; p \; \in \; \{1 \; \ldots \; |A| - 1\}$$

and

$$\Delta E \; = \; (p - (|A| - p)) \cdot (x - a^p) \;\; , \;\; a^p \; \leq \; x \; \leq \; a^{p+1} \;\; .$$

This describes change of the objective function, linear on a segment between any two element values, all arranged in increasing sequence, which can be done without loss of generality. The first derivative, or slope of function E on this segment is

$$s \; = \; 2 \; \cdot \; p \; - \; |A| \;\; , \;\; a^p \; \leq \; x \; \leq \; a^{p+1} \;\; .$$

For small p it is negative as $|A| \; > \; 2 \; \cdot \; p$ and is increasing with p. Conversely, $s > 0$ for large p. Thus, E being continuous decreases with p, but reaches a minimum when the slope is minimal. This depends on whether $|A|$ odd or even.

If $|A|$ is odd, and so the number of intervals is even, the minimizer is

$$x^* \; = a^p \;\; , \;\; p \; = \; \lfloor |A| \; / \; 2 \rfloor \; + \; 1 \;\; .$$

If $|A|$ is even, and so the number of intervals is odd, the whole middle interval is the minimizer:

$$a^p \; \leq \; x^* \; \leq \; a^{p+1} \;\; , \;\; p \; = \; [|A| \; / \; 2] \;\; .$$

Often a single reference point is taken to represent a minimizer that is not unique. In this case by convention it is calculated as mean of interval ends:

$$x^* \; = \; (a^p \; + \; a^{p+1}) \; / \; 2 \;\; .$$

Thus found point for any $|A|$ is known as the median of increasing sequence of values and for all coordinates as the medoid. △

The term of medoid was introduced by (Kaufman and Rousseeuw 1987), so as not to confuse a new notion with that of median. In their clustering algorithm medoid is the instance chosen to be central

in a cluster. Elsewhere, the notion of medoid is being used in the sense of definition above. The two do not contradict. Indeed, the reference point can be shifted to any of vertices of the minimum.

Acknowledgement

The authors are thankful to anonymous reviewers of this submission for supplying valuable and stimulating comments on how this paper can be improved.

References

Bagirov A. M. (2008), Modified global k-means algorithm for minimum sum-of-squares clustering problems. *in* 'Pattern Recognition', 10(41): 3192-3199.

Bagirov A. M., Rubinov A. M., Soukhoroukova N. V., Yearwood J. (2003), Unsupervised and supervised data classification via nonsmooth and global optimization. *in* 'TOP: Spanish Operations Research Journal', 11(1): 1-93.

Bhamidipati N.L., Pal S.K. (2006), Comparing rank-inducing scoring systems. *in* 'Proceedings of ICPR 2006 - 18th International Conference on Pattern Recognition', 3: 300-303. IEEE.

Bi J., Bennett K. P., Embrechts M., Breneman C. M., Song M. (2003), Dimensionality reduction via sparse support vector machines. *in* 'Journal of Machine Learning Research', 3: 1229-1243.

Fayyad U.M., Irani K.B. (1993), Multi-interval discretization of continuous-valued attributes for classification learning. *in* 'Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence', 1022-1027. Morgan Kaufmann.

Freire A.L., Barreto G.A., Veloso M., Varela A.T. (2009), Short-term memory mechanisms in neural network learning of robot navigation tasks: a case study. *in* 'Proceedings of LARS 2009 : the 6th Latin American Robotics Symposium', 1-6. Valparaiso, Chile.

Harrison D., Rubinfeld D.L. (1978), Hedonic prices and the demand for clean air. *in* 'Journal of Environmental Economics and Management', 5: 81-102.

Holte R.C. (1993), Very simple classification rules perform well on most commonly used databases. *in* 'Machine Learning', 11: 63-91.

Huda S., Jelinek H., Ray B., Stranieri A., Yearwood J. (2010), Exploring novel features and decision rules to identify cardiovascular autonomic neuropathy using a hybrid of wrapper-filter based feature selection. *in* 'Proceedings of the Sixth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)', 297-302. IEEE.

Kaufman L., Rousseeuw P.J. (1987), Clustering by means of medoids. *in* 'Y. Dodge (editor) Statistical Data Analysis based on L1 Norm', 405-416. Elsevier / North-Holland.

Keller J.M., Gray M.R., Givens J.A. (1985), A fuzzy k-nearest neighbor algorithm. *in* 'IEEE Transactions on Systems, Man and Cybernetics', 15(4): 580-585.

Kerber R. (1992), Chimerge: discretization of numeric attributes. *in* 'Proceedings of the Tenth National Conference on Artificial Intelligence', 123-128. AAAI / MIT Press.

Kira K., Rendell L. (1992), The feature selection problem: traditional methods and a new algorithm. *in* 'Proceedings of the Ninth National Conference on Artificial Intelligence', 129-134, AAAI Press.

Kononenko I., Robnik-Šikonja M. (2008), Non-myopic feature quality evaluation with (R)ReliefF. *in* 'Liu H., Motoda H. (editors): Computational Methods of Feature Selection', 169-191. Chapman & Hall / CRC.

Liu H., Setiono R. (1997), Feature selection via discretization. *in* 'IEEE Transactions on Knowledge and Data Engineering', 9(4): 642-645.

MacQueen J. (1967), Some methods for classification and analysis of multivariate observations. *in* 'Proceedings of the Fifth Berkley Symposium on Mathematical Statistics and Probability', 281-297.

Narendra P.M., Fukunaga K. (1977), A branch and bound algorithm for feature subset selection. *in* 'IEEE Transactions on Computers', 26(9): 917-922.

Quinlan J. (1993), C4.5: Programs for machine learning. Morgan Kaufmann.

Rissanen J. (1978), Modeling by shortest data description. *in* 'Automatica', 14: 465-471.

Saeys Y., Inza I., Larrañaga P. (2007), A review of feature selection techniques in bioinformatics. *in* 'Bioinformatics', 23(19): 2507-2517.

Schlimmer J.C. (1987), Concept acquisition through representational adjustment. *in* 'Doctoral dissertation', University of California at Irvine, USA.

Stranieri A., Yatsko A. (2009), Capped k-NN editing in definition lacking problems of classification. *in* 'University of Ballarat Research Repository', 1-16. Internet: "http://arrow.edu.au".

UCI Machine Learning Repository. Internet: "`http://mlearn.ics.uci.edu/`"

Weka Data Mining Tools. Internet: "`http://www.cs.waikato.ac.nz/ml/weka/`"

Wilson D. R., Martinez T. R. (2000), Reduction techniques for instance-based learning algorithms. *in* 'Machine Learning', 38: 275-286.

Witten I.H., Frank E. (2005), Data mining: practical machine learning tools and techniques. 2-nd edition. Morgan Kaufmann.

Improving Naive Bayes Classifier Using Conditional Probabilities

SONA TAHERI[1] **MUSA MAMMADOV**[1,2] **ADIL M. BAGIROV**[1]

[1]Centre for Informatics and Applied Optimization, School of Science, Information Technology and Engineering, University of Ballarat, Victoria 3353, Australia
[2] National Information and Communications Technology Australia (NICTA)

Emails: sonataheri@students.ballarat.edu.au, m.mammadov@ballarat.edu.au, a.bagirov@ballarat.edu.au

Abstract

Naive Bayes classifier is the simplest among Bayesian Network classifiers. It has shown to be very efficient on a variety of data classification problems. However, the strong assumption that all features are conditionally independent given the class is often violated on many real world applications. Therefore, improvement of the Naive Bayes classifier by alleviating the feature independence assumption has attracted much attention. In this paper, we develop a new version of the Naive Bayes classifier without assuming independence of features. The proposed algorithm approximates the interactions between features by using conditional probabilities. We present results of numerical experiments on several real world data sets, where continuous features are discretized by applying two different methods. These results demonstrate that the proposed algorithm significantly improve the performance of the Naive Bayes classifier, yet at the same time maintains its robustness.

Keywords: Bayesian Networks, Naive Bayes, Semi Naive Bayes, Correlation

1 Introduction

Classification is the task to identify the class labels for instances based on a set of features, that is, a function that assigns a class label to instances described by a set of features. Learning accurate classifiers from pre classified data is an important research topic in machine learning and data mining. One of the most effective classifiers is Bayesian Networks (Shafer 1990, Heckerman 1995, Jensen 1996, Pearl 1996, Castillo 1997). A Bayesian Network (BN) is composed of a network structure and its conditional probabilities. The structure is a directed acyclic graph where the nodes correspond to domain variables and the arcs between nodes represent direct dependencies between the variables. Considering an instance $X = (X_1, X_2, ..., X_n)$ and a class C, the classifier represented by BN is defined as

$$\arg\max_{c \in C} P(c|x_1, x_2, ..., x_n) \propto \arg\max_{c \in C} P(c)P(x_1, x_2, ..., x_n|c), \quad (1)$$

where x_i, c are the values of X_i, C respectively.

However, accurate estimation of $P(x_1, x_2, ..., x_n|c)$ is non trivial. It has been proved that learning an optimal BN is NP-hard problem (Chickering 1996, Heckerman 2004). In order to avoid the intractable complexity for learning BN, the Naive Bayes classifier has been used. In the Naive Bayes (NB) (Langley 1992, Domingos 1997), features are conditionally independent given the class. The simplicity of the NB has led to its wide use, and to many attempts to extend it (Domingos 1997). Since NB assumes the strong assumption of independency between features, learning semi Naive Bayes has attracted much attention from researchers (Langley 1994, Kohavi 1996, Pazzani 1996, Friedman 1997, Kittler 1986, Zheng 2000, Webb 2005). The semi Naive Bayes classifiers are based on the structure of NB, requiring that the class variable be a parent of every feature. However, they allow additional edges between features that capture correlation among them. The main aim in this area of research has involved maximizing the accuracy of classifier predictions.

In this paper, we propose a new version of the Naive Bayes classifier (semi Naive Bayes) without assuming independence of features. The proposed algorithm finds dependencies between features using conditional probabilities. This algorithm is a new algorithm and different from the existing semi Naive Bayes methods (Langley 1994, Kohavi 1996, Pazzani 1996, Friedman 1997, Kittler 1986, Zheng 2000, Webb 2005).

Most of data sets in real world applications often involve continuous features. Therefore, continuous features are usually discretized (Lu 2006, Wang 2009, Ying 2009, Yatsko 2010). The main reason is that the classification with discretization tend to achieve lower error than the original one (Dougherty 1995). We apply two different methods to discretize continuous features. The first one, which is also the simplest one, transforms the values of features to $\{0, 1\}$ using their mean values. We also apply the discretization algorithm using sub-optimal agglomerative clustering algorithm from (Yatsko 2010) which allows us to get more than two values for discretized features. This leads to the design of a classifier with higher testing accuracy in most data sets used in this paper.

We organize the rest of the paper as follows. We give a brief review to the Naive Bayes and some semi Naive Bayes classifiers in Section 2. In Section 3, we present the proposed algorithm. Section 4 presents an overview of the discretization algorithm using sub-optimal agglomerative clustering. The numerical experiments are given in Section 5. Section 6 concludes the paper.

2 Naive Bayes and Semi Naive Bayes Classifiers

The Naive Bayes (NB) assumes that the features are independent given the class, it means that all features have only the class as a parent (Kononenko 1990, Langley 1992, Domingos 1997, Mitchell 1997). A sample of the NB with n features is depicted in Figure 1. The NB, classifies an instance $X = (X_1, X_2, ..., X_n)$ using Bayes rule, by selecting

$$\arg\max_{c \in C} P(c) \prod_{i=1}^{n} P(x_i|c). \quad (2)$$

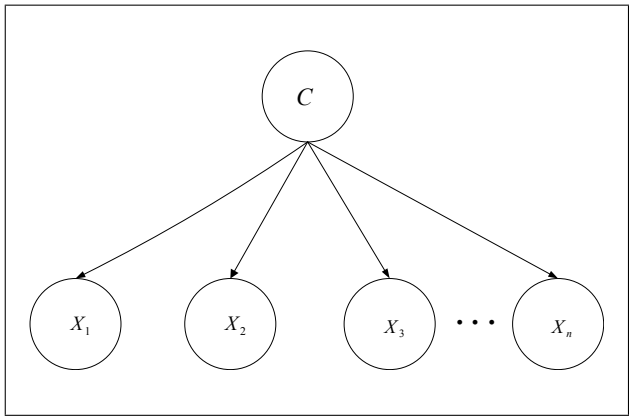


Figure 1: Naive Bayes

NB has been used as an effective classifier for many years. Unlike many other classifiers, it is easy to construct, as the structure is given a priori. Although the independence assumption is obviously problematic, NB has surprisingly outperformed many sophisticated classifiers, especially where the features are not strongly correlated (Domingos 1997). In spite of NB's simplicity, the strong independency assumption harms the classification performance of NB when it is violated. On the other hand, learning BN requires searching the space of all possible combinations of edges which is NP-hard problem (Chickering 1996, Heckerman 2004).

In order to relax the independence assumption of NB, a lot of effort has focussed on improving NB. The improved NB classifiers use exhaustive search to join features based on statistical methods. There are some improved algorithms of the NB. Langley and Sage (Langley 1994) considered Backwards Sequential Elimination (BSE) and Forward Sequential Selection (FSS) in which their methods select a subset of features using leave-one-out cross validation error as a selection criterion and establish a NB with these features. Starting from the full set of features, BSE successively eliminates the features whose elimination most improves accuracy, until there is no further accuracy improvement. FSS uses the reverse search direction, that is iteratively adding the features whose addition most improves accuracy, starting with the empty set of features. The work of Pazzani (Pazzani 1996) introduces Backward Sequential Elimination and Joining (BSEJ). It uses predictive accuracy as a merging criterion to create new Cartesian product features. The value set of a new compound features is the Cartesian product of the value sets of the two original features. As well as joining features, BSEJ also considers deleting features. BSEJ repeatedly joins the pair of features or deletes the features that most improves predictive accuracy using leave-one-out cross validation. This process terminates if there is no accuracy improvement. Kohavi (Kohavi 1996) proposed the NB Tree, a strategy that is a hybrid approach combining NB and decision tree learning. It partitions the training data using a tree structure and establishes a local NB in each leaf. It uses 5-fold cross validation accuracy estimate as the splitting criterion. A split is defined to be significant if the relative error reduction is greater than 5 percent and the splitting node has at least 30 instances. When there is no significant improvement, NB Tree stops the growth of the tree. As the number of splitting features is greater than or equals one, NB Tree is an x-dependence classifier. The classical decision tree predicts the same class for all the instances that reach a leaf. In NB Tree, these instances are classified using a local NB in the leaf, which only considers those non tested features. Friedman et al. (Friedman 1997) introduced Tree Augment Naive Bayes (TAN) based on tree structure. It approximates the interactions between features by using a tree structure imposed on the NB structure. In TAN, each feature has the class and at most one other feature as parents. Super Parent algorithm is proposed by Keogh and Pazzani (Keogh 1999). This algorithm uses the same representation as the Tree Augment Naive Bayes, but utilizes leave-one-out cross validation error as a criterion to add a link. The Super Parent is the feature that is the parent of all the other orphans, the features without a non-class parent. There are two steps to add a link: first selecting the best Super Parent that improves accuracy the most, and then selecting the best child of the Super Parent from orphans. This method stops adding links when there is no accuracy improvement. Zheng and Webb (Zheng 2000) developed Lazy Bayesian Rules (LBR), which adopts a lazy approach, and generates a new Bayesian rule for each test example. The antecedent of a Bayesian rule is a conjunction of feature-value pairs, and the consequent of the rule is a local NB, which uses those features that do not appear in the antecedent to classify. LBR stops adding feature value pairs into the antecedent if the outcome of a one tailed pairwise sign test of error difference is not better than 0.05. As the number of the feature value pairs in the antecedent is greater than or equals one, LBR is anx-dependence classifier. Webb et al. (Webb 2005) proposed Averaged One Dependence Estimators (AODE), which averages the predictions of all qualified 1-dependence classifiers. In each 1-dependence classifier, all features depend on the class and a single feature.

In the next section, we introduce a new version of the Naive Bayes classifier (semi Naive Bayes) without assuming independence of features. The proposed algorithm approximates the interactions between features by using conditional probabilities.

3 The Proposed Algorithm

In this section, we present a new algorithm that maintains the basic structure of the NB, and thus ensure that the class C is the parent of all features. The proposed algorithm, however, removes the strong assumption of independence in the NB by finding correlation between features, while also capturing much of the computational efficiency of the NB. In this algorithm, the class has no parents and each feature has the class and at most one other feature as parents. Therefore, each feature can have one augmenting edge pointing to it. The procedure for learning these edges is based on the Pearson's correlation and conditional

probabilities. First, we construct a basic structure of the NB with n features $X_1, X_2, ..., X_n$ from the set X and the class C. After that, we find the Pearson's correlations between each feature X_i and the class C using the formula (3), $Corr(X_i, C)$. Then we re-order the set X as a set X^* in a descending order of $|Corr(X_i, C)|$. In the ordered set X^*, an arc from the first feature is added to the second one. Finally, for all remain features, we find the conditional probabilities of each feature with the previous features given the class values in the ordered set X^*, formula (4). The highest value of these conditional probabilities between features is used to recognize the parent of each feature. The conditional probabilities described in (4), first introduced by Quinn et al. (Quinn 2009) and called influence weights, have been used directly for data classification. However, here, we used them for finding the dependencies between features.

The correlation coefficient (Graham 2008) between two random variables X_i and X_j is defined as :

$$Corr(X_i, X_j) = \frac{N \sum_{i,j=1}^{N} X_i X_j - \sum_{i=1}^{N} X_i \sum_{j=1}^{N} X_j}{\sqrt{(N \sum_{i=1}^{N} X_i^2 - (\sum_{i=1}^{N} X_i)^2)(N \sum_{j=1}^{N} X_j^2 - (\sum_{j=1}^{N} X_j)^2)}}, \quad (3)$$

where N is the number of data points. This measure has the property of $|Corr(X_i, X_j)| \leq 1$. When this value is close to 1, it denotes the perfect linear correlation between X_i and X_j, and $Corr(X_i, X_j) = 0$ stands for no linear correlation.

The proposed algorithm consists of six main steps:

Algorithm. Proposed Algorithm

Step 1. Construct a basic structure of the Naive Bayes with n features, $X = \{X_1, X_2, ..., X_n\}$, and the class C.

Step 2. Compute the correlation between each feature $X_i,\ i = 1, ..., n$ and the class C using the formula (3), $Corr(X_i, C)$.

Step 3. Reorder X as a set $X^* = \{X_1^*, X_2^*, ..., X_n^*\}$ in a descending order of $|Corr(X_i, C)|,\ i = 1, ..., n$.

Step 4. Add an arc from X_1^* to X_2^*.

Step 5. For $j = 3, ..., n$:

5.1 Find X_i^* that has the highest value of

$$\sum_{k=1}^{N} |P(X_{ki}^*, X_{kj}^* | C) - P(X_{ki}^*, X_{kj}^* | \overline{C})|,\ i < j, \quad (4)$$

where $X_i^* = (X_{1i}^*, X_{2i}^*, ..., X_{Ni}^*)^T$, N is the number of instances and $\overline{C} = -C$.

5.2 Add an arc from X_i^* to X_j^*.

Step 6. Compute the conditional probability tables inferred by the new structure.

Figure 2 shows the structure of Svmguide1 data set, taken from LIBSVM, with four features (see Table 1) using the proposed algorithm. The solid lines are those edges required by the Naive Bayes classifier. The dashed lines are correlation edges between features found by our algorithm.

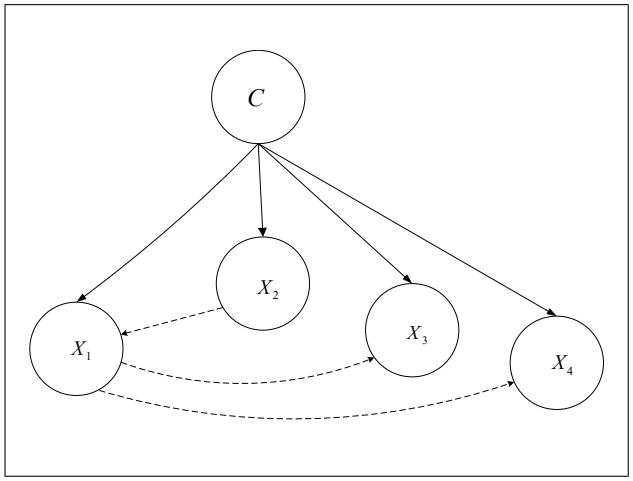


Figure 2: Proposed algorithm, Svmguide1

4 Discretization Algorithm Using Sub-Optimal Agglomerative Clustering (SOAC)

Discretization is a process which transform continuous numeric values into discrete ones. In this paper, we apply two different methods to discretize continuous features. The first one, which is also the simplest one, transforms the values of features to 0,1 using their mean values. We also apply the discretization algorithm using sub-optimal agglomerative clustering algorithm which allows us to get more than two values for discretized features. In this section, we introduce discretization algorithm SOAC which is an efficient discretization method for the NB learning. Details of this algorithm can be found in (Yatsko 2010).

Consider a finite set of points A in the n dimensional space R^n, that is $A = \{a^1, ..., a^m\}$, where $a^i \in R^n,\ i = 1, ..., m$. Assume that the sets $A^j,\ j = 1, ..., k$ be clusters, and each cluster A^j can be identified by its centroid $x^j \in R^n,\ j = 1, ..., k$. The discretization algorithm SOAC proceeds as follows.

Algorithm. Discretization Algorithm SOAC

Step 1. Set $k = m$, and a small value of parameter θ, $0 < \theta < 1$. Sort values of the current feature in the ascending order. Each feature requiring discretization is treated in turn.

Step 2. Calculate the center of each cluster:

$$x^j = \sum_{a \in A^j} \frac{a}{|A^j|},\ j = 1, ..., k$$

and the error E_k of the cluster system approximating set A:

$$E_k = \sum_{j=1}^{k} \sum_{a \in A^j} \|x^j - a\|^2.$$

Step 3. Merge in turn each cluster with the next tentatively. Calculate the error increase after each merge $E_{k-1} - E_k$ and choose the pair of clusters giving the least increase. Merge these two clusters permanently. Set $k = k - 1$.

Step 4. Once the error of the current cluster system is over the set fraction of the maximum error corresponding to the single cluster $E_k \geq \theta E_1$ stop, otherwise go to Step 2.

5 Numerical Experiments

To verify the efficiency of the proposed algorithm, numerical experiments with a number of real world data sets have been carried out. We use 10 real world data sets. The detailed description of the data sets used in this experiments can be found in the UCI machine learning repository, with the exception of "Fourclass", "Svmguide1" and "Svmguide3". These three data sets are downloadable on tools page of LIBSVM. A brief description of data sets is given in Table 1. We discritize the values of features in data sets using two different methods. In the first one, we apply a mean value of each feature variable to discritize the values to $\{0, 1\}$. The second one is the discrization algorithm SOAC (Yatsko 2010) which is presented in Section 4.

We conduct empirical comparison for the NB and the proposed algorithm in terms of test set accuracy using two different discritization methods. The results of the NB and the new algorithm on each data set were obtained via 1 run of 10-fold cross validation. Runs were carried out on the same training sets and evaluated on the same test sets. In particular, the cross validation folds were the same for all experiments on each data set.

The test set accuracy obtained by the NB and the proposed algorithm on 10 data sets using mean values for discretization summarized in Table 2. The results presented in this table demonstrate that the test set accuracy of the new algorithm is much better than that of the NB. The proposed algorithm works well in that it yields good classifier compared to the NB. Its performance was further improved by introducing some additional edges in the NB, using conditional probabilities. Improvement is noticed mainly in large data sets. In 8 cases out of 10, the new algorithm has higher accuracy than the NB. The accuracy of this algorithm is same with the NB in data sets Fourclass and Svmguide1.

Table 3 presents the test set accuracy obtained by the NB and the proposed algorithm on 10 data sets using discretization algorithm SOAC. The results from this table show that the accuracy obtained by the new algorithm in all data sets are higher than those obtained by the NB.

Figures 3 to 4 show the scatter plot comparing the proposed algorithm with the NB, using two different discritization methods. In these plots, each point represents a data set, where the x coordinate of a point is the percentage of miss classifications according to the NB and the y coordinate is the percentage of miss classification according the proposed algorithm. Therefore, points above the diagonal line correspond to data sets where the NB performs better, and points below the diagonal line correspond to data sets where the proposed algorithm performs better.

According to the results explained above, the proposed algorithm outperforms the NB, yet at the same time maintains its robustness. However, the proposed algorithm requires more computational effort than the NB since we need to compute conditional probabilities between features to recognize the parent of each feature in our algorithm.

Table 1: A brief description of data sets

Data sets	# Features	# Instances
Congres Voting Records	16	435
Credit Approval	14	690
Diabetes	8	768
Fourclass	2	862
Haberman Survival	3	306
Heart Disease	13	270
Phoneme CR	5	5404
Spambase	57	4601
Svmguide1	4	7089
Svmguide3	21	1284

Table 2: Test set accuracy of NB and the proposed algorithm using mean value for discretization

Data Sets	Naive Bayes	Proposed Algorithm
Congres Voting Records	90.11	91.47
Credit Approval	84.85	86.85
Diabetes	75.78	77.68
Fourclass	76.82	76.82
Haberman Survival	74.51	75.66
Heart Disease	84.14	85.18
Phoneme CR	75.96	78.30
Spambase	90.13	93.45
Svmguide1	92.17	92.17
Svmguide3	80.61	87.18

Table 3: Test set accuracy of NB and the proposed algorithm using discretization algorithm SOAC

Data Sets	Naive Bayes	Proposed Algorithm
Congres Voting Records	90.11	91.47
Credit Approval	84.85	86.85
Diabetes	75.78	77.68
Fourclass	78.58	79.70
Haberman Survival	74.66	75.33
Heart Disease	78.62	79.31
Phoneme CR	77.01	79.36
Spambase	89.30	92.30
Svmguide1	95.61	97.54
Svmguide3	77.25	80.85

6 Conclusion

In this paper, we have developed the new version of the Naive Bayes classifier without assuming independence of features. An important step in this algorithm is adding edges between features that capture correlation among them. The proposed algorithm finds dependencies between features using conditional probabilities. We have presented the results of numerical experiments on 10 data sets from UCI machine learning repository and LIBSVM. The values of features in data sets are discritized by using mean value of each feature and applying discretization algorithm SOAC. We have presented results of numerical experiments. These results clearly demonstrate that the proposed algorithm significantly improve the performance of the Naive Bayes classifier, yet at the same time maintains its robustness. Furthermore, this improvement becomes even more substantial as the size of the data sets increases.

7 References

References

Castillo, E., Gutierrez, J.M & Hadi, A.S. (1997), Expert Systems and Probabilistic Network Models, Springer Verlag, New York.

Chang, C., & Lin, C. (2001), A library for support vector machines, Software available at http://www.csie.ntu.edu.tw/cjlin/libsvm.

Charniak, E. (1991), E. Bayesian Networks Without Tears. AI Magazine, 12 (4).

Chickering, D.M. (1996), Learning Bayesian Networks is NP-complete. In: Fisher, D., Lenz, H. Learning from data: Artificial Intelligence and statistics V, Springer, pp. 121–130.

Domingos, P., & Pazzani, M. (1997), On the optimality of the simple Bayesian classifier under zero-one loss. Mach Learn 29, pp. 103–130.

Dougherty, J., Kohavi, R., & Sahami, M., (1995), Supervised and unsupervised discretization of continuous features. In Proceedings of the 12th International Conference on Machine Learning, pp. 194–202.

Friedman, N., Geiger, D., & Goldszmidt, M. (1997), Bayesian network classifiers. Machine Learning 29 , pp. 131–163.

Graham, E., (2008), CIMA Official Learning System Fundamentals of Business Maths. Burlington : Elsevier Science and Technology.

Heckerman, D., Geiger, D., & Chickering, D.M. (1995), Learning Bayesian Networks: the Combination of Knowledge and Statistical Data. Machine Learning, 20, pp. 197–243.

Heckerman, D., Chickering, D.M., & Meek.C. (2004), Large-Sample Learning of Bayesian Networks is NP-Hard. Journal of Machine Learning Research, pp. 1287–1330.

Jensen, F. (1996), An Introduction to Bayesian Networks. Springer, New York.

Kohavi, R. (1996), Scaling up the accuracy of naive-Bayes classifiers: a decision-tree hybrid. In: Proc. 2nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, pp. 202–207.

Keogh, E.J., & Pazzani, M.J., (1999), Learning augmented Bayesian classifers: A comparison of distribution-based and classification-based approaches. In: Proc. Int. Workshop on Artificial Intelligence and Statistics, pp. 225–230.

Kittler, J. (1986), Feature selection and extraction. In Young, T.Y., Fu, K.S., eds.: Handbook of Pattern Recognition and Image Processing. Academic Press, New York.

Kononenko, I. (1990), Comparison of Inductive and Naive Bayesian Learning Approaches to Autpmatic Knowledge Acquisition. In Wielinga, B., Boose, J., B.Gaines, Schreiber, G., van Someren, M., eds. Current Trends in Knowledge Acquisition. Amsterdam: IOS Press.

Langley, P., Iba, W., & Thompson, K. (1992), An Analysis of Bayesian Classifiers. In 10th International Conference Artificial Intelligence, AAAI Press and MIT Presspp, pp. 223–228.

Langley, P., & Saga, S. (1994), Induction of selective Bayesian classifiers. In: Proc. Tenth Conf. Uncertainty in Artificial Intelligence, Morgan Kaufmann , pp. 399–406.

Lu, J., Yang, Y., & Webb, G. I. (2006), Incremental Discretization for Naive-Bayes Classifier, Springer, Heidelberg, vol. 4093, pp. 223–238.

Mitchell, T.M. (1997), Machine Learning. McGraw-Hill, New York.

Pazzani, M.J. (1996), Constructive induction of Cartesian product attributes. ISIS: In- formation, Statistics and Induction in Science, pp. 66–77.

Pearl, J. (1988), Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann.

Quinn, A., Stranieri, A., Yearwood, J., & Hafen, G. (2009), A classification algorithm that derives weighted sum scores for insight into disease, Proc. of 3rd Australasian Workshop on Health Informatics and Knowledge Management, Wellington, New Zealand.

Shafer, G., & Pearl, J. (1990), Readings in Uncertain Reasoning. Morgan Kaufmann, San Mateo, CA.

Wang, S, Min, Z., Cao, T., Boughton, J., & Wang, Z. (2009), OFFD: Optimal Flexible Frequency Discretization for Naive Bayes Classification, Springer, Heidelberg, pp. 704–712.

Webb, G.I, Boughton, J., & Wang, Z. (2005), Not so naive Bayes: Aggregating one- dependence estimators. Machine Learning 58, pp. 5–24.

Yatsko, A., Bagirov, A. M., & Stranieri, A. (2010), On the Discretization of Continuous Features for Classification. School of Information Technology and Mathematical Sciences, University of Ballarat Conference, http://researchonline.ballarat.edu.au:8080/vital/access/manager/Repository.

Ying, Y., & Geoffrey, I., (2009), Discretization For Naive-Bayes Learning: Managing Discretization Bias And Variance, In Machine Learning, 74(1): pp. 39–74.

Ying, Y., (2009), Discretization for Naive-Bayes Learning, PhD thesis, school of Computer Science and Software Engineering of Monash University.

Zheng, Z., & Webb, G.I. (2000), Lazy learning of Bayesian rules. Machine Learning 41, pp. 53–84.

UCI repositary of machine learning databases (http://archive.ics.uci.edu/ml/datasets.html)

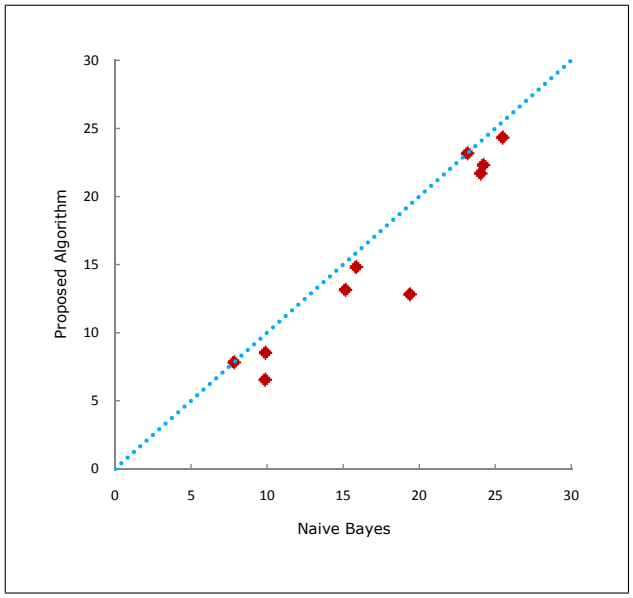


Figure 3: Scatter plot comparing miss classifications of the proposed algorithm (y coordinate) with Naive Bayes (x coordinate); using mean value for discritization

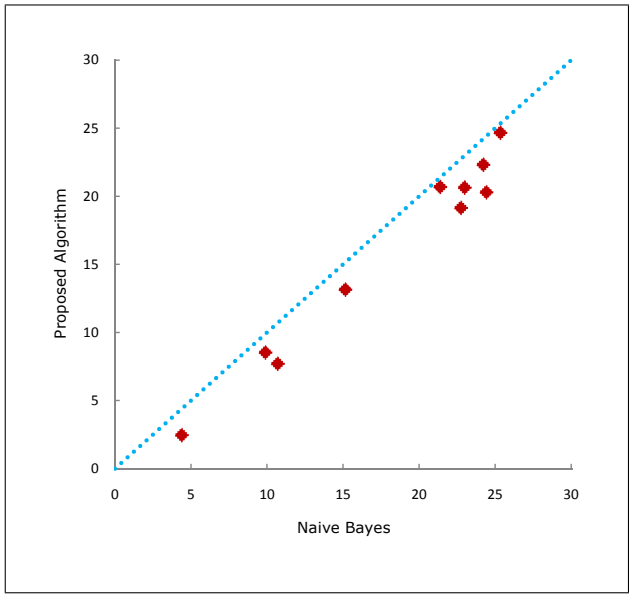


Figure 4: Scatter plot comparing miss classifications of the proposed algorithm (y coordinate) with Naive Bayes (x coordinate); using Algorithm SOAC for discritization

Concept Based Query Recommendation

Poonam Goyal
Department of Computer Science
Birla Institute of Technology & Science
Pilani, India 333 031

poonam@bits-pilani.ac.in

N Mehala
Department of Computer Science
Birla Institute of Technology & Science
Pilani, India 333 031

mehala@bits-pilani.ac.in

Abstract

For a search engine, the challenge of finding relevant information from the web is becoming more and more difficult with rapid increase/change in content of the web. This difficulty further increases as queries submitted by users are general, imprecise, short and ambiguous. Relevance between user's information need and documents returned by search engine is largely dependent on the query given by them. In this paper, we have proposed a method to facilitate users with query recommendations which are the concepts related to their information needs. In this work, we have extracted concepts from the web snippets and we have proposed two weight functions to measure the relevance between query and concepts. Related concepts with different meaning are selected and recommended as query suggestions. To evaluate our method, we have used a Google middleware for the extraction of concepts. We have estimated the relevance between the query and concepts using the proposed weight functions and compared with the support of the concepts as well as with the TFIDF approach using the standard information-retrieval metrics of precision and Mean Average Precision(MAP). We show that our approach leads to gains in average precision than the other existing approach for different type of queries.

Keywords: Query recommendation, search engine, concepts, and weight function.

1 Introduction

For a search engine, finding relevant information/documents from the web is nowadays becoming a challenging task. The most common factors are 1. Rapid growth in the number of pages indexed in a search engine. 2. Short and ambiguous queries submitted by the web users. 3. Ineffective organization of the search results 4. User's different goals and expectations from the web etc. Lot of focus is on finding the relevance between the query and documents by many researchers like Frakes, Baeza-Yates and Fang, Tao and C.Zhai (2004).The search engines generally use the keywords or keyword phrases lying in the query for finding related documents. For example; the queries "apple iPod" (an MP3 player) and "apple pie" (a dessert) are very similar since they both contain the keyword "apple." However, the queries are actually expressing two different search needs. Some researchers focused on clustering similar queries to recommend the URL's to frequently asked queries of a search engine referenced in Beeferman, Berger(2000) and Zahera, El Hady, bd El-Wahed(2010) and Li, Yang, Liu, Kitsuregawa(2008), Wen, Nie, Zhang(2001), Zaiane and Strilets(2002) and Zhiyuan and Maosong(2008). Association rules are applied to relate the similar queries by Fonseca, Golgher, Moura, and Ziviani (2003).

In finding the relevance between the query and documents, query plays an important role. Terms in the query are not always a good descriptor of the information needs of the user as these may have ambiguous meaning or users may specify insufficient or very short queries. Jansen, Spink, Bateman, and Saracevic (1998) concluded in his study that most queries are short and imprecise. Many search engines provide query suggestions to users to formulate more effective queries such as Yahoo's "Also Try", Google's "Searches related to" etc. These suggestions are semantically related, but mostly start with the terms which users have used in their queries. Thereby, ambiguity which was present in the original query still remains in it. As mentioned by Zhang and Nasraoui (2006), client side query recommendation has been suggested by studying and modelling users' sequential search behaviour. The clustering and organizing users' queries into a hierarchical structure of topic classes is proposed by Chuang and Chien (2003). The concept of query flow graph is introduced by Boldi, Bonchi, Castillo, Donato, and Vigna (2009) for query recommendations. The same concept is extended by Anagnostopoulos, Becchetti, Castillo, and Gionis (2009) with an idea of probabilistic reformulation query graph. Most of the researchers have focused on the clustering or classification of the queries without considering the ambiguous meaning of the queries. A two phase classification method for short and ambiguous queries is used for query enrichment and categorization in Shen, Pan, Sun, Jeffrey, Wu, Yin and Yang (2006). In this approach a query is mapped to the intermediate objects which are nothing but some of the selected categories of ODP directory and then query are mapped to one/few

intermediate categories/objects. A limitation of the same is query enrichment and expansion can be done based on the existing manually formed category. Not all the pages indexed by the search engine can be mapped with the manually built existing hierarchy.

This paper focuses on the identification of ambiguous meaning of the given query. Concepts which are important terms or phrases in the search results can be good descriptors of the search query so as to identify the ambiguity. In this paper, a method for identifying the precise meaning of search queries has been proposed. This method facilitates the users to get the required information easily and quickly. The method suggests users the related concepts from each category corresponding to different meanings of the query as query recommendations. Thereby, users can build a proper search query according to their intent and with the knowledge domain terminology which will help search engine to get the desired results.

Moreover, many queries consist of newly created words and sometimes their meanings evolve over time based on Jansen, Spink, Bateman, and Saracevic (1998). Traditional approach of query expansion won't work due to the lack of the clear definition of the query terms. For example, the well known query "apple" can now be interpreted as "Ariane Passenger PayLoad Experiment". Since our approach categorizes the concepts at the first time when user posed a query and presents the categories and their related concepts to the user so it helps users to select the appropriate query.

The main contributions of the paper are as follows

1. Two measures are proposed and established that identify the relevant concepts for the given query by filtering irrelevant concepts.
2. A simple method for query recommendation is proposed which includes forming the dynamically building categories from the identified relevant concepts without the knowledge of existing manually built category hierarchy like ODP, Yahoo, Google etc.

The paper is organized as follows. Section 2 describes the methodology of the proposed method in detail. Experimental setup and results are described in Section 3. Section 4 states our basic findings and future work objectives.

2 METHODOLOGY

Our approach is to recommend the related queries to the search engine users by extracting the important concepts from the search results. Given a query as input the proposed approach works in three steps:

1. Extract concepts from the document snippets returned by the search engine.
2. Identify the relevant concepts for the given query. We have proposed two weight functions to measure the relevance between query and concepts. Concepts with greater weights (greater than some threshold) are considered.
3. Recommend concepts as query suggestions by categorizing the related concepts in case of ambiguous query.

2.1 Concept Extraction

When a query is supplied to a search engine, documents based on the general context of the query assumed by search engine are returned. This causes the users to visit most of the pages to get the desired information. We believe that the necessity of visiting each page could be removed if the concepts, i.e. over-arching ideas of the underlying page, could be revealed to the end user as recommendations. This would require mining the concepts from documents that results in search. It is our contention that this could be done automatically, rather than relying on the current convention of mandating that the searchers extract these concepts manually through examination of result links. This ability to mine concepts would not only be useful for query recommendation but also in further identifying relevant pages.

A search result usually includes the URL, title and snippet of its corresponding Web page. Most users are unwilling to wait for the system to download the original Web pages. Therefore we use a Web page's snippet in the search results for concept extraction. The purpose of concept extraction is to build an index of the terms that occur in the document collection. Not all of the terms in the document collection should be included in the index. Generally nouns, verbs and noun phrases are most predictive of the content of the documents. We identify these terms while ignoring others. We also eliminate words commonly found in Web documents that are considered as stop-words.

2.2 Relevance between concept and Query

All the terms/concepts extracted from the search results might not be relevant to the query. Thereby, query and concept relevance has to be estimated. For this, two weight functions are proposed and applied. The first function is using the content of the retrieved documents alone whereas the second is making use of the content as well as the link structure of the documents. We have used top N document's snippets for this purpose.

2.2.1 Weight Function 1 (W1)

In order to estimate the relevance between query and concepts/terms in the documents returned by the search engine, we have identified the following important factors. First, concepts/terms should occur in more number of document's snippets i.e. concepts should be frequent. Second, along with the frequent occurrence these must be good descriptors of the documents. Usually if the terms appear more times in a document then these might be good descriptors of the document. But many terms may occur many times in many documents without much relevance like the terms "news", "articles" etc. These are frequently occurring terms in news article documents. But these are not good descriptors of the documents. General importance of these words is high. This concludes that the general importance of the terms should be less to be good descriptors of the documents. This factor is similar to the TFIDF value of query term and the document. Here we have extended the idea of TFIDF for concepts with respect to the documents which contain these concepts. Third, collective importance of the concept with respect to documents containing it.

The following is the formula to calculate the weight $W_1(Q, C_i)$ of concept C_i with respect to query Q :

$$W_1(Q, C_i) = \left(\sum_{\forall d_i contains C_i} \left(\frac{n_{ij}}{\sum_{k=1}^{z} n_{kj}} \log\left(N / m_i \right) \right) \right) \frac{m_i}{N}$$

Where n_{ij} is a number that the number of times the concept C_i occurs in the document d_j, z is total number of distinct terms in the document d_j. $\sum_{k=1}^{z} n_{kj}$ is the total number of terms in the document d_j. N is the total number of documents considered. m_i is the total number of documents in which the concept C_i appears. The weight is divided by the number of documents considered i.e. N . The concepts, for which the weights are greater than the threshold δ , are considered to be relevant to the query.

2.2.2 Weight Function 2 (W2)

This function exploits both the content and the link structure of the documents to find the relevance between concept and the query. In the function above, we have estimated the relevance between the concepts and the query based on the content of the document. It is good to consider the content of the neighbouring pages to estimate the importance of the concept with respect to the document. For example the concept "system" is occurring frequently in a document obtained for a query "apple". By considering the document content alone, the query "apple" and the concept "system" will be considered more related as "system" is frequent in the document. The importance of the concepts like "system" can be balanced by considering the contents of the specific neighbouring pages. It is expected that these concepts would not be highly frequent in these neighbours.

Through the links (both in-links and out-links) users can navigate the neighbouring pages and use the information contained in it. All the linked documents may not be related to the same query concept. Thereby, we have given the weightage to the neighbouring pages which contain the concept. We have added an additional factor to the W1 that is the popularity score of the document based on the popularity score of the neighbouring pages considered.

The following is the formula to calculate the weight $W_2(Q, C_i)$ of concept C_i with respect to query Q :

$$W_2(Q, C_i) = \left(\sum_{\forall d_i contains C_i} \left(\frac{n_{ij}}{\sum_{k=1}^{z} n_{kj}} \log\left(N / m_i \right) + PS(j) \right) \right) \frac{m_i}{N}$$

Where $PS(j)$ is the popularity score of the document d_j containing the concept C_i and is described by the following

$$PS(j) = \frac{d}{N} + (1 - d) \sum_{z=1}^{n} \frac{PS(W_z)}{C(W_z)}$$

Where $W_1, W_2, \ldots W_n$ are the documents that points to the document d_j which contains the concept C_i. $C(W_z)$ is the number of outgoing links from the page W_z that contains the concept C_i. d is a dumping factor whose default value is set to 0.15.

The concepts, for which the weights are greater than the threshold δ , are considered to be relevant to the query.

The term $PS(j)$ in W2 plays a role of balance factor. That is it increases or decreases the value of the weight function W_1 according to the degree of relevance of the concept and query. This is because W2 includes both the content and link structure of the respective documents. Thus, it clearly separates the important relevant concepts from the unimportant concepts as compare to the W1. However, the W2 uses the documents to explore the in/out links of the documents. Therefore, the W1 can be treated as light weight function in comparison to W2.

2.2.3 Other weight functions

The performance of the two proposed weight functions W1 and W2 is compared with that of support and TFIDF.

Support (S) :

Support is well known measure in finding the frequent item sets in data mining as mentioned by Boldi, Bonchi,Castillo, Donato, and Vigna(2009), and Zahera, El Hady and Abd El-Wahed(2010). If a keyword or a phrase appears frequently in the web-snippets of a query, it may represent an important concept. The following formula for support S as a relevance measure is used by Leung, Ng and Lee(2008):

$$S(Q, C_i) = \left(\frac{m_i}{N} \right) t_i$$

Where t_i is the number of terms in the concept C_i .

TFIDF:

TFIDF is another well known and common measure to describe the text feature on vector space information retrieval paradigm introduced by Sparck Jones (1972).

TFIDF can highlight keywords that distinguish classes, and reducing the influence of keywords that can't distinguish classes effectively. TF (Term Frequency) is the frequency that the keyword occurs in the text, DF (Document Frequency) is the frequency of documents that contain the keyword. If a keyword occurs often in one document, while rarely in others, then the keyword should be suitably used for classification.

We have used the following TFIDF formula as relevance measure among document D_j and concept T_i :

$$TFIDF(D_j, T_i) = \frac{n_{ij}}{\sum_{k=1}^{z} n_{kj}} \log\left(N / m_i \right)$$

2.3 Concept Categorization and Query Recommendation

In the previous two sections we have extracted and then selected the related concepts from the document snippets

using the proposed weight functions. These concepts along with queries can be recommended as query suggestions. But, as stated earlier that most of the queries are short and imprecise as mentioned by Jansen, Spink, Bateman, and Saracevic (1998). Therefore, it is likely for a query to have multiple meanings. These meanings of the query are captured by the categorization of the obtained concepts. Moreover, many queries consist of newly created words. For example, the word "green apple" is now referring to "Tourism". Some times their intended meanings are changing over time. For example, the well known query "apple" can now be interpreted as "Ariane Passenger PayLoad Experiment". To catch all the under lying ideas/meanings, it is required that the query categorization should take place at the first time when user posed a query. For this, we have proposed a simple model without including the knowledge of manually built existing hierarchy like ODP, Yahoo, Google etc. This is because not all the pages indexed by the search engine can be mapped with the categories of manually built hierarchies.

We have used correlation function for concept categorization. However, any other categorization function like extended Jaccard, LSI etc. may work. We have formed the correlation matrix of the concepts based on the documents that contain them. The concepts which are not linked together (unrelated) are separated by partitioning the correlation matrix. These unrelated concepts will have zero/less values in the matrix. After partition step, concepts in a partition will form a category. Each category will represent a different intention of the query. As of now, the label of the category is given manually. For example, for the query "apple", categories formed by the concepts are "Fruit", "Apple Products" and "Tourism" etc (see Section 3).

After categorization we get categories along with the concepts under a given query referring the different intentions of the query. In the last step of the method the categories along with the concepts are presented to the users as query suggestions. This allows users to build a proper search query with the knowledge domain terminology which will help search engine to get the desired results.

3 EXPERIMENTATION

In this section, we have presented the results obtained by proposed method. Experimental setup is described (see section 3.1). Performance evaluation of the proposed method with two proposed weight functions and results of query recommendation are given. (See section 3.2). The performance of the proposed weight functions is compared with that of S and TFIDF. The overall precision of the proposed method is also given.

3.1 Experimental Setup

We have implemented Google middleware for evaluation of the proposed method. When a query is submitted to the Google search engine, search results will be processed by the middleware (see Figure 1).

We have considered top N (N =100) retrieved web snippets for concept extraction. We have used MontyLingua 2.1 tool [19] for the purpose of concept extraction. MontyLingua is a free, commonsense-enriched, end-to-end natural language understander for English. Feed raw English text into MontyLingua and the output will be a semantic interpretation of that text. Perfect for information retrieval and extraction, request processing, and question answering. From English sentences, it extracts subject/verb/object tuples, extracts adjectives, noun phrases and verb phrases, and extracts people's names, places, events, dates and times, and other semantic information. We have used this tool to extract the subject/verb/object tuples, noun phrases and verb phrases from the web snippets as concepts after removing the stop words from the web snippets. Extracted concepts are stored and their relevance with queries is estimated using the weight functions W1, W2, S and TFIDF. Concepts with weights greater than the threshold δ are considered for concept categorization. The threshold δ (i.e 0.003) is kept low so that no related concept would be eliminated.

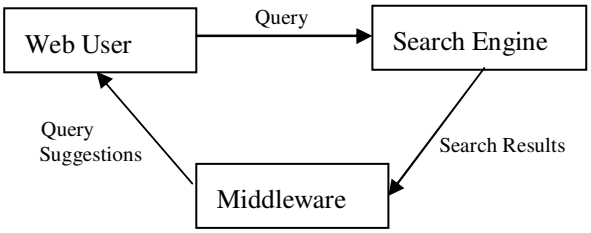


Figure 1. System Model

We have separated the concepts into a number of categories depending on the relation among them. To relate these concepts we have used the correlation values which are greater than some threshold. In our experiment threshold value is taken 0.05.

To evaluate the performance of the proposed method, we need queries, snippets as search results and corresponding URLs. As far as we know, there is no standard dataset, which contains URLs along with the snippet for the queries, is available. Therefore, we have collected the dataset to evaluate the performance of our approach. The detail statistics such as number of users involved, number of queries considered, number of URLs retrieved etc. about the collected dataset is given (see Table 3). The full data set can be made available on request. The dataset consists of queries, corresponding top 100 document snippets and respective URLs returned by the Google search engine. The size of the dataset is approximately 29,46,498 bytes. We have considered different type of queries such as specific (one meaning), ambiguous (multiple meaning), and general queries. The list of domains for all type of queries collected is presented (see Table 2). Sample queries of different type and statistics about them are given (see Table 1). Weight

Query Type	Sample queries	TQC
General	career counselling, code, tax, world wide web, guide etc.	14
Specific	apple fruit, jaguar animal, guitar, Income tax, sony digital camera etc.	17
Ambiguous	Blackberry, apple, jaguar, cell, bow, home etc.	8

Table 1: Sample Queries of each type

precisions (P) are calculated individually for every query from all three categories. For this all concepts are arranged by weight values. Precision is obtained by calculating the ratio (number of related concepts/number of total concepts (m)) for top m concepts. The concepts are marked related or unrelated manually by the cumulative feedback of 12 users who were involved in the experiment.

Category No.	Category Description
1.	Fruits
2.	Computer software
3.	Computer hardware
4.	Mobile Phone
5.	Scientist
6.	Animal
7.	Digital Cameras
8.	Sports
9.	World wide web
10.	Business
11.	Travel
12.	Conference
13.	News
14.	Biology
15.	Career

Table 2: List of domains considered for queries

In the experimental study, the proposed method is applied on the dataset considered to make comparative study of relevance between queries and associated concepts (see Section 3.2) and to perform query recommendations (see Section 3.3).

Number of users	12
Maximum Number of queries assigned to each user	5
Number of test queries	48
Number of unique queries	39
Maximum number of retrieved URLs for a query	100
Maximum number of extracted concepts for a query	718
Minimum number of extracted concepts for a query	309
Number of URLs retrieved	3900
Total Number of concepts retrieved	22963

Table 3: Statistics of the Data Collected

3.2 Experimental Results

In this section, firstly we will present the performance analysis of the proposed weights and then we will be presenting the results of query recommendation using proposed method.

As discussed earlier (see Section 2.2), we have used four weight measures to find the relevance between queries and respected extracted concepts. Now, we will present the comparison of the performance of the proposed weights W1 and W2 with two other weights support (S) and TFIDF. All four weight values are normalized in the range [0,1] for comparison.

For the first set of experiments, we have compared the precision of weights W1 and W2 with that of S and TFIDF calculated individually for every query from three categories i.e. “General”, “Specific” and “Ambiguous”. The precision plots for few sample queries from each category are presented (see Figures 2-4). Precision is plotted against different values of m (top m concepts).

Firstly, we will discuss the results of ambiguous queries. Few of them are presented in the Figure 2. The performances of the two measures W1and W2 are much better than the measure support for all the ambiguous queries. The weight TFIDF is also performing better than support except the one query which is “Blackberry”. In case of a query is too ambiguous, (having more than one well known meanings) W2 is best among all e.g. query “Apple”, “Blackberry” etc. This implies that W2 is capable of identifying the relevant concepts if a query is having multiple meanings. In some cases, where more than 70% of the documents are related to the most popular meaning of the ambiguous query, W1 shows better performance than the W2 e.g. “Jaguar” etc. otherwise performs equally well as W2.

In case of specific type of queries, support is behaving as badly as in ambiguous type. In most of the cases the curve is increasing which shows that it is not capable of capturing the relevance between query and concepts. We have applied these measures on 17 unique queries. It is found that the values of precision for both W1 and W2 are almost same (E.g. query “World Wide Web Conference”, “Soccer” etc.). Moreover, W1 and W2 attain highest values of precision among all measure for all queries in this category. The precision of TFIDF is sometimes approximately equal to W1 and W2 e.g. query “World Wide Web Conference”, “Tennis” etc., sometimes it is lowest e.g. query “Apple Fruit”, “Sony Digital Camera” etc, and sometimes between support and W1or W2 e.g. query “Soccer”, “Income Tax” etc.(see Figure 3).

In the general category of the queries, precisions for all measures are calculated and plotted. Some of which are given (see Figure 4). The same behaviour is found for the weight support. The precision values for W1, W2 and TFIDF are quite close in many queries. W2 is performing exceptionally well (attains high values close to 1) in the queries like “Guide”, “News” etc. This is because; the concepts related to these queries are popular and so as the linked documents containing these concepts. The proposed weight W2 can capture this phenomenon well.

The second set of experiments is conducted to estimate the performance of the measures for each category of the queries. The MAPs for the measures have been calculated for query categories “General”, “Specific” and “Ambiguous” and shown (see Figure 5). The measure support not only acquires the less precision (for all categories of queries) as compared to other measures but also its graph is increasing for general and ambiguous queries. This indicates that the measure is not able to capture the high relevance of concepts with queries for its higher values.

The precision curves of the other three weights are almost same, showing the capability of identifying the relevant concepts for “General” category of the queries (see Figure 5a). However, the values of W2 are higher than the values of W1 and values of W1 are higher than values of TFIDF. The weight TFIDF is low as compared

to W1 and W2 for the other two categories "Specific" and "Ambiguous". This indicates that if the query is specific or ambiguous the two weights W1 and W2 works better to identify relevant concepts. Both the measures W1 and W2 are performing distinguishably well in case of other two categories: "Specific" and "General". It can be seen that W1 is performing slightly better than W2 in the "Specific" category whereas W2 is better for "Ambiguous" category.

Lastly the experiments are performed for the overall behaviour of the measures. The MAP of all weights for all the 39 unique queries are calculated and presented (see Figure 6). It is clear from the figure that the MAP for W2 is highest among all and attains value around 0.9 till top 300 concepts. However, the MAP of W1is slightly lower than that of W2 but having almost same figure as W2. This shows that W2 and W1 weights are capable of identifying the relevant concepts. Moreover, it is observed that the concepts, identified by both W1 and W2, among top 150 are more relevant to the queries than the concepts from 150 to 300 approximately. Furthermore, the measure TFIDF is also showing the same behaviour that is capable of finding the relevance but with the lower precision. The measure support is not at all comparable to the above three measures because 1) Precision is quite low. 2) Curve is increasing rather than decreasing with m.

Now, we will illustrate the results of the query recommendation using the proposed approach. In the first step of the approach, concepts are extracted and then filtered in the second step of the method using the weight W1. After identification of relevant concepts for the given query, we build categories of the filtered relevant concepts for query recommendation. For this, we have proposed a simple model which includes concept categorization using correlation and then query recommendation based on the obtained categories. The correlation threshold used for the experiment is 0.05. We kept it low in order not to miss any category. However, the low threshold may include some noise in the category such as the concept "Green Apple Experience" is present in the category "Fruit" but it is related to tourism. "We have done concept categorization for some queries (see Table 4). The number of categories is ranging from one to many. The number of categories is depending on the query type. For the ambiguous query, the number of categories is large as nine in case of query "Cell" 6 in case of "Apple" etc. On the other hand, this number is as small as one in case of specific queries as "Sony digital Camera". In this paper, we have labelled the concept categories manually. User will be given concept categories as query recommendation which will help users to decide the intent of the query. For example, if user's intent from the query "Cell" is cell phone, then user may select the category "cell phone". Similarly, for the query "Apple" category of interest can be "Apple Tourism". The concepts under the category will help search engine to bring the related documents as a search result.

4 CONCLUSION AND FUTURE WORK

In this paper, we have proposed an approach for query recommendation that builds categories dynamically based on the concepts lying in the web snippets. These categories are dynamically built and manually labelled after forming the groups of relevant concepts. We have proposed two weight functions for eliminating the irrelevant concepts which are extracted by the concept extraction process. The performance of the weight functions are compared and tested on many queries of different types against the two popular measures: support and TFIDF. The performances of the two measures W1 and W2 are outstanding for all queries of different type. The overall performance of the weight W2 is better than the weight W1. However, calculating W1 is a light weight process. It is clear from the experimental results that the proposed method for query recommendation is able to identify all the categories which are possible for a query to belong.

The method which we have described in this paper is simple and able to provide an ease to web users to build a proper search query with the knowledge domain terminology which will help search engine to get the desired results.

The approach can be extended to cluster the documents returned by the search engine and to cluster search engine queries. Category labelling can be done automatically and can be used in clustering of queries and automatic hierarchy building.

5 REFERENCES

[1] Aris Anagnostopoulos, Luca Becchetti, Carlos Castillo, and Aris Gionis(2009): An optimization framework for query recommendation. In Proceedings of the ACM International Conference on Web Search and Data Mining. ACM.

[2] D. Beeferman and A. Berger(2000): "Agglomerative Clustering of a Search Engine Query Log," Proc. ACM SIGKDD.

[3] P. Boldi, F. Bonchi, C. Castillo, D. Donato, and S.Vigna(2009): Query suggestions using query-flow graphs. In WSCD '09: Proc. of the 2009 workshop on *Web Search Click Data,* New York, USA, ACM, 56–63.

[4] S. Chuang and L. Chien(2003):Automatic Query Taxonomy Generation for Information Retrieval Applications, Online Information Rev., 27(4): 243-255.

[5] Dou Shen, Rong Pan, Jian-Tao Sun, Jeffrey J. Pan, Kangheng Wu, Jie Yin and Qiang Yang(2006): Query Enrichment for Web-query Classification. *ACM Transactions on Information Systems* (TOIS). 24(3): 1-33.

[6] W. Frakes and R. Baeza-Yates: Information Retrieval: Data Structures & Algorithms. Prentice-Hall.

[7] H.Fang, T. Tao and C.Zhai (2004):A formal study of information retrieval heuristics. Proceedings of the 27th international ACM SIGIR conference, Sheffield, U.K., 49-56.

[8] B. M. Fonseca, P. B Golgher, E. S. De Moura, and N. Ziviani(2003): Using association rules to discovery search engines related queries. In *First Latin American Web Congress (LAWEB' 03)*, Santiago, Chile.

[9] Hamada M.Zahera, Gamal F. El Hady, Waiel.F Abd El-Wahed(2010): Query Recommendation for Improving Search Engine Results. Proceedings of the World Congress on Engineering and Computer Science 2010, San Francisco, USA.

[10] M. Jansen, A. Spink, J. Bateman, and T. Saracevic(1998): Real life information retrieval: a study of user queries on the web. *ACM SIGIR Forum,* 32(1):5-17.

[11] A. N. Langville and C. D. Meyer(2005): Deeper Inside Pagerank, Internet Mathematic J., 1(3): 335-380.

[12] K.W.-T. Leung, W. Ng, and D.L. Lee(2008): Personalized Concept-Based Clustering of Search Engine Queries, IEEE Trans. Knowledge and Data Eng., 20(11):1505-1518.

[13] L. Li, Z. Yang, L. Liu, and M. Kitsuregawa(2008): Query-url bipartite based approach to personalized query recommendation. In AAAI'08, 1189–1194.

[14] J. Wen, J. Nie, and H. Zhang(2001): Clustering user queries of a search engine. In *Proc. at 10th* International World Wide Web Conference, 162–168.

[15] Karen Sparck Jones(1972): A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21

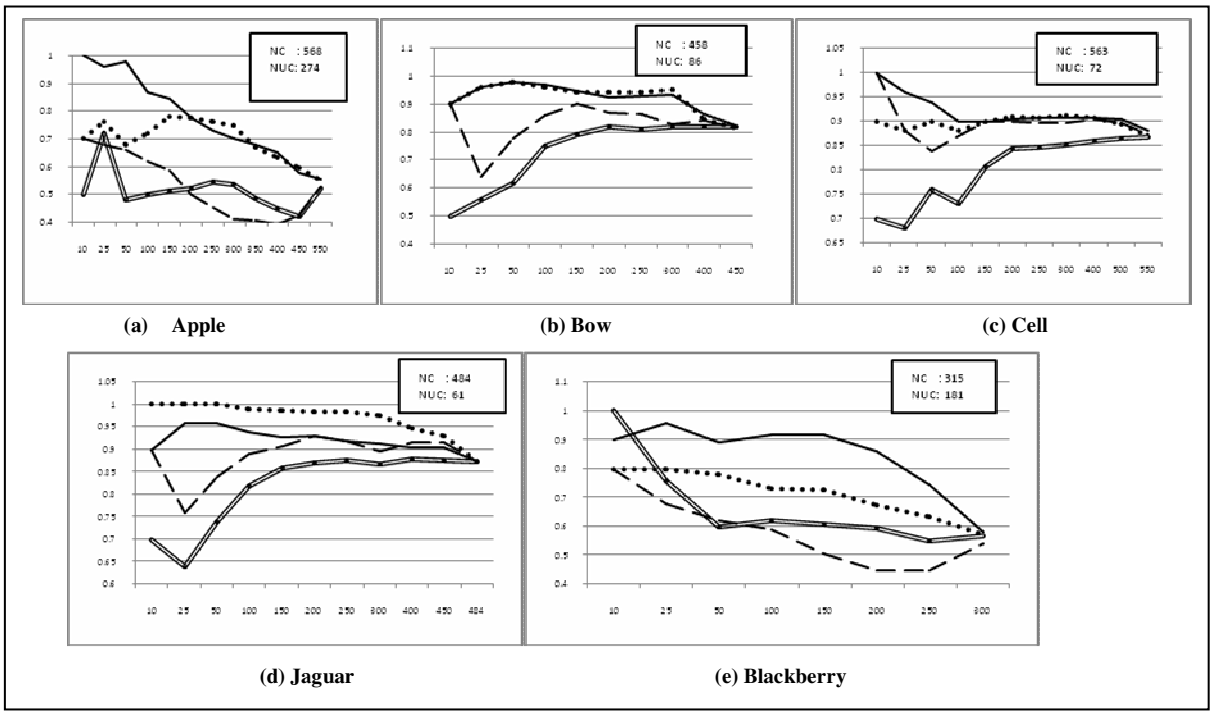
[16] O. R. Zaiane and A. Strilets(2002): Finding similar queries to satisfy searches based on query traces. In Proceedings of the International Workshop on Efficient Web-Based Information *Systems (EWIS)*, Montpellier, France.

[17] Z. Zhang and O. Nasraoui(2006):Mining search engine query logs for query recommendation. Proceedings of the 15th international conference on World Wide Web, New York, NY, USA, 1039–1040, ACM Press.

[18] L.Zhiyuan and S.Maosong (2008): Asymmetrical query recommendation method based on bipartite network resource allocation. WWW 2008,1049-1050.

[19] MontiLingua: http://web.media.mit.edu/~hugo/montylingua/

Figure 2: Precision of W1, W2, S and TFIDF for different queries of type "Ambiguous"
(=== S, -------- TFIDF……. W1, —— W2)

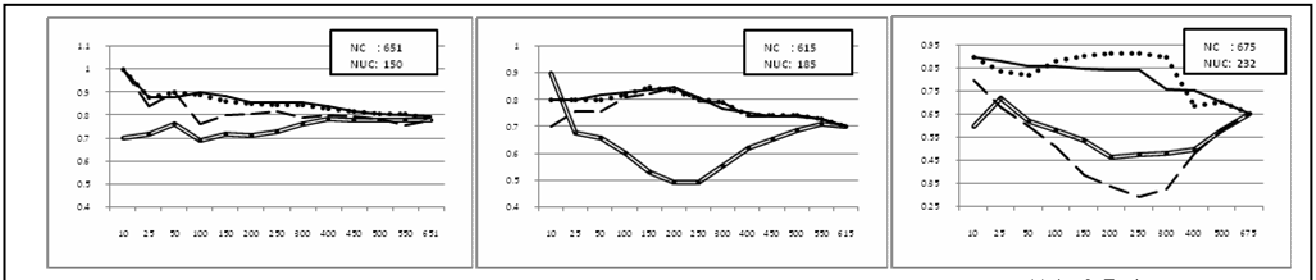


(a) Apple **(b) Bow** **(c) Cell**

(d) Jaguar **(e) Blackberry**

Figure 3: Precision of W1, W2, Support and TFIDF for different queries of type "Specific"

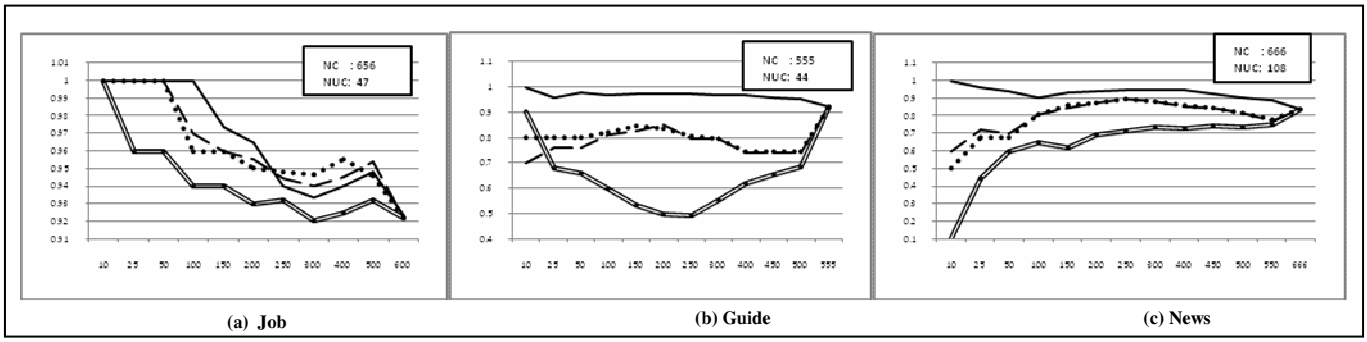
(=== S, -------- TFIDF……. W1, —— W2)



(a) Soccer **(b) WWW Conference** **(c) Apple Fruit**

Figure 4: Precision of W1, W2, Support and TFIDF for different queries of type "General"

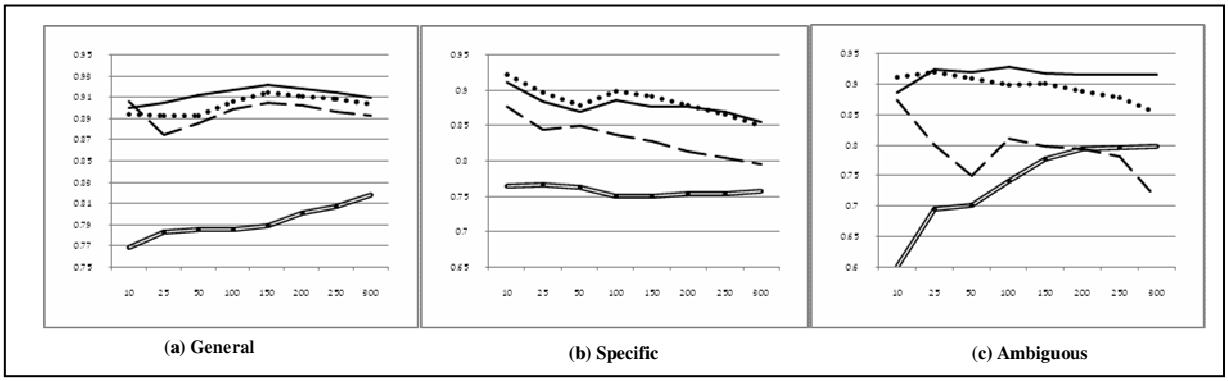
(=== S, -------- TFIDF……. W1, —— W2)



(a) Job (b) Guide (c) News

Figure 5: Mean Average Precision of W1, W2, S and TFIDF for different query types

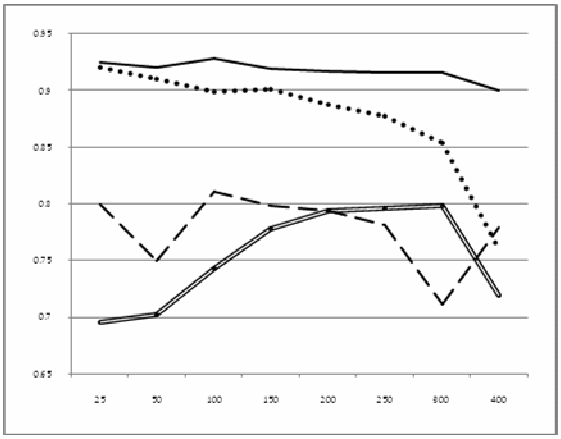
(=== S, -------- TFIDF……. W1, —— W2)



(a) General (b) Specific (c) Ambiguous

Figure 6: Mean Average Precision of measures W1, W2, S and TFIDF over all considered queries

(=== S, -------- TFIDF……. W1, —— W2)



Category name	Sample Concepts Recommended
Apple (Ambiguous type)	
Fruit	Pomaceous fruit, apple tree, Red Apple Chutney Recipe, red apple communications, Marie Helens Apple Cake, Apple cake, Green apple**, red apple, Green Apple Experience* etc.
Apple Products	iPhone, iPod, iPad, Apple Inc, phone, Mac, AAPL, GarageBand, Price, iTunes, Apple II, products, photos, Advertising, iPod Touch, Apple laptop, Apple Mobile Phones, OS X, Android, Music, iStore, MacBooks, Apple View, Apple patents, Apple II, ipod touch, Mac laptop, desktop computers, Apple Products*, Apple Premium Store, basic AAPL stock chart, Blue Apple*, Blue Apple Techno labs ,Big Apple stores*, Blue Apple Technolabs Pvt Ltd*, Store**, Space Apple*, Big apple** etc.

Satellite	Ariane Passenger PayLoad Experiment, Satellites, radio networking, ISRO, Apple II History etc.
Tourism	Apple Tourism, travel habits, people, Apple Valley Resort, Apple Country Resort, Guesthouse Manali India etc.
Industries	Apple rubber industries, rubber and silicone industrial seals etc.
School	Apple Kids International Preschools, Hi-tech International Standard Pre-Schools etc.
Jaguar(Ambiguous type)	
Car	Jaguar Cars, jaguar, Land Rover, Tata Motors Ltd., XJ, XK, Cray XT, prices, XF, Models, News, XF, Jaguar Cars Limited, Jaguar Royal Air Force, CarWale, Jaguar Car Prices**,** Jaguar spare parts, Jaguar Surface Coating Equipments Manufacturer and Exporter, Atari, Jaguar Love, Photos, facts, Atari Jaguar etc.
Animal	Tiger, big cat, wild cat ,Jaguar animal, Tikal National Park, Jaguar Inn Hotel *, Jaguar Jungle hotel *, pictures**, photos** etc.
Mining	Jaguar Mining
Software program	Jaguar Chemistry Software Program, Jaguar Software Solutions, AtariAge, jaguar Download, jaguar software, free downloads, largest Open Source applications and software directory, video game console, Atari Corporation, Atari brand, Atari Museum etc.
Plumbing	Jaguar Plumbing, dependable Arizona Plumber Commercial, Residential Plumbing Specialist
Guide(General type)	
Travel	Guide, Country guide, travelers Guide, Tourism, India Travel Guide, tourist destinations, Trade Directory, Indian Tourism Hotels, Vacations Packages Holidays,Tours & Travel, India tourism and travel information, maps, transport and weather, uttaranchal tourism, heritage india, indian beaches, Wiki travel Open source travel guide, hotels, restaurants, travel tips, sightseeing, tours & travel, beaches, islands, Goa Tourism, Andaman and Nicobar Islands tourism information, OOTY tourism information, Hotel Reviews, Holiday Ideas, Rajasthan Travel Guide, festivals, city guide, Kolkata, Calcutta, Bengal**,** Finance and Investment Guide *, Tax Guide *, Administration & Procedures *, Tax Authorities & Power *, Kids*, Exams guide*, facts**, news*, movies** etc.
Programming	Free Java Tutorials & Guide , Java programming source code, Free java guide website, Java programming, Beginners tutorials, pl sql and sql, java source code, C Programming Guide, Management Study Guide, Free Training Guide, Students Management Study Guide, basics, concepts, management students*, students* etc.
Other guide	Child Labour Guide, One World Child Labour Guide, children
Cell (Ambiguous type)	
Cell biology	Cell Biology, Typical Animal Cell, Cell Therapy, organisms, Cell Science, Stem Cell Treatment, molecular biology, cellpress podcast, PLANT CELL, Cell Models, animal and bacterial cells, Cell Press journal, American Journal, Human Genetics, cell structure, cell cycles, Sickle Cell Disease Association, anatomy, gene expression, Cell Signaling Technology, biotechnology, Plant Cell Anatomy, modern cell theory, microbiology, red blood cells, Diseases & Conditions, biological equipment, International Cell Death Society, Biologists Ltd, Cell Bikes* , E-Cell Project*, Fuel Cells 2000*, starch Amyloplasts*,starchy plants*,tubers and fruits*, Ozone Cell* etc.
Cell phone	Cell Phone, Entrepreneurship Cell, latest cell phones, Free Cell Phone Wallpapers, Cell Phone Videos, Cell phone games and software, Mobile Phone Games Downloads, Free Mobile Fundoo, Free Mobile Java Games, Mobile Theme Download, Free Cell Phone Screensaver, Micromax, Samsung, Karbonn, Onida, powerful Android cell phone, cellone, Figures** etc.
Fuel cell	Fuel cell industry
Recruitment cell	Railway Recruitment Cell, Placement Cell, Indian Air force Placement Cell, IAFPC
Investigation cell	Cyber Crime Investigation Cell Information
Municipal cell	Municipal Reforms Cell
Public private partnership cell	Uttarakhand Public Private Partnership Cell, Uttarakhand PPP Cell
Game	Splinter Cell, Series, Stealth games
Load cell	Crane load cell, pressure sensor, compression load cell
SONY DIGITAL CAMERA (Specific type)	
Sony digital camera	Digital Camera products, Cyber-shot digital and pact cameras, sony SLR digital camera, Wholesale Purchase, Gifting, digital camera mobile phones, best sony digital cameras, lowest price online, Latest Sony digital cameras, sony digital camera price, Sony Reviews, Sony DSC-H10, Sony DSLR-A550 Digital Camera, Express Review, Sony Cyber-shot DSC-TX1 digital camera, Sony Cyber-shot DSC-WX1, Sony Cyber-shot DSC-HX1 Reviews, Sony Cyber-shot-DSC-TX7, new slim and sleek camera, Sony Ericsson K790, Sony Ericsson XPERIA X10, Digital Camera Review Test, Camera Specifications, Online Shopping, India deals, SLR Camera Lenses, Sony SLR Digital Camera, Sony Network Cameras, Sony 1-2 Megapixel Digital Cameras, Sony 3 Megapixel Digital Cameras, Sony 4 Megapixel Digital Cameras, fact, sony handycam hdr-cx550v, Fuji JX250 Digital Camera, Sony W510 Digital Camera, models, digital cameral, maximum resolution, images, quality, digital camera India, cheap price, sony cyber-shot camera accessories, Kodak**, Olympus**, Panasonic**, Fuji**, greatmall**, sify**, Samsung ES20 Digital Camera**, sulekha**, Apple**, Philips**, Sandisk** etc.

Table 4: Sample Queries with corresponding categories and sample concepts

* Concepts not related to the corresponding category i.e. noise introduced during the categorization.

** Concepts grouped under a category due to its popularity i.e. appearing in more documents, but relevant to some other category.

Enhancing Short Text Clustering with Small External Repositories

Henry Petersen **Josiah Poon**

School of Information Technologies,
University of Sydney, NSW 2006, Australia
Email: {hpet9515,josiah}@it.usyd.edu.au

Abstract

The automatic clustering of textual data according to their semantic concepts is a challenging, yet important task. Choosing an appropriate method to apply when clustering text depends on the nature of the documents being analysed. For example, traditional clustering algorithms can struggle to correctly model collections of very short text due to their extremely sparse nature. In recent times, much attention has been directed to finding methods for adequately clustering short text. Many popular approaches employ large, external document repositories, such as Wikipedia or the Open Directory Project, to incorporate additional world knowledge into the clustering process. However the sheer size of many of these external collections can make these techniques difficult or time consuming to apply.

This paper also employs external document collections to aid short text clustering performance. The external collections are referred to in this paper as Background Knowledge. In contrast to most previous literature a separate collection of Background Knowledge is obtained for each short text dataset. However, this Background Knowledge contains several orders of magnitude fewer documents than commonly used repositories like Wikipedia. A simple approach is described where the Background Knowledge is used to re-express short text in terms of a much richer feature space. A discussion of how best to cluster documents in this feature space is presented. A solution is proposed, and an experimental evaluation is performed that demonstrates significant improvement over clustering based on standard metrics with several publicly available datasets represented in the richer feature space.

keywords: Text Mining, Clustering, Short Text, Background Knowledge

1 Introduction

The huge volume of information available through resources such as the world wide web has driven much interest in the clustering and automated analysis of textual data. Most algorithms represent text using a model derived from a bag-of-words. In the bag-of-words model a single feature is created for each word in the corpus and each document is assigned an attribute value for that feature corresponding to the number of occurrences of that word the document.

A fundamental requirement for effective text clustering algorithms is the ability to compare documents according to their semantic content. However for such tasks the use of a bag-of-words representation introduces a number of problems. For realistic document collections, vocabulary sizes in the tens or even hundreds of thousands are not uncommon which can lead to a feature space that is highly sparse. On top of this, issues such as synonymy (different words used to denote the same concept) and polysemy (a single word that can denote multiple concepts) can further degrade the ability of an algorithm to successfully analyse a text collection. These issues are even more pronounced when the text being analysed comprises short strings (ie. documents containing perhaps only a few words each).

A number of researchers have made use of external knowledge repositories in an attempt to extract additional information and compensate for the sparsity of the feature space. Previous literature has reported a wide range of sources for gaining such external knowledge including search engines such as Google (Sahami & Heilman (2006)) as well as linguistic resources like Wordnet (Hotho et al. (2003)).

Recent work in both the supervised and unsupervised literature has explored obtaining external knowledge from large static repositories of text like Wikipedia (Gabrilovich & Markovitch (2007), Hu et al. (2009)) and the Open Directory Project (Gupta & Ratinov (2008)). These approaches have achieved significant success, however they are not without drawback. Because of the very large size of the collections these techniques can be quite time consuming to apply (Phan et al. (2008)). Additionally it has been claimed that for a given text analysis task some consistency between the topic structure of the external knowledge repository and the text collection being analysed (referred to from now on as the target collection) is required in order for the external knowledge to be effective (Phan et al. (2008)). While repositories such as Wikipedia have been shown to be effective for many tasks they are unlikely to be as effective for highly technical or specific problem domains. For such domains the collection of an appropriate, suitably large corpus for use as an external knowledge repository may not always be straightforward.

Some work within the supervised classification literature (Zelikovitz & Hirsh (2001), Zelikovitz & Hirsh (2002), Weng & Poon (2006)) has explored using much smaller external repositories of unlabelled text. These smaller external collections are referred to as Background Knowledge. Their work uses the Background Knowledge to map short text strings into an alternative representation called the Bridging space.

The Background Knowledge need not be drawn from the same distribution as the short text documents, and can differ significantly in length and structure. Additionally, and in contrast to much of the supervised literature the size of the Background Knowledge collection is very small, often only one or two thousand documents in total. However the methods proposed by Zelikovitz et al. for use of this Background Knowledge are specific to supervised classification, and to the best of our knowledge no previous application of such Background Knowledge to document clustering exists.

The following is a brief summary of several interesting contributions provided in this paper:

- Previous clustering literature has often used very large collections such as Wikipedia and the ODP. We demonstrate that small external document collections can still substantially increase short text clustering performance. Furthermore the methods used to exploit these small collections need not be particularly complex.
- To the best of our knowledge all previous applications of Background Knowledge and the Bridging space have focused on supervised classification problems (Weng & Poon (2006), Zelikovitz & Hirsh (2002)). We demonstrate the effectiveness of the Bridging space for short text clustering tasks.
- A function is proposed for use clustering text represented in a Bridging space created using the Background Knowledge. The cluster purity obtained using the proposed function in this feature space is demonstrated experimentally to be substantially better that obtained using several standard similarity functions including the cosine and euclidean distance.

2 Background Knowledge

We now present an explicit definition of Background Knowledge as it is used in this work. An item of Background Knowledge can be any text document semantically relevant to the problem domain of the target corpus. These text documents are unlabelled and no requirement is made that the target and Background documents be drawn from the same distribution. Background documents may be substantially greater in length than the short text in the target collection. The only requirement is that the Background documents be semantically related to the target domain.

This requirement means that identifying an appropriate source of Background Knowledge is a problem specific task. For example given a target collection of short text consisting of a set of technical paper titles we wished to compare according to sub-discipline, the text documents used to create the Background Knowledge could be abstracts or full text from similar papers, excerpts from text books, or even text from relevant mailing lists or forums.

The Background Knowledge collections used in this work are also in general quite small (at most several thousand documents - see section 5.1). This is in contrast to many other algorithms involving static text repositories where the number of external documents is frequently in the hundreds of thousands or even millions (Gabrilovich & Markovitch (2006), Phan et al. (2008)).

2.1 Motivation

Due to the highly sparse nature of short text, it can be very difficult for algorithms to effectively model the co-occurrence structure of a short text collection. Within such a problem domain, it is highly likely that many related words will exist that might never occur together. For example the words 'computer', 'laptop', 'pc' and 'notebook' are all semantically related, however when dealing with short text corpora are unlikely to appear together in a single document.

Each individual Background document in the corpus of Background Knowledge is drawn from one or more latent topics relevant to the clustering task at hand, and the words they contain will be drawn from these topics. Therefore, given a collection of larger Background documents suitably drawn from latent topics semantically related to the problem domain, there is a good possibility that such a pair of related words from the short text will co-occur in the larger documents (Zelikovitz (2002)). As a results of this, additional information on the co-occurrence structure of the problem domain can be obtained from the Background Knowledge (Zelikovitz (2002)).

2.2 Bridging Space

In order to utilise the additional co-occurrence information in the Background Knowledge we map the target documents into a much richer space that will facilitate better comparison between each target instance. This alternate representation is referred to as the 'Bridging Space' (Weng & Poon (2006)). In order to represent a short text in the Bridging space we generate one feature for each document in the Background corpus, and assign to each feature an attribute value equal to the result of the cosine similarity between the short text and the corresponding Background document. More explicitly, given a vector x describing a target document, and a collection of N items of Background Knowledge $B = \{b_1, b_2, \ldots, b_N\}$ expressed over an identical vocabulary, the target document represented in the Bridging space is defined as follows[1]:

$$\hat{x} = \{\hat{x_1}, \hat{x_2}, \ldots, \hat{x_N}\}$$

$$\hat{x}_i = \frac{x \cdot b_i}{\|x\| \times \|b_i\|}$$

Despite two short strings potentially having no common terms, any semantic relationships between them are far more likely to be apparent in the Bridging space. This is because related terms from the two strings are likely to occur together somewhere in the Background Knowledge, producing similar values for the features corresponding to the Background documents in which they co-occur. It is unlikely however that each Background document will describe only a single latent topic. As Background documents almost certainly contain terms drawn from separate (although likely still related) topics, the potential for links based on terms drawn from separate topics may introduce noise when comparing text using the Bridging space. In practice, this effect appears to be is mitigated by selecting a corpus of Background Knowledge sufficiently relevant to the target domain, and by ensuring it contains enough Background documents to

[1] In this work, comparisons between x and each b_i are made using the cosine similarity function. Although we do not do so, other functions could of course be used.

$$euclidean = \sqrt{\sum |x_i - y_i|^2}$$

$$cosine = \frac{x^T \cdot y}{\|x\| \times \|y\|}$$

$$extendedjaccard = \frac{x^T \cdot y}{\|x\|^2 + \|y\|^2 - x^T \cdot y}$$

Figure 1: Several common similarity and distance functions for two vectors x and y

overcome the noise (the ideal number of Background documents is likely to be specific to the domain at hand, however further investigation is left for future work).

The Bridging space has been used several times previously within the literature on supervised learning (Zelikovitz & Hirsh (2001), Zelikovitz & Hirsh (2002), Zelikovitz & Hirsh (2005), Chan et al. (2006), Weng & Poon (2006)). The work presented in this paper is novel however in that it describes the first application of the Bridging space to unsupervised tasks (several of the prior approaches could be adapted to unsupervised learning but their use in such scenarios is potentially sub-optimal. These methods are discussed in more detail in section 4). Additionally to the best of our knowledge there has been no previous treatment of how documents represented in the Bridging space should be compared. We provide such an analysis in the following section.

Finally although it is the case for all Background collections used in this work there is no reason to indicate that Background Knowledge needs to be formed from documents drawn from a single source. In fact such Background Knowledge has previously been used effectively in the supervised literature concerning the Bridging space (Zelikovitz & Kogan (2006)) although further examination of such Background Knowledge for our purposes is left for future work.

3 Clustering in the Bridging Space

When clustering a collection of text, we typically try to find a solution that maximises the intra-cluster and minimises inter-cluster values of some measure of relatedness between instances over the entire data. Within the literature, a wide range of similarity and distance functions have been employed for this purpose with a variety of applications. Several widely used functions are shown in Figure 1. It is well known that certain functions produce better performance with different types of data (Joydeep et al. (2000)). This section presents a discussion of what properties a good function for measuring how well two instances belong in the same cluster should possess. We then propose a clustering function that, when applied over a clustering of short texts represented in the Bridging space will provide a good measure of the overall quality of the solution.

Given a short text collection and associated corpus of Background Knowledge, each Background document can be said to describe some combination of latent concepts from the problem domain. Each document may describe multiple concepts and an individual latent concept may also appear in multiple Background documents. Representing a text document in the Bridging space then describes that snippet in terms of a set of similarities between the snippet and groups of latent concepts.

While the bridging space used in this work is similar in many ways to previous approaches such as Explicit Semantic Analysis (ESA) (Gabrilovich & Markovitch (2006), Gabrilovich & Markovitch (2007)) there are some differences due to the type of external text corpora employed. For example, in their paper on ESA, Gabrilovich & Markovitch (2007) use Wikipedia articles to form the Background Knowledge. They note that each article will describe a single topic, each of which is "Explicitly defined and described by humans", which is not the case for the work described in this paper. Each Background document we employ may reference several latent concepts, which in turn may appear in one or more documents throughout the Background Knowledge corpus. The requirements for an optimal clustering function when using Background Knowledge may indeed be different therefore to such previous work.

We start by stating the following desirable property that a good clustering function for use in the Bridging space should possess:

Proposition 1: *The value produced by the function for any pair of short texts should depend only on features with non-zero similarity to both documents (ie. the short texts both share at least one term with the corresponding Background document).*

As previously stated, each Background document is understood to describe a combination of latent semantic topics from the problem domain. It is reasonable to assume that for a given corpus of Background Knowledge some topics will occur more frequently than others. If a given set of topics is over represented in the Background Knowledge then a comparison function that considers features with zero similarity may adversely effect comparisons between snippets unrelated to those topics. In this way using such a comparison function helps compensate for topic bias in the Background Knowledge.

Additionally, given that a single Background document is unlikely to contain all terms relevant to the topics it describes, a similarity function conforming to proposition 1 has the added benefit of guarding against a short string being erroneously adjudged as not related to a relevant topic.

None of the standard comparison functions described in Figure 1 conform to proposition 1. For example when comparing two snippets the Euclidean function (which is based on the difference between attribute values) will treat Background documents sharing no terms with the two snippets identically to Background documents that share many with both. This clearly violates the desired property that Background documents sharing no common terms with either snippet be discarded. The cosine similarity and extended jaccard coefficient also do not have this property as can be seen from their equations in Figure 1; a Background document that shares terms in common with only one snippet will increase the length of the vectors (and therefore the denominators) with out increasing the dot-product (leaving the numerator unchanged).

We now present a proposed function for use within the Bridging space that contains the aforementioned

desirable property. Let x and y denote two vectors describing strings represented in a Bridging space with N features, and x_i and y_i reference the i^{th} attribute of the strings x and y respectively. We then compute the vector corresponding to the the element wise product of x and y:

$$EWP = \{ewp_1, ewp_2, \ldots, ewp_N\}, ewp_i = 1 - x_i \cdot y_i \quad (1)$$

Let π define an ordering on EWP such that:

$$ewp_{\pi(i)} \leq ewp_{\pi(j)} \forall i < j$$

For two documents x and y, the proposed clustering function is then defined as:

$$bridge(x, y) = 1 - \prod_{i=1}^{k} ewp_{\pi(i)} \quad (2)$$

In order to satisfy proposition 1 the proposed function operates on the product of individual attribute values. Observe that the vector computed in (1) will produce a value of 1 iff either x or y are 0 and a value in the range [0,1) otherwise (assuming of course that both x and y are in the range [0,1]). Note also that the bridge function presented in (2) does not employ an operator such as the sum, but instead uses the product operator to combine the values. Because of this, proposition 1 is upheld as the values of one in EWP will not impact the computed score.

Note that the sum operator is not appropriate in this case. For two document vectors, the sum of the product of the attribute values is proportional to the angle between the vectors. However in order to satisfy property one we consider only the features for which both x and y are non-zero. Because of this, in practice the angle between the two vectors would be very small and lead to much less variation in values between different pairs of documents.

The proposed function also contains several other desirable properties. The value k in equation (2) controls the number of Background documents that are taken into account in measuring similarity between pieces of text. Recall that some topics can be expected to occur more frequently in the Background Knowledge than others. If this imbalance is large enough, it is possible that the results of comparisons related to these topics could be unfairly inflated. Capping the number of Background documents in this manner helps to introduce some tolerance for this problem; only the top k links through the Background Knowledge are considered which helps to alleviate over-representation of topics in the corpus. In practice, the value used for k is identified as a parameter of the algorithm.

The proposed function will also produce lower values for pairs of snippets that share terms with less than k Background documents. Consider two snippets x and y that have a cosine similarity of 0.2 to 10 Background documents. The function presented in 2 will return a similarity value for these snippets of 0.8926. However if x and y have a cosine similarity of 0.2 to only 5 Background documents, the function will return 0.6723. This property is desirable as having a larger number of links through the Background Knowledge provides confidence that two snippets are in fact related and the result is not merely anomalous.

Choosing a good value for the parameter k involves ensuring a reasonable number of Background documents will be considered while not unfairly penalising snippets related to rarer but still relevant topics in the Background Knowledge. In this work we use a k value 10 for all experiments. We note however that the optimal value will likely depend on the target collection and Background Knowledge, although a more detailed evaluation is left for later work.

$$M = \begin{pmatrix} d_1 & d_2 & d_3 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$BG = \begin{pmatrix} b_1 & b_2 & b_3 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\hat{M} = \begin{pmatrix} d_1 & d_2 & d_3 \\ \frac{\sqrt{3}}{2} & 0 & \frac{3}{4} \\ 0 & \frac{\sqrt{3}}{2} & \frac{1}{2} \\ \frac{1}{3\sqrt{2}} & \frac{1}{3\sqrt{2}} & \frac{1}{2\sqrt{6}} \end{pmatrix}$$

Figure 2: Sample term document matrices for target and Background document collections, along with the target collection represented in the Bridging space

By maximising the total combined values of (2) between elements in each cluster, we expect to generate clusters with a high likelihood of describing similar sets of latent topics from the Background Knowledge. However, when considered purely as a direct measure of similarity between pairs of short texts, (2) appears to possess some interesting properties. When used as a measure of distance, (2) is non-metric as it does not obey the triangle inequality, nor is the distance between an element and itself guaranteed to be 0. To prove these claims, consider figure 2 which shows the attribute values for three short text snippets as well as a Background Knowledge corpus with three documents. Figure 3 shows the EWP vectors for each pair of documents along with a distance matrix corresponding to 1 minus the values of 2 for each pair of documents using a k value of 1.

Theorem 1. *When treated as a measure of distance, the proposed function does not obey the triangle inequality.*

Proof. The value produced by the proposed function between documents d_1 and d_2 is 0.9444. This value is greater than the sum of values between documents d_1 and d_3 and documents d_2 and d_3 ($0.3505 + 0.5670 =$

$$EWP = \begin{pmatrix} & b_1 & b_2 & b_3 \\ d_1, d_1 & 0.25 & 1 & 0.9444 \\ d_1, d_2 & 1 & 1 & 0.9444 \\ d_1, d_3 & 0.3505 & 1 & 0.9519 \\ d_2, d_2 & 1 & 0.25 & 0.9444 \\ d_2, d_3 & 1 & 0.5670 & 0.9519 \\ d_3, d_3 & 0.375 & 0.75 & 0.9583 \end{pmatrix}$$

$$OneMinusBridge = \begin{pmatrix} & d_1 & d_2 & d_3 \\ d_1 & 0.25 & 0.9444 & 0.3505 \\ d_2 & 0.9444 & 0.25 & 0.5670 \\ d_3 & 0.3505 & 0.5670 & 0.375 \end{pmatrix}$$

Figure 3: EWP vectors and similarity matrix for Figure 2

$0.9175 < 0.9444$), and thus the proposed distance function does not obey the triangle inequality. □

The notion of semantic similarity has long been known to not follow the triangle inequality (Tversky (1977)). We can observe this by considering the similarity between the terms $Trees$, $Flowers$, and $Chocolates$. It can be seen that the word pairs $(Trees, Flowers)$ and $(Flowers, Chocolates)$ have a high similarity (they describe plants and gifts respectively), however this is not the case for the pair $(Trees, Chocolates)$. That the proposed function does not obey the triangle inequality is therefore not a problem.

Theorem 2. *The distance between an element and itself is not always 0. Furthermore it may not be minimal.*

Proof. From the matrix in 3, the values between documents d_1, d_2 and d_3 and themselves are respectively 0.25, 0.25 and 0.375. It can therefore be seen that when using the proposed function as a measure of distance, the distance from an element to itself will not necessarily be 0. That the distance between an element and itself can be greater than it is to another element can be demonstrated by observing that $bridge(d_1, d_3) = 0.3505 < 0.375 = bridge(d_3, d_3)$. □

Recall that each Background document describes a number of latent topics from the problem domain, and that when represented in the bridging space an individual attribute can be considered to describe the similarity between a short text string and the topics described in a piece of background knowledge. It follows then that when applied to two short text snippets, the function described in (2) can be regarded as a combination of the similarities between the two snippets and the set of latent topics described in k pieces of background knowledge. In other words, rather than producing a value that directly compares the two snippets, the function will produce a value comparing their similarity to some set of latent topics.

Consider the example from Figure 2 discussed above in Theorem 2. All three terms in d_1 are present in the background document b_1, which implies that it is very likely d_1 is related to the latent topics described in b_1. The short string d_3 also shares three terms with b_1, however d_3 also contains an additional term not contained in b_1. This could be considered to reduce the likelihood that d_3 is related to the latent topics described in b_1. The additional term means that the similarity between d_3 and b_1 will be less that that of d_1 and b_1. This leads to the function in (2) indicating d_1 and d_3 have a greater similarity to a common set of topics in the Background Knowledge than it does with d_3 and d_3.

Although these properties imply that the clustering function proposed in (2) is less than ideal for directly comparing individual pairs of short strings, it is still reasonable to produce a clustering based on optimising (2) over a collection of documents. Although local discrepancies may exist, such a clustering would still tend to group documents related to similar latent topics. Providing the Background Knowledge adequately describes an appropriate set of latent topics, the effect of any local inconsistencies should be outweighed. In the following sections we demonstrate the effectiveness of clustering using (2) in the Bridging space compared to optimising clusters based on other functions.

The proposed function in 2 bears some similarity to the supervised Bridging algorithm (Zelikovitz & Hirsh (2002)) (see section 4 for further detail). Again however, the supervised Bridging algorithm relies on the presence of labelled training data to function, and as opposed to comparing text directly it instead measures similarity between text and each class label in the data set.

4 Related Work

Clustering short text based on semantic similarity is a problem that has seen much interest in recent times. A wide range of approaches have been proposed to compensate for the difficult nature of the task which in turn can be broadly divided into two categories; internal and external.

Internal analysis techniques are those that attempt to discover the semantic relationships between individual terms through statistical analysis of the target document collection. They consider no additional knowledge repositories. This class of approach includes techniques like Latent Semantic Indexing (LSI)(Deerwester et al. (1990)), Probabilistic Latent Semantic Analysis (pLSA) (Hofmann (1999)), and Latent Dirichlet Allocation (Blei et al. (2003)). While these approaches have proved successful in many cases, their effective application can be difficult where the target collection is extremely sparse or there are insufficient instances to adequately model the problem domain. For brevity's sake we do not discuss them further here and the interested reader is directed to the appropriate literature.

The second class of solution for measuring semantic similarity between pairs of short text involves the application of additional data not available in the original dataset (hence we refer to such approaches as external).

A popular approach within the literature has been the application of lexical resources such as WordNet (Miller (1995)) to aid in the comparison of textual data. Wordnet provides a manually annotated lexical database of the English language, and was originally created by G. Miller at Princeton university. By taking advantage of the semantic relationships expressed between terms in Wordnet, several methods have been proposed for compensating issues of semantic ambiguity when comparing text (Hotho et al. (2003), Jing et al. (2006), Li et al. (2008)). One drawback to these methods is that the creation and maintenance of such resources can be very expensive, and obtaining a suitable resource may be difficult for some domains.

Several previous works (ie. Sahami & Heilman

Table 1: Summary of the datasets used.

Dataset	# Docs	Avg. Doc. Length	# Classes	# BG Docs.	Avg. BG. Doc. Length
2CNews	1033	6.02	2	1165	56.92
2CPhys	953	5.82	2	1531	74.46
3CPhys	1066	5.80	3	1702	72.69
7CNetv	1723	2.53	7	1160	103.31

(2006), Yih & Meek (2007), and Bollegala et al. (2007)) propose methods to employ the results of Google searches on short text strings to measure their similarity. While such algorithms have proven effective for suitably short text, they are inappropriate for application to longer documents. This is due to the algorithms' use of the target short text snippet as Google queries. Our approach has no such limitation.

Some researchers (ie. Gabrilovich & Markovitch (2007), Banerjee et al. (2007), Hu et al. (2008), Hu et al. (2009), and Phan et al. (2008)) have made use of large static repositories of text such as Wikipedia and the Open Directory Project (ODP). One issue with such methods however is that the sheer number of documents in these collections (often hundreds of thousands or even millions) can lead to serious issues with regards to the processing time required (Phan et al. (2008)). As well as this many of the more general sources like Wikipedia are unlikely to be optimal for highly technical domains, and finding a suitably large corpus for such problems is not straightforward. This class of approach shares some similarities with the one presented in this paper as both employ static text collections to obtain additional domain knowledge, although ours differs in that that the collections used are of a much smaller size.

Within the clustering literature work exists where additional domain knowledge is added through placing constraints on the output of the algorithm. For example, in the paper by Wagstaff et al. (2001) an expert user is employed to annotate part of the target dataset with must-link and cannot-link constraints, and an extension to the k-means algorithm is provided to utilise this information and significantly improve the resultant clustering. These approaches differ from the other external methods presented above in that they do not address the problem of measuring similarity between individual instances, rather they modify the algorithm applied to analyse a given dataset. Such methods suffer from the fact that the manual annotation by an expert user required for their application is not always feasible, even if such an expert user exists.

The Bridging space used in this work has been seen previously in the supervised text classification literature. The Bridging space was first used implicitly in the work of Zelikovitz & Hirsh (2001) in their nearest neighbour Bridging approach. Given a set $(t_i, l_i) \in Tr$ of N labelled training documents, a set $b_j \in Bg$ of M Background documents, a query string q, a similarity function sim and an integer value k the nearest neighbour Bridging algorithm works by first computing the length $N \times M$ vector of 2-tuples:

$$\{(1 - sim(t_i, q) \times sim(b_i, q), l_i)\}$$

The tuples are then ordered increasing according to their first value, and all but the first k are discarded. Each class is then assigned a score equal to one minus the product of the first value for all remaining tuples with that class label. The query is finally assigned the label corresponding to the largest such value[2].

The supervised Bridging algorithm (Zelikovitz & Hirsh (2002)) is in some respects very similar to the approach proposed in this work. The Bridging algorithm however is only suitable for supervised tasks, as it at no time computes a single similarity value between individual text strings (rather it computes similarities between instances and classes).

The Bridging space has also been used in the literature on classifying imbalanced data (Weng & Poon (2006)). Documents are expressed in the Bridging space defined by the Background Knowledge before standard classification algorithms (in their case a support vector machine) are applied. The work described in by Weng et al. differs from ours in that we propose a method for comparing documents in the Bridging space as an alternative to standard algorithms.

The Bridging space has also been used previously (although it is not referred to as such) in Explicit Semantic Analysis (ESA) (Gabrilovich & Markovitch (2007)) using Wikipedia articles as Background Knowledge. Our work differs in that, like Weng & Poon (2006), the approach in Gabrilovich & Markovitch (2007) compares snippets in the Bridging space using standard similarity metrics. Additionally ESA employs a very large, general repository of Background Knowledge (Wikipedia), while our work employs much smaller, targeted document collections.

Further interesting work using the Bridging space has been performed in Chan et al. (2006) by modifying the Bridging algorithm of Zelikovitz et al. for use with Background Knowledge of the same size and form as the target text snippets (eg. Semi-supervised learning). In their work the authors note a slight deterioration in the performance of the Bridging algorithm when used with Background Knowledge of this type which can be attributed to the decreased generality of the Background documents and the associated domain knowledge they provide. The authors show however that this effect can be offset by employing the Bridging algorithm in conjunction with semi-supervised techniques such as co-training (Blum & Mitchell (1998)) and assigning labels to a portion of the Background Knowledge.

Finally we note one prior use of this type of Background Knowledge that could be applicable to clustering tasks, however our method improves upon this in several ways. Zelikovitz & Hirsh (2001) uses Latent Semantic Analysis on the combined target and Background Knowledge collections, and observe a substantial increase in classification performance on the target documents. This method assumes that both the target and Background collections are drawn from very similar distributions. In fact for supervised problems it has been shown that this method is often outperformed by the nearest neighbour Bridging algorithm for Background Knowledge of a significantly different structure to the target collection (Zelikovitz (2002)). The method proposed in this paper requires

[2]Due to space constraints a more complete description of the supervised nearest neighbour Bridging algorithm is not included. The interested reader is directed to the literature (Zelikovitz & Hirsh (2002))

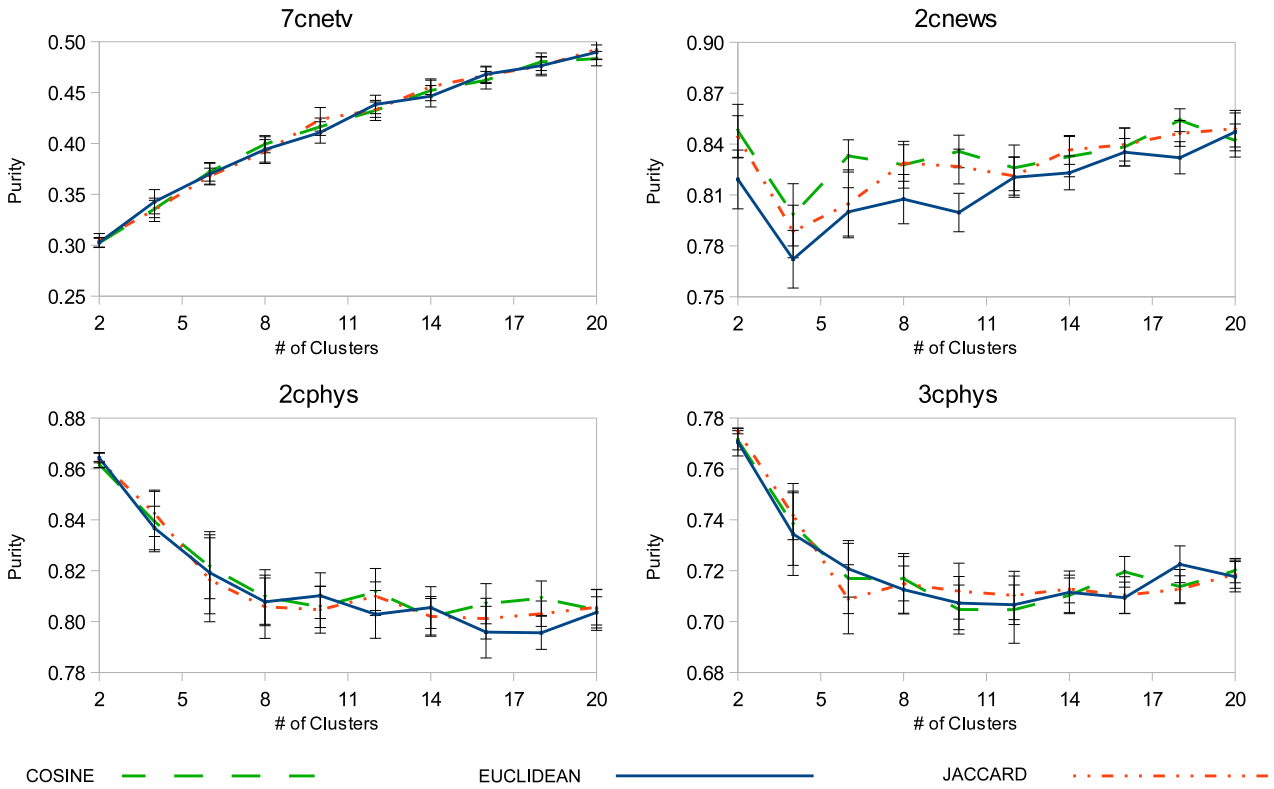


Figure 4: Cluster purity (y-axis) versus number of clusters (x-axis) using the cosine, extended jaccard and euclidean similarity functions in a Bag-of-words space. Graphs are ordered from top to bottom, left to right and present results for the 7CNetv, 2CNews, 2CPhys and 3CPhys datasets respectively.

no such assumption on the Background Knowledge.

5 Experimental Evaluation

In order to evaluate both the Bridging space and the proposed function in terms of their ability to measure similarity between snippets of text, we compare the performance of a number of unsupervised text categorisation tasks both with and without the proposed method. In order to minimise the risk that a negative result is due to a poor choice of Background Knowledge, we perform the evaluation using short text collections for which there is an existing set of Background documents available from the supervised applications of the Bridging space.

The remainder of this section is divided into three parts; a presentation of the datasets employed in our evaluation, a description of the algorithms used to produce the clusters, both in the standard bag-of-words and the Bridging space, and a discussion of the results obtained using our approach.

5.1 Datasets

We now provide a description of each dataset used in this paper. A summary is provided in Table 1. All datasets employed were originally used with Background Knowledge by Zekikovitz et al. (see Zelikovitz (2002)) and are freely available for download[3].

5.1.1 2CNews

The first dataset used is the 2CNews collection, which comprises 1033 news article headings originally published on the ClariNet news site. Each article is labelled as relating to either Business or Sports. The Background Knowledge used is a collection of 1165 partial excerpts from related ClariNet articles that were not included in the test collection.

5.1.2 2CPhys

The second dataset used is a collection of physics technical papers titles. The 2CPhys dataset has a total of 953 titles which are labelled as being related to either astrophysics or condensed matter physics. The Background Knowledge used is 1531 abstracts from other related technical papers.

The 2CPhys dataset provides a more technical problem domain in which to evaluate our proposed approach. The difference in distributions over the classes in 2CPhys is likely to be much more subtle that that for the Business and Sports classes in the 2CNews collection. We expect that this should provide a challenging and interesting task on which to perform our evaluation.

5.1.3 3CPhys

The third dataset used is very similar to 2CPhys in that is is also a collection of titles from physics technical papers. 3CPhys differs however in that it contains 1066 titles, and there are three possible classes (astrophysics, condensed matter physics, and quantum cosmology). The Background Knowledge used with these documents is 1702 abstracts from other related technical papers.

[3] http://www.cs.csi.cuny.edu/~zelikovi/data.htm

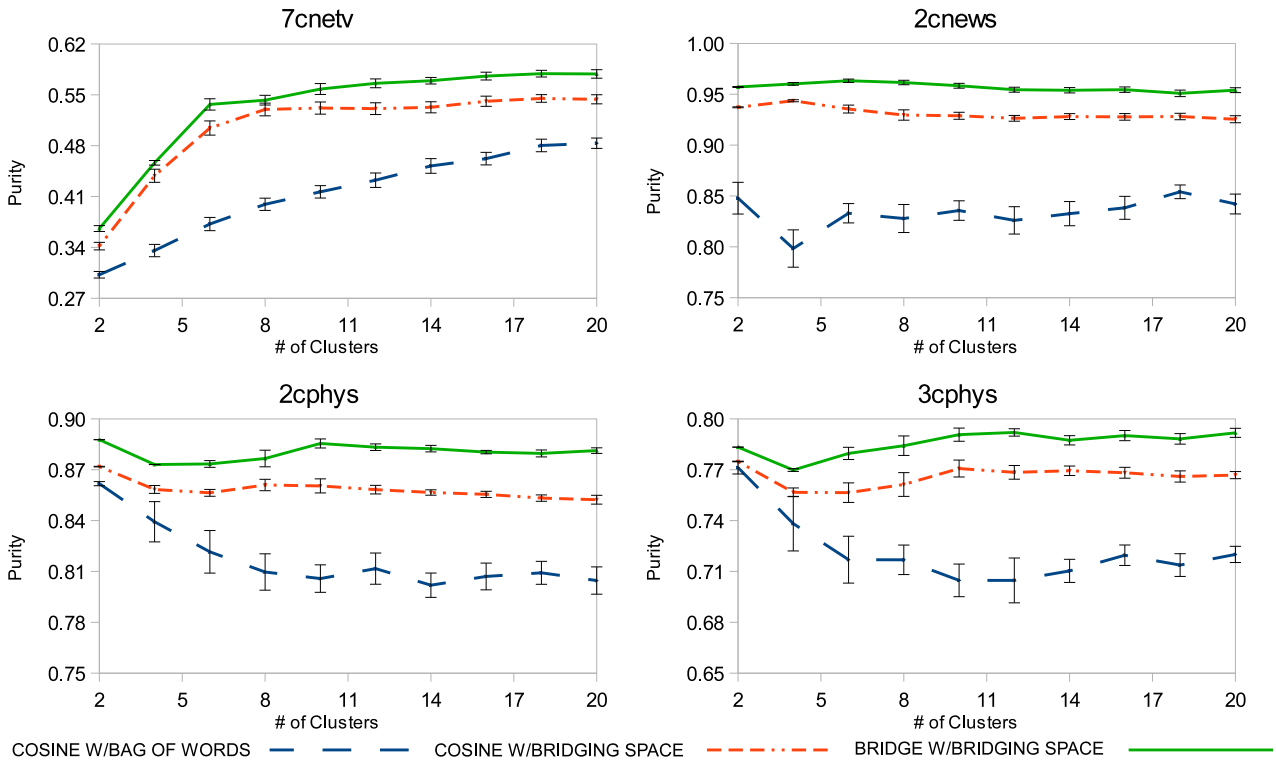


Figure 5: Cluster purity (y-axis) versus number of clusters (x-axis) for the cosine and Bridging functions in the Bridging space. A baseline computed using the cosine similarity in a Bag-of-Words space is also shown. Graphs are ordered from top to bottom, left to right and present results for the 7CNetv, 2CNews, 2CPhys and 3CPhys datasets respectively.

5.1.4 7CNetv

The fourth dataset we use is 7CNetv, web page headings collected from the NetVet website [4]. Each heading is labelled as relating to one of 7 classes; dogs, cats, cows, horses, rodents, primates, and birds. Background Knowledge is created using part of the text from other pages in the NetVet domain. In total there are 1723 test documents and 1160 Background documents. Unlike 2CNews, 2CPhys and 3CPhys the greater number of classes in 7CNetv presents a more complex learning challenge. 7CNetv is also interesting in that the number of Background documents is significantly less than the number of headings.

5.2 Methodology

Document clustering is performed using the freely available CLUTO clustering toolkit (Karypis (2006)). CLUTO is specially designed for use with high dimensional data, and has been used a number of times throughout the text clustering literature (Banerjee et al. (2007), Doucet & Lehtonen (2007), Wang et al. (2008)).

Datasets are input to CLUTO as a matrix of similarity values. In order to construct a similarity matrix for distance functions such as the euclidean distance, we use a value inversely proportional to the equation described in Figure 1, then scale the matrix over the range [0,1]. An objective function f is then selected, along with an algorithm which then attempts to produce the clustering that optimises f over the dataset.

All experiments in this paper were performed using CLUTO's direct clustering algorithm with 10 trial runs. We employed the default objective function which is based on maximising the intra-cluster similarities between instances. All other parameters are left as default settings.

We evaluate the quality of the clustering using the cluster purity evaluation metric (Li et al. (2008), Hu et al. (2008)). All results are averaged over 20 runs and are reported with their 95% confidence interval.

We do not explicitly evaluate any stopping criteria to determine an ideal number of clusters. Instead, for each experiment we vary the number of clusters produced by CLUTO from 2 to 20 (using a step size of 2) and report the purity values over this range of cluster numbers.

The first experiment described in this paper aims to demonstrate that small Background Knowledge collections can be used to significantly increase clustering performance. To the best of our knowledge there have been no previous uses of the datasets described in section 5.1 in the clustering literature. We evaluate the hypothesis by comparing the clustering obtained using the Background Knowledge against a baseline clustering using standard similarity measures in a bag-of-words feature space. Figure 4 shows the purity values obtained using CLUTO with a range of similarity functions and number of output clusters. We observe that the cosine similarity function performs as well or better than all other measures tested with this data. As such we use the cosine similarity function to compute the baseline.

5.3 Results

Figure 5 shows the cluster purity for clustering using the cosine similarity function in both the Bag-of-Words and Bridging spaces for each of the 2CNews, 2CPhys, 3CPhys and 7CNetv datasets. For 2CNews,

[4] http://netvet.wustl.edu

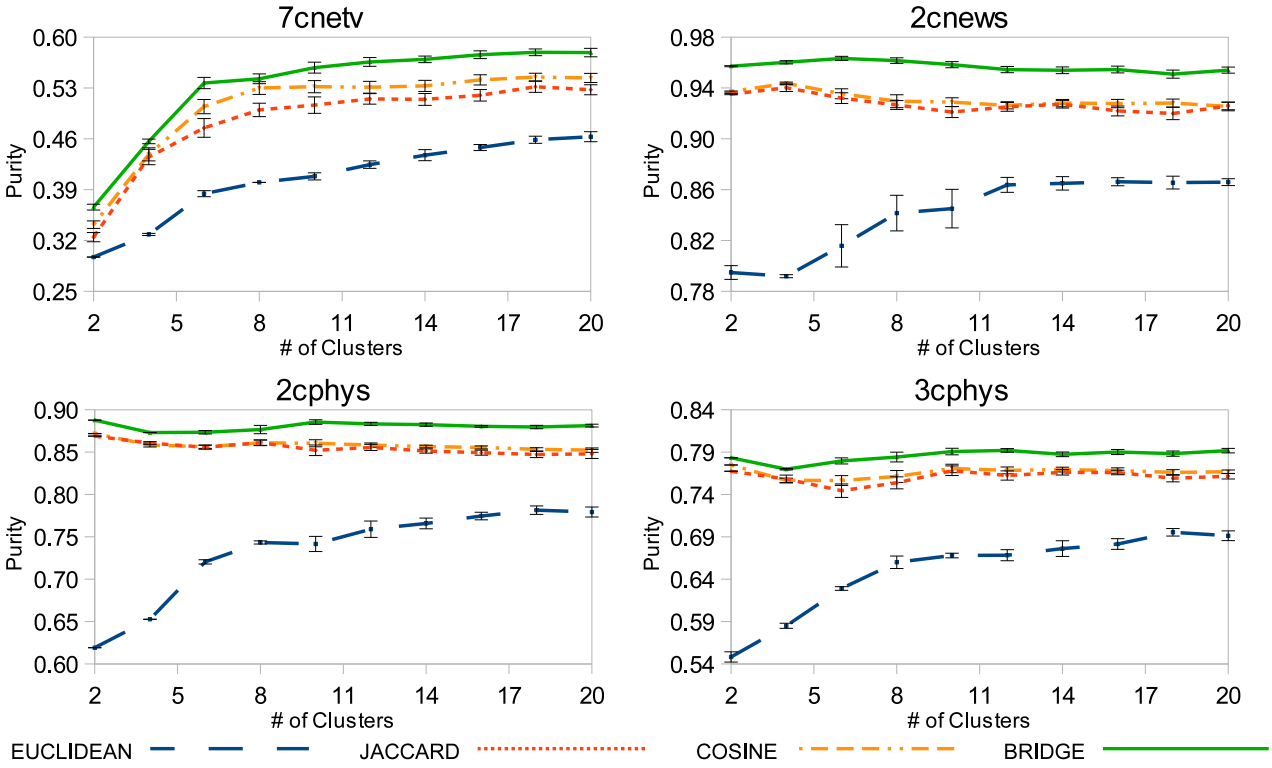


Figure 6: Cluster purity (y-axis) versus number of clusters (x-axis) using the euclidean, extended jaccard, cosine and bridging functions in a Bridging space. Graphs are ordered from top to bottom, left to right and present results for the 7CNetv, 2CNews, 2CPhys and 3CPhys datasets respectively.

7CNetv, and 2CPhys a substantial increase in purity values is observed for all numbers of clusters obtained when using the Background Knowledge to map the target collections into the Bridging space. The purity for the 3CPhys dataset in the Bridging space is also substantially higher than that of the Bag-of-Words representation space for all numbers of clusters tested, with the exception of 2 and 4. We believe this presents a strong case that using Background Knowledge to reexpress text in the Bridging space can improve the ability of clustering algorithms to measure similarity between short text documents.

We note that when clustering the 2CNews, 2CPhys and 3CPhys in the Bridging space measured purity values are relatively consistent as the number of clusters is varied. However with 7CNetv a very sharp increase in purity is observed as the number of clusters is increased from 2 to 6. We hypothesise that this is due to the number of hidden classes in 7CNetv being larger than the number of clusters produced. For example when generating 4 clusters for the 7CNetv dataset, all documents from at least 3 hidden classes will be counted as being incorrectly clustered.

Figure 6 compares the measured purity values obtained using the proposed clustering function to that of the cosine, euclidean, and jaccard functions (see figure 1) when clustering documents in the Bridging space. For all datasets and number of clusters evaluated, we note that the proposed clustering function substantially outperforms all other functions tested. This demonstrates the effectiveness of the proposed function when clustering documents that have been represented in the Bridging space.

We note that when clustering using the proposed function in the Bridging space, the measured purity values for all datasets are consistently greater than those for the best baseline (see figure 5). This provides a strong case for the use of the proposed function on data represented in the Bridging space.

We note the relatively poor performance of clustering using the euclidean distance compared to the cosine, jaccard, and proposed Bridging functions. As the euclidean distance is based on the difference of individual attribute values, euclidean distance will not distinguish between features with identical, high values for each vector (ie. from Background documents that share terms with both short text strings), and features for which both vectors are zero (ie. Background documents that share no terms with either snippet). This behaviour is not shared by the other comparison functions tested in this work, which are based on the product of attribute values and therefore ignore features for which both vectors are zero. Recall also that as the cosine and extended jaccard features are sensitive to the length of the vectors that they will be effected by the features for which only one of the vectors is non-zero. The relative performances of the cosine, euclidean, jaccard, and Bridging functions supports proposition 1 given in section 3; namely that Background documents sharing no terms with one or both of the short text strings should not influence the result of the function.

6 Conclusions

In this paper we have presented a method for leveraging relatively small, unlabelled collections of semantically relevant documents to improve the clustering of short text data. We refer to this unlabelled, relevant text as Background Knowledge. A summary of the novelty and major contributions of this paper is as follows:

- We demonstrate a simple method for using Back-

ground Knowledge to construct an alternative representation for short text called the Bridging space. We show that using Background Knowledge with this method significantly increases cluster purity. While the Bridging space has been used before in supervised document categorisation, to the best of our knowledge this is the first such use for unsupervised clustering.

- Unlike much of the previous literature concerning external document collections, the Background Knowledge corpora we employ are small and contain only a few thousand documents each. The use of simple methods to exploit small Background Knowledge collections is novel. The reduction in the size of the external collections is likely to provide significant practical benefits with regards to obtaining and use of the external corpora.
- We propose a clustering function for use on short text documents represented in the Bridging space. Experimental results have shown this method to be very effective, outperforming standard distance and similarity measures by a substantial margin on four separate document collections.

A number of directions for future work exist. Possible extensions to the work described in this paper include:

- The value of k used in equation (2) was 10 for all experiments reported in this paper. While we have achieved good results with this value setting a more formal evaluation of the ideal value would be of interest.
- Exactly what makes an effective corpus of Background Knowledge is dependent on the specific clustering task at hand. For a given text clustering problem, finding an appropriate set of Background Knowledge is by no means a trivial task. While there has been research in the supervised literature on automatically obtaining Background Knowledge for a given dataset (Zelikovitz & Kogan (2006)), the methods described are specific to supervised learning. Exploring methods for obtaining Background Knowledge for use with unsupervised tasks would be useful.
- The application of Background Knowledge along with the proposed similarity function has been shown to significantly increase the purity when clustering short text documents. An explicit comparison of small and large external document corpora was however not performed. A comparison of small collections of Background Knowledge with alternative sources of external knowledge such as Wikipedia would be interesting.

References

Banerjee, S., Ramanathan, K. & Gupta, A. (2007), Clustering short texts using wikipedia, *in* 'SIGIR'.

Blei, D. M., Ng, A. Y. & Jordan, M. I. (2003), 'Latent dirichlet allocation', *Journal of Machine Learning Research* **3**, 993–1022.

Blum, A. & Mitchell, T. (1998), Combining labeled and unlabeled data with co-training, *in* 'COLT: Proceedings of the Workshop on Computational Learning Theory', pp. 92–100.

Bollegala, D., Matsuo, Y. & Ishizuka, M. (2007), Measuring semantic similarity between words using web search engines, *in* 'WWW'.

Chan, J., Koprinska, I. & Poon, J. (2006), Nearest neighbour classification with background knowledge extended to semi-supervised learning, Technical report, University of Sydney.

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W. & Harshman, R. A. (1990), 'Indexing by latent semantic analysis', *Journal of the American Society of Information Science* **41**(6), 391–407.

Doucet, A. & Lehtonen, M. (2007), Unsupervised classification of text-centric xml document collections, *in* 'INEX'.

Gabrilovich, E. & Markovitch, S. (2006), Overcoming the brittleness bottleneck using wikipedia: enhancing text categorization with encyclopedic knowledge, *in* 'AAAI'.

Gabrilovich, E. & Markovitch, S. (2007), Computing semantic relatedness using wikipedia-based explicit semantic analysis, *in* 'IJCAI'.

Gupta, R. & Ratinov, L. (2008), Text categorization with knowledge transfer from heterogeneous data sources, *in* 'AAAI'.

Hofmann, T. (1999), Probabilistic latent semantic analysis, *in* 'Proc. of Uncertainty in Artificial Intelligence, UAI'99', Stockholm.

Hotho, A., Staab, S. & Stumme, G. (2003), Wordnet improves text document clustering, *in* 'In Proceedings of the SIGIR Semantic Web Workshop'.

Hu, X., Sun, N., Zhang, C. & Chua, T. (2008), Enhancing text clustering by leveraging wikipedia semantics, *in* 'SIGIR'.

Hu, X., Sun, N., Zhang, C. & Chua, T. (2009), Exploiting internal and external semantics for the clustering of short texts using world knowledge, *in* 'CIKM'.

Jing, L., Zhou, L., Ng, M. & Huang, J. (2006), Ontology-based distance measure for text clustering, *in* 'SIAM International Conference on Data Mining'.

Joydeep, A. S., Strehl, E., Ghosh, J. & Mooney, R. (2000), Impact of similarity measures on web-page clustering, *in* 'In Workshop on Artificial Intelligence for Web Search (AAAI 2000)', AAAI, pp. 58–64.

Karypis, G. (2006), Cluto-a clustering toolkit, release 2.1.1, Technical report, Department of Computer Science, University of Minnesota. http://www.cs.umn.edu/ karypis/cluto/.

Li, Y., Chung, S. & Holt, J. (2008), Text document clustering based in frequent word meaning sequences, *in* 'Data and Knowledge Engineering'.

Miller, G. A. (1995), 'Wordnet: A lexical database for english', *Communications of the ACM* **38**(1), 39–41.

Phan, X., Nguyen, L. & Horiguchi, S. (2008), Learning to classify short and sparse text and web with hidden topics from large-scale data collections, *in* 'WWW'.

Sahami, M. & Heilman, T. (2006), A web-based kernel function for measuring the similarity of short text snippets, *in* 'WWW'.

Tversky, A. (1977), 'Features of similarity', *Psychological Review* **84**(4), 327–352.

Wagstaff, K., Cardie, C., Rogers, S. & Schrodl, S. (2001), Constrained k-means clustering with background knowledge, *in* 'ICML', pp. 577–584.

Wang, P., Domeniconi, C. & Hu, J. (2008), Using wikipedia for co-clustering based cross-domain text classification, *in* 'ICDM'.

Weng, G. & Poon, J. (2006), A data complexity analysis on imbalanced datasets and an alternative imbalance recovering strategy, *in* 'WI'.

Yih, W. & Meek, C. (2007), Improving similarity measures for short segments of text, *in* 'AAAI'.

Zelikovitz, S. (2002), Using Background Knowledge to Improve Text Classification, PhD thesis, Rutgers University.

Zelikovitz, S. & Hirsh, H. (2001), Using lsi for text classification in the presence of background knowledge, *in* 'CIKM'.

Zelikovitz, S. & Hirsh, H. (2002), Integrating background knowledge into nearest-neighbor text classification, *in* 'ECCBR'.

Zelikovitz, S. & Hirsh, H. (2005), Improving text classification using em with background text, *in* 'FLAIRS'.

Zelikovitz, S. & Kogan, M. (2006), Using web searches on important words to create background sets for lsi classification, *in* 'FLAIRS'.

Reassembling Multilingual Temporal News Datasets with Incomplete Information

Calum S. Robertson

School of Computer Science and Engineering, The University of New South Wales; Smart Services CRC; Sirca UNSW, Sydney, NSW, Australia 2052

calum.robertson@unsw.edu.au

Abstract

Institutional investors are building increasingly more sophisticated algorithmic trading engines that account for textual as well as numerical information. To train these engines they need large datasets of information with highly accurate timestamps that cover long periods with differing trading conditions. Thus, the demand for temporal news datasets beyond the point where full archives are available is increasing. Rebuilding the actual temporal news dataset that was transmitted to the market relies on merging multiple datasets, each with incomplete information and sometimes questionable quality. Doing so requires near duplicate detection in a very large dataset including news in many languages. This research is novel as in our scenario we are unaware of the language used in any given news article. In this paper we describe a language independent near duplicate detection algorithm and demonstrate its performance on a dataset consisting of tens of millions of news messages in over 20 languages consisting of hundreds of gigabytes of content.

Keywords. News, Near Duplicate Detection.

1. Introduction

Markets are efficient if prices fully reflect all available information: that is prices change randomly until new information causes a price adjustment (Fama 1970). Highly unexpected news (a shock) cause the biggest price changes so the informational efficiency of the market is measured by the time it takes to incorporate this type of news (Latham 1986).

Over the years as technology has improved investors could choose to acquire new information for assets in a wide variety of markets significantly faster. In 1830 the London Stock Exchange began to transmit prices via the electric telegraph (London Stock Exchange 2011) and Reuters began to relay these prices and news via a combination of telegraph cables and carrier pigeons in 1851 (Thomson Reuters 2011a). Reuters started using computers to transmit prices in 1964 and their journalists began transmitting news via computer in 1971.

Theoretically the increasing speed of delivery and the availability of information should have made markets more efficient. Indeed the evidence in the early years overwhelmingly confirmed that real world markets were efficient (Fama 1970). This included the revelation that disregarding periods when news was released to the market led to a slight improvement in the predictability of stock returns (Roll 1988). However, doubts began to arise over the years as notable inefficiencies were discovered (Thaler and De Bondt 1985). Shiller argues that it is incorrect to claim markets are completely efficient due to prominent examples of inefficiency, though it is also invalid to claim that markets are completely inefficient (Shiller 2003).

It should be noted that early research used low frequency prices (i.e., daily, weekly or monthly) primarily due to the difficulty in acquiring high frequency data. In 1987 Reuters launched their Integrated Data Network (IDN) to electronically deliver prices and news, time stamped to the millisecond, globally to their clients at high speed. The reduced delay in delivering content has enabled investors to react faster to new information. This is clearest in examples where information was expected, such as the case where macroeconomic news was found to trigger interest rate and foreign exchange futures market reaction within 10 seconds, with the news being fully incorporated into the price within 50 seconds (Ederington and Lee 1995).

The increasing speed of information delivery has encouraged institutional investors to build increasingly more sophisticated algorithmic trading engines that account for textual as well as numerical information. For example recent research has analysed the content of news to predict abnormal market behaviour using sentiment scores (Tetlock et al. 2008), and machine learning (Mittermayer and Knolmayer 2006). A detailed analysis of work in this area is provided within (Mitra and Mitra 2011).

To train their trading engines institutional investors need large datasets of information with highly accurate timestamps that cover long periods with differing trading conditions. However, the major focus of market data providers is to deliver content as fast as possible to their clients, not maintain comprehensive archives. For legal purposes they maintain a record of what they transmitted for a limited period. Though the sheer volume of content makes it expensive to maintain archives for prolonged periods, so historically tape backups were recycled and thus high frequency archives are limited. Therefore demand for temporal news datasets from institutional investors extends beyond the point where full archives are available.

In this paper we conduct a real world case study demonstrating how multiple data sources can effectively be merged to rebuild temporal textual document datasets with incomplete information. In this scenario the content is known to be replicated, though also modified, which is more commonly known as a near duplicate.

Typically near duplicate detection algorithms are utilised in situations with relatively small datasets. In one case researchers needed to identify similar documents within a dataset containing hundreds of documents (Wei et al. 2008) without revealing any private information to third parties. In another example a thousand email messages from a dataset of hundreds of thousands were analysed to find emails referring to proposals released for public comment (Yang and Callan 2006).

Significantly larger datasets, including tens of millions of documents, are used when researchers are interested in finding exact duplicates or documents which quote verbatim part of another document (Zhang et al. 2010). In other cases the purpose is to track the evolution of a story over time (Shahaf and Guestrin 2010).

Our problem is further complicated by the inclusion of over 20 languages in our dataset, with no clear definition of which language is used in any given news article. Therefore, we need a language independent near duplicate detection algorithm. To the best of our knowledge, near duplicate detection algorithms are targeted at a specific language where the language used in documents is known. Thus this research is novel as we are dealing with a scenario where the language is unknown.

2. Data

In 1996 Reuters agreed to provide backup tapes to a group of academics, which have since formed a company known as Sirca (Sirca 2011). Sirca's focus originally was to provide academics with high frequency prices and as such they only retained classes of messages which were used for prices. In 2003 Sirca decided to also make news available and discovered that a crucial class of message, known as Logical Broadcast Messages (LBM), used to transmit news had not been kept.

In this section we describe how information is transmitted over the Thomson Reuters IDN and why the lack of LBMs is a major problem for assembling a high frequency archive of news. We also describe a mechanism used by Thomson Reuters to provide customers with a limited real time backup of news, which we dub "IDN Backup Pages". Finally, we describe how a subset of Thomson Reuters content was transferred to and is now available in the product known as Dow Jones Factiva (Dow Jones 2011).

The Thomson Reuters NewsScope product (Thomson Reuters 2011b) dataset includes news transmitted over IDN from 2003 onwards. Both this and the Factiva dataset described in this paper are available to any academic whose institution subscribes to Sirca.

2.1. IDN Messages

Messages transmitted over the IDN include an identifier known as a Reuters Instrument Code (RIC). Each RIC has a record classification which is used to distinguish the purpose of the message. The majority of people associate a RIC with an equity (e.g., Microsoft has a RIC of MSFT.O), though there are theoretically 256 different record classifications.

Furthermore, each RIC has a permission code which is used to determine whether customers have the right to access the data. For all messages except those used to transmit news, which we will dub "News RICs", there is a single value. However, News RICs can have up to 256 Permissionable Entity (PE) codes.

The IDN dataset effectively contains all messages transmitted over IDN from the start of 1996 onwards, excluding LBMs until the start of 2003, and messages lost to Data Gaps. Known Data Gaps in the IDN dataset were calculated by analysing periods where no messages occurred during a small time window (which should not happen as there is always content transmitted over IDN). However, that is not to say that there periods of Unknown Data Gaps where information is missing for a particular RIC. The dataset contains over 400 terabytes of content indexed by RIC and time. This included over 843 million news related messages, and over 267 GB of content from 1996 through till the end of 2004.

The classes of messages known as Verify and Update messages are used to transmit the bulk of content over the IDN. A Verify message contains all the necessary information for the RIC, whilst an Update message only overrides the necessary information contained in the most recent Verify message.

The vast majority of trading data is transmitted as Update messages as only the price and volume data changes, potentially hundreds of times per millisecond. However, the vast majority of news content is transmitted as Verify messages and Update messages are only used to adjust pointers between Verify messages.

News messages are transmitted over the IDN by LBMs where the RIC of the LBM is known as the Primary News Access Code (PNAC) for the story. There is a naming convention which requires that a PNAC and all news related RICs begin with a lowercase "n".

The LBM contains the timestamp when the story was released, the Headline, the Attribution, Product Codes, Topic Codes, Named Item Codes, Company Codes, and a Language Code. As Reuters transmits third party content the Attribution field informs customers who published the story, and the Product Codes helps to restrict users from reading content that they haven't subscribed to. The Topic, Named Item and Company codes are added to allow users to search for content which they are interested in (e.g., Corporate Results has a Topic code of "RES", and the company identifier for Microsoft is "MSFT.O"). The Language Code helps users to find content in languages which they can understand.

Journalists are encouraged to re-use the same PNAC for all iterations of the story, so if they add more content to the story or change the metadata they will release another LBM with the same PNAC. LBMs can either be Alerts or Stories, where the content for stories is transmitted by Verify messages. Stories can be **Overwrites**, in which case the entire content is transmitted. They can also be

Appends, in which case an Update message is used to change the pointer details between the Verify messages.

Alerts simply contain the headline and are generally sent by journalists to let customers know that critical information is available, though the journalist has not had time to write the story as yet. Note that Journalists are under considerable pressure to release as much detail to customers as quickly as possible so it is common for multiple alerts to be released within a limited time period before the news is released (e.g., in one case there were 20 alerts in 19 minutes before the story was transmitted).

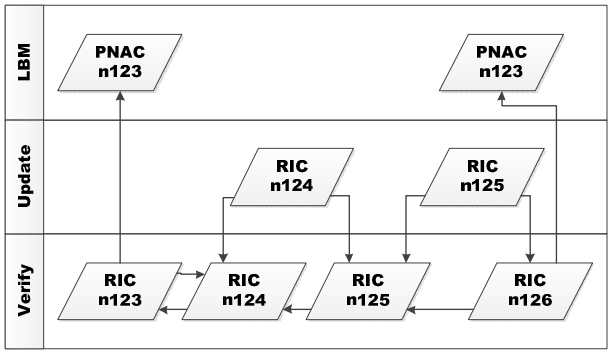


Figure 1. Message Transmission over IDN.

Figure 1 demonstrates how stories are transmitted over the IDN, and reveals that the individual verify messages which make up the story (each has up to 256 bytes of content) are not necessarily connected to each other at the time of transmission.

The Update with RIC n124 demonstrates when extra content is appended to the end of the story without the need to transmit a new LBM. This typcially occurs within a few seconds of the last verify message, as the system had not assigned the identifiers to the story parts before transmission.

The Update with RIC n125 appends further information to the story, but in this case a new LBM was transmitted as the journalist deemed it necessary (e.g., new headline, or change in the metadata).

In the case where there are no LBM messages, it is guaranteed that an LBM would have been sent if the RIC is equal to the PNAC (included in Verify but not Update messages). However, an Update message does not guarantee that another LBM was transmitted, and as such the information can be copied from the most recent Overwrite.

2.2. IDN Backup Pages

Until August 2008 Reuters transmitted News Backup messages over the IDN. The reason for this was that customers only received news when their terminal was turned on, and thus if the customer experience a failure (e.g., power or network outage) they failed to receive the news which they were entitled to during the outage.

The Backup Pages contained a snapshot of news which had been transmitted for the relevant Product including the Story Timestamp, a flag indicating if it was an Alert, the PNAC (in brackets) and the Headline Text. The Story Timestamp provided was in the format HH:MM though it could have been used for the time of the first iteration of the story, not necessarily the current iteration and thus was not entirely reliable. The character between the Story Timestamp and the opening bracket for the PNAC was an asterisk if the message was an alert.

Each message within a Backup Page contains up to 256 bytes of content, though much of that is used in formatting. Therefore, there can be a maximum of three headlines per message. However, a single message can span multiple messages making it necessary to follow pointers between the messages to construct the full headline.

Note that Thomson Reuters' customers can subscribe to multiple products and thus news transmitted to multiple products was replicated across different backup pages, providing further redundancy. The example in Table 1 shows the relevant content for Equity Backup page on the 5th of February 2003, including the time of transmission, the text and the Retransmission code (RTL). The shaded rows are the first verify message (RTL = 0) of messages comprising the page, and bolded text is the first appearance of a headline. Update messages (RTL >= 2) override the content of the message and thus at the time of the last entry users would have seen the text for the last entry at the top of the screen, followed by the text for the second last entry.

Time	Text	RTL
12:13:38.475	**12:13 [nTOPFRX] *TOP NEWS* Foreign exchange**	0
12:14:56.862	**12:14 [nTOPNEWS] *TOP NEWS* Front Page** 12:13 [nTOPFRX] *TOP NEWS* Foreign exchange	2
12:15:43.736	**12:15 [nL0561075] CORRECTED - GUS <GUS.L> offers benchmark euro/sterling bond** 12:14 [nTOPNEWS] *TOP NEWS* Front Page 12:13 [nTOPFRX] *TOP NEWS* Foreign exchange	3
12:15:53.463	**12:15 [nN05131251] RESEARCH ALERT-Prudential raises Alcoa to "buy"**	0

Table 1. Example Backup Page Transition.

2.3. Factiva Messages

In 1987 Reuters began transmitting data over their IDN and shortly afterwards the content was replicated on a service made available by a company known as Finsbury. In 1999 a joint venture between Dow Jones and Reuters saw this product rebranded as Factiva, which is currently known as Dow Jones Factiva since Reuters sold their stake in 2006. After the sale Reuters were provided with a complete set of all the content which was made available via Factiva, excluding company identifiers that were proprietary to Factiva. This included all data between 1987 and the end of 2007, which was split across 1,199 XML files taking roughly 80GB on disk. For the period from 1996 through till the end of 2004 there were over 10 million messages, and over 26 GB of content.

Casual observation reveals that typically only a single iteration of a Reuters' story is included in the Factiva dataset and the timestamps are inaccurate. Most commonly the final iteration of the story is stored, though this is not always the case.

The Factiva dataset included fragments of information which could be used to derive the PNAC of the story though not for the entire dataset. The headline and the story did not necessarily match what was transmitted over IDN as Factiva removed any proprietary Reuters codes

from the content including links to other RICs (contained in triangular brackets "<>"), references to other stories (contained in square brackets "[]"), and instructions to Reuters customers. Furthermore, headlines were frequently altered in Factiva, presumably to add extra information for their customers, or remove proprietary information from Reuters. For example, Factiva added information regarding corrections (e.g., appended "[CORRECTED]") to the headline.

Therefore comparing content between Factiva, IDN and IDN Backup Pages is often not possible via direct comparison.

In this paper we define Factiva Metadata as the combination of all data contained in four separate Factiva fields which were used for content transmitted over IDN. In the XML schema these had field identifiers of "rbbcm", "ipc", "ipcre", and "ipccoric". These fields contained Line Feed delimited values, such as those contained in Table 2. They tended to, but did not necessarily begin with "N2K:" and company codes could begin with "N2K:RICCODE:" or simply "RICCODE:". Not all information contained in these fields was actually transmitted over IDN, so the data has to be validated.

The example in Table 2 reveals the Product Codes were "DA" (Danish News), and "DNP" (Domestic News Pool); the Topic Codes were "DK" (Denmark), and "TEL" (Telecommunications); and the Company Code was "TDC.CO" ("TDC A/S" is a Denmark-based provider of telecommunications solutions).

Factiva Metadata
N2K:DA N2K:DK N2K:DNP N2K:RICCODE:TDC.CO N2K:TEL

Table 2. Example Factiva Metadata.

2.4. Summary

Table 3 provides an overview of how information which would have been transmitted with LBMs can be sourced from the other datasets. In summary before 2003 the IDN dataset contains enough information to know when a story occurred, and what its content was, but not its metadata nor its headline. Furthermore, it is not possible to determine whether Alerts occurred from the IDN News dataset.

	Source		
Data	**IDN**	**Backup**	**Factiva**
Alert	False	True	False
Story Headline	Every Iteration	Every Iteration	Final Iteration
Story Content	Every Iteration	False	Final Iteration
PNAC	True	True	Maybe
PE Codes	True	Subset	False
Product Codes	Subset via PE	Subset	Subset
Topic Codes	Subset	False	Subset
Named Item Codes	Subset	False	Subset
Company Codes	Subset	Subset	Subset
Language Code	False	False	True

Table 3. LBM Fields available from Datasets.

The IDN Backup Page dataset contains enough information to regarding the transmission of both Alerts and Stories occurred, provided no Data Gaps occurred and the Product was Backed Up (which many were not), though neither the story content nor its metadata. However, in the case of Data Gaps it is necessary to validate the PNAC from the IDN News dataset (i.e., ensure that a News RIC with the given PNAC was transmitted at the time).

The Factiva dataset contains enough information to know most of the headline and content (i.e., excluding Reuters proprietary information), and metadata for a single iteration of the story (typically the final iteration). However, the timestamps are suspect and there is no way to verify whether a previous Alert of iteration of the story occurred.

3. Methodology

In this section we provide a brief overview of the methods used to merge the relevant datasets.

3.1. Infer LBM

The first stage in the process is to infer when LBMs for Stories would have occurred based on the arrival of Verify and Update messages, as shown in Figure 1. The timestamp for the Virtual LBM (VLBM) is set as the timestamp of the last Verify message (the Verify message without a Next pointer), as is the case with the Thomson Reuters NewsScope product (Thomson Reuters 2011b).

Independent evaluation of this algorithm by staff at Thomson Reuters has confirmed that there is no delay between the timestamp we propose and the timestamp used in NewsScope.

3.2. Recover Headline

Each Backup message is processed and the headlines (including the PNAC) are assembled by following the necessary links if the headline extends beyond a single message. Indices are constructed such that messages with the exact same headline are stored together, regardless of the Backup Page it was sourced from.

The headline was deemed to be have been transmitted at approximately the time provided on the Backup Page message, if the headline was backed up with one of the following

1. A Verify Message with a Previous Link to a Root Page (has a RIC of the main page for the product).
2. The first line of an Update Message with a RTL of 2.
3. An Update Message with a RTL > 2 where every Update with a lower RTL (>=2) and Verify are available, and headline not stored in any of them.
4. Headline is for a Story (i.e., no "*" after the timestamp), and a Verify Message with same PNAC was transmitted over IDN within 60 seconds beforehand.

If the Headline is for a Story then the timestamp derived within the Infer LBM process is used. Otherwise the (Wei et al. 2008)timestamp from the minimum Backup message containing the headline is used.

3.3. Factiva Overlap

Careful consideration needs to be made when deciding to match IDN content with that within the Factiva dataset. In this section we discuss the three key factors which complicate this process and then propose a naïve, though

effective, method for identifying near duplicates within these datasets.

Firstly, the size of the dataset causes problems as there are tens of thousands of assembled stories per weekday, and it is known that a match can occur a week later. Therefore, for any given message there are potentially hundreds of thousands of comparisons which must be made.

Secondly, we know that the text in the Factiva dataset was frequently modified to remove Reuters' proprietary content. Therefore, even if there were only one copy of a story in each dataset (i.e., ignoring multiple iterations of a story) the content would not be exactly the same. Thus we need to look for near duplicates.

Finally, there are over 20 languages in the IDN dataset and without the LBM there is no clear indication which language was used. Furthermore, even when LBMs are available journalists occasionally assign an incorrect language code. Therefore we need to design a language independent method for performing large scale near duplicate detection.

The first step is to design a parser which is language independent. Apart from Chinese, Japanese and Korean, every language in the dataset includes spaces as delimiters between words (see Table 4 for list of languages used in the Reuters' attributed content), and most use the same delimiters as used in English. Thus our parser includes characters known to be delimiters in any of the languages in the dataset (not necessarily a comprehensive list). Furthermore, we consider each character in the Chinese, Japanese and Korean (CJK) and Hangul (used for Korean text) ranges within UTF8 to be a "word". Finally, we know that Factiva frequently removed Reuters' proprietary content which appeared in brackets (both "<>" and "[]"), and thus our parser removes and content within these delimiters.

When we compare the content of documents we disregard any numbers which occurred. However, when we compare headlines we include numbers as these can be vital for distinguishing between two similar though unrelated stories.

After the content has been parsed, we need a method to compare two documents. The Overlap function for comparing the content of documents *j* and *k* is provided in Eq. (1), where $tf(w_i, d_j)$ is the term frequency of word *i*, in document *j* and $|d_j|$ is the length of document *j* in the case where there are *n* terms.

$$O(j,k) = \frac{\sum_{i=1}^{n} min\left(tf(w_i,d_j), tf(w_i,d_k)\right)}{max(|d_j|,|d_{jk}|)} \quad (1)$$

The Best Match function in Eq. (2) chooses the document, in the subset of all documents where the time between document d_j and d_z document is less than the allowable time window Δt, with the highest overlap value.

$$B(j,\Delta t) = max\big(O(j,k)\big)\,|k \in \{\forall d_z \mid |t(z,j)| < \Delta t\} \quad (2)$$

Once the Recover Headline process has been completed an attempt is made to find a match within the Factiva dataset for each VLBM where at least one of the PE codes was known to be used for transmitting Reuters content prior to 2003 (domain knowledge).

Note that the "formatted" headline text discussed in this section involved a series of heuristics to tokenise the headline and perform a series of token substitutions (domain knowledge). For example Factiva is known to substitute non-ASCII Latin characters with their ASCII equivalent. Furthermore, we know that subsequent iterations of the same PNAC transmitted over IDN can alter the headline text by including the language dependant tokens for Update, Corrected, Correction and Repeat.

A match was established if one of the following were true, processed in the given order calculated:

1. The formatted Factiva headline text exactly matched the formatted headline from the Recover Headline process.
2. The Factiva PNAC exactly matched the IDN PNAC.
3. The result of the Best Match function, restricted to documents with at least one of the same tokens in the formatted headline, was >= 80%.
4. The result of the Best Match function, using All documents, was >= 80% (computationally far more expensive than previous methods).

3.4. Deriving Metadata

In this section we describe the methods used to derive metadata from the available sources. We use the definition of "Factiva Metadata" provided in 2.3. Furthermore we define the "Story Metadata" as any text delimited by brackets (both "<>" and "[]") that occurred in the headline or story body for the current or previous iterations of the IDN story.

Note that long ranges are used to scrape documented metadata as it has been noticed that it can take years before a code is officially documented.

3.4.1. Product Codes

Product codes can be derived from Backup Pages, or from the Factiva Metadata. Furthermore, for stories we have the list of PE Codes and can derive the list of possible Product Codes from the statistics messages (described in the following paragraph). However, a single PE can be used to transmit multiple Products and therefore this information is not always useful. To determine whether the code is a valid Product Code we scrape information from Statistics messages over the range 01/01/1996 through 30/06/2010.

Statistics messages have a PNAC beginning with "n%" (not included in NewsScope), which were transmitted once per day with a list of PE Codes mapped to the corresponding Product Code with the count of the number of messages transmitted during the past day. Our dataset includes over 500 unique values.

3.4.2. Topic Codes

Topic codes can be derived from Story Metadata and Factiva Metadata. This field frequently includes the code for the relevant language, which is simply "L" followed by the Language Code. Furthermore, the code for the Attribution is frequently included, which is "RTRS" for

Reuters' content. Therefore we supplement codes derived from Story Metadata and Factiva Metadata with the relevant language and attribution codes.

To determine whether the code is valid we scrape the content of news with the PNAC nSUB01033, which is transmitted typically twice per day, from 01/01/1996 through 30/06/2010.

The nSUB01033 news contains a Line Feed delimited list of entries beginning with a Code and followed by a description after some whitespace. It contains descriptions for Product codes as well as Topic Codes. Therefore, if the code is known to be a Product Code we do not include it in our dataset, except when determining if a code was documented.

We know that Product Codes are used as Topic Codes occasionally though it isn't clear if this was intentional or a mistake by the journalist. Therefore we reduce complexity by ignoring this scenario. Our dataset includes over 1,300 unique values.

3.4.3. Named Item Codes

Named Item codes can be derived from Story Metadata and Factiva Metadata. To determine whether the code is valid we scrape the content of news with the PNACs nREP01033 and nADD01033, which is transmitted typically twice per day, from 01/01/1996 through 30/06/2010 (the latter has only been transmitted since November 2000).

These news articles contain Line Feed delimited lists of entries beginning with a Code and followed by a description after some whitespace. We know that there are numerous cases where Named Item codes are also RICs, and thus can be considered to be Company Identifiers (they refer to a Page transmitted over IDN). However, we reduce complexity by disregarding this issue as it is unlikely that clients would want to search for a Page when they are actually looking for an asset with prices. Our dataset includes over 5,100 unique values.

3.4.4. Company Codes

Company codes can be derived from Story Metadata and Factiva Metadata. To determine whether the code is valid we construct lists of every RIC transmitted over IDN, barring news related RICs (beginning with "n"), for the month before, during and after the news was transmitted. We supplement this with a list of legitimate RICs provided by Thomson Reuters for the period in question. The list provided by Thomson Reuters includes RICs not transmitted over IDN, and the IDN list includes RICs not in the Thomson Reuters list. Therefore, combing the two provides a comprehensive list of what where legitimate RICs during the given period.

In January 1996 the dataset included over 2 million unique values, though by December 2004 the dataset included over 7.5 million unique entries. As a matter of interest the dataset grows to over 27.1 million unique entries by May 2010.

3.4.5. Language Codes

Language codes can be derived by copying the Language Code from the matched Factiva content.

There are Product Codes which are only meant to be used to transmit news in a particular language, so theoretically the language code could be derived that way. However, this requires in-depth analysis of the dataset which we cannot do until we have generated it.

Furthermore, tools like Google Translate (Google 2011), and Babel Fish (Yahoo! 2011) could be used to make an educated guess regarding the language. However, this is not exactly practical for datasets of the size in question, not to mention the legal issues involved.

3.5. Delay

We know that there were inherent delays between the time an LBM was transmitted and the time it appeared in a Backup Page or in the Factiva dataset. We need to determine these times so we can use realistic time windows in future research for determining the impact of news.

3.5.1. Backup Messages

For alerts there is no way to verify whether the timestamp was accurate from either the IDN or Factiva datasets. However, for stories we can determine the delay based on the timestamp of the first Verify message for the Story (the PNAC and RIC are the same) and that of the Backup. As Append messages frequently occur shortly after Overwrite messages, and thus the Backup message could be mapped to either, we restrict the process to the First Story for a given PNAC.

3.5.2. Factiva

Casual Observation has indicated that the Factiva dataset tends to only contain a single iteration of a story transmitted over the IDN (typically the final iteration). Therefore we measure the average delay between the timestamp of the VLBM and that of the mapped Factiva message in the case where there is only one instance of VLBMs with said PNAC within the allowable time window Δt.

Note that there are cases where journalists have transmitted subsequent iterations of the same story using different PNACs. However, the personnel that maintain Factiva seem to have kept a single version of the actual story, regardless of the PNAC, so this should not impact the results.

4. Results

We divide the results section into subsections covering the characteristics of the Raw messages, and the success of the Recovering Headlines in general. We also provide more in depth analysis of the Language specific characteristics of the process, and the success of deriving metadata. Finally we provide details of the delays calculated using the methodology outlined in the previous section.

For all experiments Δt was seven days, as there were several known examples where a match could only be found after several days, and seven days was within the limits of computational power.

4.1. Raw Messages

To grasp the scale of the dataset we have plotted the number of raw IDN and Factiva messages on an average weekday over time in Figure 2. Note that the Verify and Update messages are News related, though also include the Backup Pages. It is clear from the downturn in Update messages in 2008 that the Backup Pages ceased to be transmitted over IDN.

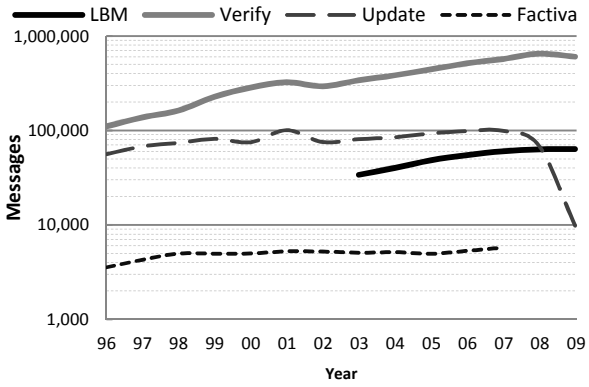


Figure 2. Average Raw Messages.

In Figure 3 we plot the sizes of the various Hash Tables required to index and access the story content on an average weekday over time. Note that P is short for PNAC and H is short for Headline, whilst IDN, Fact (Factiva) and Back (Backup) refer to the source of the data. The relationship between the number of Unique PNACs and Headlines is evident for both Factiva and the IDN versus Backup messages. The results demonstrate that the PNAC information was not available in Factiva messages prior to 1998 and was discontinued in 2004. Thus all matching had to be performed based on the headline or story content during these periods.

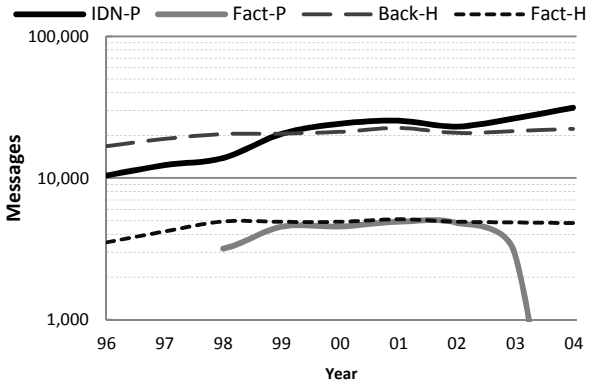


Figure 3. Average Hash Table Sizes.

4.2. Recovering Headlines

In Figure 4 we investigate the ratio of Backup and Factiva messages which were recovered. We do not include the day of or the day after a Known Data Gap as there is no guarantee that all information was available. The results clearly reveal that our algorithm for extracting headlines from the Backup Pages is highly effective, with a rate of over 99% for every year.

The results in Figure 4 demonstrate that our naïve algorithm detects near duplicates in over 90% of cases for every year. The results for 1996 are much worse than for other years as there were far more Data Gaps during said year. Furthermore the results in 2003 are low, in comparison to all years except 1996. In 4.3 we investigate this more closely and find that there is an issue with Chinese content on that year. Disregarding these two years, our algorithm successfully matches over 95% of content between the two datasets.

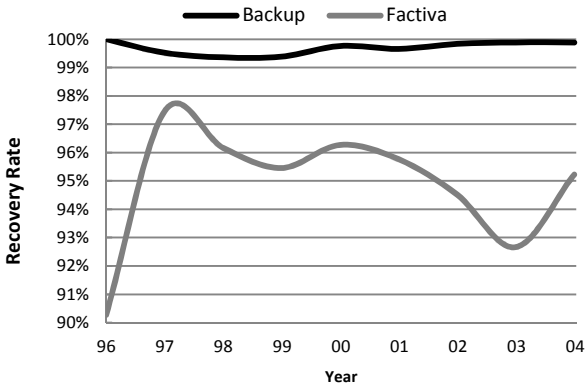


Figure 4. Recovery Rate.

In order to establish how Factiva matches are made we plot the percentage of matches made using each method in Figure 5. In this figure DH, DP, OH, and OS defines the rate of Direct Headline, Direct PNAC, Overlap by Headline, and Overlap by Story matches respectively, as defined in methods 1-4 of 3.3.

Clearly the Direct Headline method which utilises the formatted headline text is the most effective. This is desirable as it is not only computationally cheap it is also more likely to be a true match. To keep things in perspective we must consider that only 42.22% of matches made in the entire dataset had the exact same headline in the IDN and Factiva datasets. Overall 88.96% could be matched by the formatted headline, so it is very important to apply heuristics when matching the headlines.

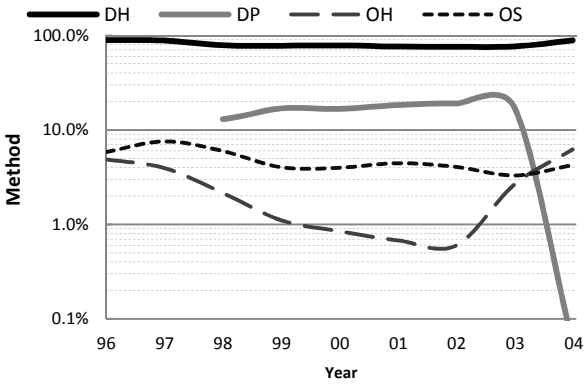


Figure 5. Factiva Overlap Method.

Furthermore, Figure 5 shows that the PNAC was a useful method for finding matches when the data was available. The Overlap by Headline method was nowhere near as effective as the Overlap by Story content method. The

most likely explanation is that the headline was not in the IDN Backup Pages, making it impossible to compare the IDN and Factiva headlines.

4.3. Language

In this section we analyse the language specific characteristics of the dataset to determine how our language independent algorithm performs. The reader should note that an analysis of English content in the 2003 and 2004 period has revealed that the Factiva dataset contains content which was not transmitted over IDN as news. Examples of such content seem to be summaries of news relevant to a particular market, or the summaries or prices for a type of asset. This type of content may actually have been transmitted over IDN as Page Based data, but further investigation is required to be sure. Therefore, even if a perfect algorithm were available we would not expect 100% of content to be matched with Factiva content.

In Table 4 we investigate what percentage of content in each dataset is in each of the 20 most commonly used languages in the IDN dataset, in descending order. The IDN column is the ratio of the final iteration of Reuters attributed news (i.e., each PNAC is only counted once) transmitted over IDN for 2003 and 2004. The Factiva columns show the ratio of Reuters' copyrighted content from the Factiva dataset for 2003 and 2004, and pre-2003 respectively.

Language Code	Language	IDN (2003&2004)	Factiva (2003&2004)	Factiva (pre-2003)
EN	English	48.46%	52.81%	60.12%
JA	Japanese	**6.73%**	**4.92%**	6.05%
FR	French	6.26%	6.87%	6.33%
ES	Spanish	5.76%	6.28%	6.57%
ZH	Chinese	5.01%	8.54%	3.32%
DE	German	4.48%	5.42%	7.19%
IT	Italian	3.34%	3.66%	2.36%
RU	Russian	3.04%	3.33%	1.93%
KO	Korean	2.68%	0.00%	0.00%
PT	Portuguese	2.55%	3.02%	1.34%
SV	Swedish	**1.98%**	**1.15%**	1.08%
NO	Norwegian	**1.39%**	**0.74%**	0.80%
TR	Turkish	1.15%	1.26%	0.98%
PL	Polish	0.96%	0.98%	0.79%
CS	Czech	0.61%	0.60%	0.29%
EL	Greek	0.46%	0.00%	0.00%
DA	Danish	**0.45%**	**0.42%**	0.69%
NL	Dutch	0.00%	0.00%	0.06%
BG	Bulgarian	0.00%	0.00%	0.05%
HU	Hungarian	0.00%	0.00%	0.07%

Table 4. Language Content Ratio.

Table 4 demonstrates that Factiva does not contain any Korean or Greek content, and therefore we disregard those languages in future analysis. We have shaded all results where the difference between the IDN and Factiva ratio; where the IDN value is scaled to reflect the lack of Korean and Greek content; exceeds one standard deviation from the mean difference for languages other than Japanese, Chinese, Swedish and Norwegian (the outliers). Furthermore, we have bolded those shaded results where there is a higher IDN than Factiva content ratio.

The shaded results in Table 4 reveal that the Scandinavian languages (Swedish, Norwegian, and Danish) are less likely to appear in the Factiva dataset. Close inspection has revealed that IDN content with these Language Codes often has a headline in the given language, but story content in Factiva. Therefore, it is likely that staff at Factiva opted to ignore said content.

In Table 5 we investigate the ratio of content where a match was successfully found in the other source. In all cases we disregard the day before, after and day of a Known Data Gap as we cannot guarantee the iteration of the story used by Factiva would be in both datasets. The IDN column only includes the final iteration of Reuters attributed content. We have bolded every entry with greater than 99%, and shaded every entry with less than 90% of content matched.

In every language; excluding Chinese, Dutch, Bulgarian and Hungarian; over 90% of the Factiva content was successfully matched in 2003 and 2004. The latter three languages can be disregarded as no content for said languages was contained in the IDN dataset (at least attributed to Reuters). The results in Table 4 show that there is much more Chinese content in the Factiva dataset than the IDN dataset. It appears that this was a temporarily quirk in the Factiva dataset, as in 2004 the language match ratio was 99.72%, and prior to 2003 it was 99.45%. This requires further investigation, but as the authors have no knowledge of Chinese it may be difficult to establish the exact cause.

Language Code	Language	IDN (2003&2004)	Factiva (2003&2004)	Factiva (pre-2003)
EN	English	85.89%	92.84%	93.82%
JA	Japanese	55.42%	**99.21%**	**99.71%**
FR	French	95.93%	**99.52%**	**99.30%**
ES	Spanish	85.56%	97.98%	95.68%
ZH	Chinese	94.15%	86.37%	**99.45%**
DE	German	97.39%	98.01%	97.82%
IT	Italian	97.58%	**99.83%**	**99.48%**
RU	Russian	94.68%	92.60%	90.60%
PT	Portuguese	96.36%	**99.05%**	97.05%
SV	Swedish	90.10%	**99.68%**	**99.75%**
NO	Norwegian	92.13%	**99.94%**	**99.56%**
TR	Turkish	97.62%	**99.29%**	**99.77%**
PL	Polish	92.05%	**99.92%**	**99.71%**
CS	Czech	89.79%	**99.90%**	**99.76%**
DA	Danish	82.95%	**99.52%**	**99.78%**
NL	Dutch			**99.78%**
BG	Bulgarian			**99.98%**
HU	Hungarian			97.69%

Table 5. Language Match Ratio.

In every language over 90% of the Factiva content was successfully matched prior to 2003. The Russian specific Product was not Backed Up using the IDN Backup mechanism and thus all matches were made by the PNAC or by Content, leading to lower value than the other languages. However, there is little difference between the pre and post 2003 periods.

In every language; excluding English, Japanese, Spanish, Czech, and Danish; over 90% of IDN and Factiva content were matched. Over 99% of the Japanese, Czech, and Danish Factiva content was matched to IDN stories during the same period. This result could mean that the Factiva dataset did not include all of the content for these languages. The Japanese result in Table 4 supports this hypothesis, as there is substantially less content in the Factiva dataset than in the IDN dataset during the same period. Alternatively it could mean that these languages are more likely to suffer from journalists using different PNACs for subsequent iterations of the same story.

The results in Table 5 prompted further analysis of the English and Spanish content in the two datasets. Deliberate manual analysis revealed that there is content from both these languages which were not stored in the Factiva dataset.

We evaluate the quality of matches in Table 6 by calculating the mean Overlap between IDN and Factiva content, using the function defined in Eq. (1). We selected the maximum result of all iterations of the story and only used cases where the final iteration contained at least 50 tokens. The number of tokens was chosen arbitrarily as it was found that many stories with fewer tokens contained tables quoting numerous prices, and as numbers were ignored it became harder to establish the quality of the match. Furthermore all iterations of the story were evaluated, as it was found that Factiva did not always update stories to reflect the final iteration which was transmitted over IDN.

Language Code	Language	2003&2004	Pre-2003
EN	English	90.75%	94.37%
JA	Japanese	96.99%	97.88%
FR	French	94.71%	95.55%
ES	Spanish	93.51%	95.92%
ZH	Chinese	96.82%	96.81%
DE	German	96.67%	96.18%
IT	Italian	90.36%	94.51%
RU	Russian	93.59%	96.94%
PT	Portuguese	94.08%	95.22%
SV	Swedish	91.99%	94.45%
NO	Norwegian	92.53%	95.05%
TR	Turkish	86.91%	91.60%
PL	Polish	91.53%	94.68%
CS	Czech	87.54%	90.36%
DA	Danish	89.97%	94.06%
NL	Dutch		91.78%
BG	Bulgarian		91.39%
HU	Hungarian		90.58%

Table 6. Overlap.

The results in Table 6 reveal that the overlap is over 90% for all languages prior to 2003 and all but Turkish, Czech and Danish in the 2003 and 2004 period. Whilst the value was only marginally lower the reader should note that the standard deviation for Turkish, Danish, Czech, French, and Italian were higher than for other languages (Note: the statistical significance of these results was not evaluated). Anecdotally we noticed that these languages tended to suffer from character substitutions in the Factiva dataset. As we mentioned in 3.3 we performed character substitutions for comparisons of headlines, but not for stories. Therefore the results could be skewed when journalists used a large number of non-ASCII Latin based characters.

These results suggest that our naïve language independent near duplicate detection algorithm is effective at finding matches for all the languages in the datasets.

4.4. Deriving Metadata

To further evaluate the quality of the match we analysed the derived metadata in the processes described in 3.4. In Table 7 we test this method by measuring the percentage of metadata in each field which was documented (as detailed in 3.4), the percentage which was successfully derived from the Factiva Metadata and Story Metadata, and the percentage of derived data which was sourced exclusively from the Factiva Metadata. We have added a further row which addresses Company Codes that include a dot, as these are used for Equities (e.g., "MSFT.O" is the RIC for Microsoft) that are arguably more important to investors who purchase the Thomson Reuters NewsScope data.

For the purpose of evaluation we have ignored Japanese and Chinese content as the Factiva Metadata is sparse for these languages, though the Company Codes are derived as effectively as the other languages.

Metadata Field	Documented	Derived	Factiva
Product Codes	100.00%	95.91%	0.56%
Topic Codes	99.99%	97.27%	67.22%
Named Item Codes	95.93%	93.45%	86.70%
Company Codes	97.03%	91.20%	8.28%
Company with Dot	98.15%	94.60%	8.49%

Table 7. Metadata Quality.

To put these values in perspective the reader should appreciate that every LBM includes a Product Code, and every LBM should include a Topic Code even it simply includes the code for the relevant language. In the 2003 and 2004 dataset 40.64% of Company Codes and 21.70% of Named Item Codes for the final iteration of stories with a Factiva match had non-blank values.

The results in Table 7 strongly suggest that our documentation is thorough as all fields have values over 95%. The Named Item Codes and Company Codes values may be slightly lower as journalists are more likely to make typographical errors with these values as they include more characters.

All metadata fields have derived values exceeding 90%. The reader will note that we are more likely to derive a RIC which includes a dot, than otherwise. Reuters' journalists have programmable keyboards which they can use to store the RICs they frequently refer to. Therefore, more commonly used Equity RICs are likely to be programmed, reducing the likelihood of typographical errors and increasing the likelihood of the journalist inserting the code into the Story Metadata. These results are highly encouraging as we know this metadata is extremely important.

The Factiva column shows that Product Codes and Company Codes tend not to be sourced exclusively from the Factiva Metadata. This is expected for Product Codes

as they are derivable from the PE codes, and transmitted via the IDN Backup Pages. The Company Code data tends to be transmitted in the headline and/or Story Metadata, making it less important to rely on Factiva for this data.

Whilst the Topic Codes typically are available in the Factiva Metadata, they do not include the topic code for the language or the attribution. Thus only slightly more than two thirds of Topic codes are sourced exclusively from Factiva. The Named Item Codes tend not to be included in the Story Metadata, making us reliant on the Factiva Metadata for this data.

Whilst we cannot guarantee that the pre-2003 dataset derived a similar ratio of metadata, the results indicate that our algorithm effectively reassembles metadata with incomplete information.

4.5. Delays

To establish the accuracy of timestamps we need to investigate the delays inherent in the dataset. Figure 6 shows the average delay between the First Alert and the First Story (FA-FS), the delay between the First and Last Story (FS-LS), and the delay for the headline to occur on an IDN Backup page (Backup) using the methodology define in 3.5.1. In 1996 it took a journalist roughly 12 minutes between sending the first alert and sending the first iteration of the story, though by 2004 this was reduced to 4.5 minutes as journalists come under increasing pressure to release content to the market quickly. In most years a story tends to span over 2 hours from the first to the last iteration. Figure 6 demonstrates that the IDN Backup mechanism consistently incorporated new headlines in less than 2 seconds (the average is 1.49 seconds).

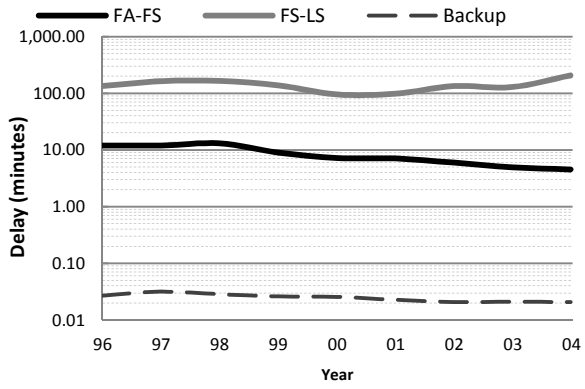


Figure 6. IDN Delays.

In Figure 7 the average delay for Factiva matches are shown overall (All), for matches found on within 24 hours (Within Day), and matches found within an hour (Within Hour), using the methodology defined in 3.5.2. The average delay for all matches is very large as some matches are found up to a week later. However, 97.38% of matches are found within a day, and 93.46% are found within an hour.

As the IDN Backup Pages are not comprehensive, we cannot guarantee that an Alert did not occur. Furthermore, as there are periods of Known Data Gaps (periods where we know no message were kept in the IDN dataset) we need to consider the average delay between the First and Last Story. Even if the average Factiva delay was less than four minutes, as it was for all the Within Hour results except in 1996 years, this is still unacceptable for research into the high frequency impact of news, such as (Ederington and Lee 1995). However, we also need to allow for over two hours for previous iterations of the story, and alerts. Thus the timestamps provided by Factiva are not reliable when being used to investigate the intraday effect of news.

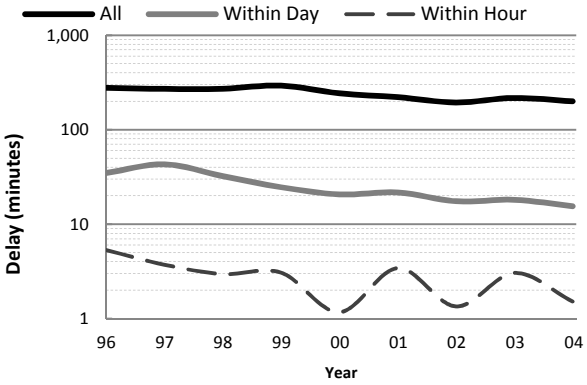


Figure 7. Factiva Delays.

5. Conclusions

In this paper we have conducted a real world case study demonstrating how multiple data sources can effectively be merged to rebuild temporal textual document datasets with incomplete information. Specifically we have combined the story content transmitted over the Thomson Reuters Integrated Data Network (IDN); the headlines which were backed up by Thomson Reuters; and the Thomson Reuters headlines, content and metadata archived by Dow Jones Factiva.

We have built a very close approximation, with highly accurate timestamps, of the actual temporal news dataset which was transmitted over IDN during a seven year period (1996 through to and including 2002). We have validated our algorithm on a further two years of data (2003&2004) which has proved its effectiveness at deriving metadata.

One of the novel contribution is the application of a near duplicate detection algorithm to a dataset where the language used in given news article is unknown. The effectiveness of our naïve language independent near duplicate detection algorithm has been confirmed on 18 languages within the given dataset. The results show that the algorithm effectively matches the content for all these languages.

The primary motivation for constructing this temporal textual document dataset is that institutional investors are building increasingly more sophisticated algorithmic trading engines that account for textual as well as numerical information. To train these engines they need large datasets of information with highly accurate timestamps that cover long periods with differing trading conditions. Thus it is critical to determine the scale of the inherent delays which have been introduced into the dataset due to the lack of actual LBMs.

Our results have demonstrated that the timing of our messages is highly accurate (worst case is a delay of 1.49 seconds for Alerts). This has been independently evaluated by staff at Thomson Reuters who have confirmed that there is no delay between the timestamp we propose and the timestamp used in NewsScope. A by-product of this analysis is the revelation of the inadequacy of timestamps, critical to algorithmic trading engines, provided by Dow Jones Factiva, at least in respect to Thomson Reuters' content.

6. References

Dow Jones: Dow Jones Factiva. http://www.factiva.com/. Accessed 11 Oct 2011.

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):117-34.

Fama, E. F. (1970): Efficient Capital Markets: A Review of Theory and Empirical Work. *Journal of Finance,* **25**(2):383-417.

Google: Google Translate. http://translate.google.com/. Accessed 11 Oct 2011.

Latham, M. (1986): Informational Efficiency and Information Subsets. *Journal of Finance,* **41**(1):39-52.

London Stock Exchange: History. http://www.londonstockexchange.com/products-and-services/rns/history/history.htm. Accessed 11 Oct 2011.

Mitra, L. and Mitra, G. (2011): Applications of News Analytics in Finance: A Review. In *The Handbook of News Analytics in Finance*. 1-36. Mitra, G. and Mitra, L. (eds.). John Wiley & Sons.

Mittermayer, M. A. and Knolmayer, G. F. (2006): NewsCATS: A News Categorization and Trading System. *Proc. 6th IEEE International Conference on Data Mining (ICDM)*, Hong Kong, China. 1002-7.

Roll, R. (1988): R^2. *Journal of Finance,* **43**(3):541-66.

Shahaf, D. and Guestrin, C. (2010): Connecting the Dots between News Articles. *Proc. 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington, DC, USA. 623-632, ACM Press.

Shiller, R. J. (2003): From Efficient Markets to Behavioral Finance. *Journal of Economic Perspectives,* **17**(1):83-104.

Sirca: Home. http://sirca.org.au/. Accessed 11 Oct 2011.

Tetlock, P. C., Saar-Tsechansky, M. and Maskassy, S. A. (2008): More than Words: Quantifying Language to Measure Firms' Fundamentals. *Journal of Finance,* **63**(3):1427-67.

Thaler, R. and De Bondt, W. F. M. (1985): Does the Stock Market Overreact? *Journal of Finance,* **40**(3):793-805.

Thomson Reuters: History. http://thomsonreuters.com/about_us/company_history/. Accessed 11 Oct 2011.

Thomson Reuters: NewsScope. http://thomsonreuters.com/products_services/financial/financial_products/a-z/newsscope_application_license/. Accessed 11 Oct 2011.

Wei, J., Murugesan, M., Clifton, C. and Luo, S. (2008): Similar Document Detection with Limited Information Disclosure. *Proc. 24th IEEE International Conference on Data Engineering (ICDE)*, Cancún, Mexico. 735-743.

Yahoo!: Babel Fish. http://babelfish.yahoo.com/. Accessed 11 Oct 2011.

Yang, H. and Callan, J. (2006): Near-duplicate Detection by Instance-level Constrained Clustering. *Proc. 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Seattle, Washington, USA. 421-428, ACM Press.

Zhang, Q., Zhang, Y., Yu, H. and Huang, X. (2010): Efficient Partial-duplicate Detection based on Sequence Matching. *Proc. 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Geneva, Switzerland. 675-682, ACM Press.

Unsupervised Fraud Detection in Medicare Australia

MingJian Tang*, B. Sumudu.U. Mendis, D. Wayne Murray, Yingsong Hu and Alison Sutinen
Strategic Data Mining Section
Department of Human Services
134 Reed Street, Tuggeranong 2900, ACT
*ming.jian.tang@humanservices.gov.au

Abstract

Fraud detection is a fundamental data mining task with a wide range of practical applications. Finding rare and evolving fraudulent claimant behaviour in millions of electronic Medicare records poses unique challenges due to the unsupervised nature of the problem. In this paper, we investigate the problem of efficiently and effectively identifying potential non-compliant Medicare claimants in Australia. We propose an unsupervised and data-driven fraud detection system called UNISIM. It integrates various techniques, such as feature selection, clustering, pattern recognition and outlier detection. By utilising the beneficial properties of these techniques, we are able to automate the detection process. Additionally, useful temporal patterns are extracted from the existing data for future analysis. Through extensive empirical studies, UNISIM is shown to effectively identify suspects with highly irregular patterns. Additionally, it is capable of detecting groups of outliers.

Keywords: unsupervised fraud detection; health data; Hidden Markov Models; temporal pattern recognition.

1 Introduction

As a major service delivery program of the Department of Human Services, Medicare Australia (MA) looks after the health of Australians through efficient services and payments, such as the Medicare Benefit Schedule (MBS), the Pharmaceutical Benefits Scheme (PBS), the Australian Childhood Immunisation Register and the Australian Organ Donor Register. According to MA's annual report (Medicare Australia, 2011), the PBS subsidies the cost of listed prescription medicine and the Repatriation PBS (RPBS) provides eligible veterans and war widows and widowers some additional medicines and dressings at concession rates. In *2009-2010*, MA processed approximately *198* million services or *$8.3* billion in benefits under the PBS and RPBS indicating an increase of *7.8%* over the previous year. As part of the Human Services portfolio, MA bears the responsibility for ensuring that public funds are used appropriately by maintaining the integrity of the programs it administers. In *2009-2010* alone, MA recovered more than *$10.2* million from compliance activities. With the unprecedented growth of services and payments, MA faces new challenges with respect to efficiently and accurately detecting non-compliant patients. Patients are considered as consumers in MA since they consume certain medical resources.

As part of MA's integrity program, the Prescription Shopping Program aims at protecting the integrity of the PBS by identifying and reducing the number of patients obtaining medicine subsidised under the scheme in excess of medical need (Medicare Australia, 2011). Automating the process of detecting possible prescription shoppers is very challenging in nature, due to:

- Large amount of real-life medical data coupled with complex and implicit correlations.
- Noise is prevalent in real-life data hampering the direct application of many state-of-art data mining techniques (garbage in and garbage out).
- Absence of holistic and standardised domain knowledge (from data miners' point of view).
- Prescription behaviours are constantly evolving (existing predictive models or past knowledge can become obsolete).
- Minimising the number of false positive (i.e. identifying consumers as prescription shoppers when they have legitimate medical reason for their PBS load)

Two analytical systems have been developed in MA for facilitating efficient detection of prescription shoppers by utilising the PBS data. The work in (Ng *et al.* 2010) focuses on capturing the temporal (explicit) and spatial (postcode-based) aspects of consumers' prescription behaviours, whereas the paper (Mendis *et al.* 2011) examines sequential prescription patterns from either a global or a localised view. Due to the complex nature of human behaviours coupled with their implicit health conditions, there can be many different fraudulent cases with peculiar behavioural patterns. With some global knowledge (e.g. ranks about consumers prescription history either cost-wise or quantity-wise), some cases can be easily detected yet most of them are disguised deeply amongst genuine consumers. Considering the labour-intensive manual approach and the complex nature of these cases, it is highly unlikely that all cases can be enumerated. Therefore, an automated and adaptive detection system is urged for complementing the existing systems.

1.1 Contributions and Paper Organisation

In this paper, we investigate an important real-life fraud detection problem in the domain of health care data. Due to the complex nature of the underlying data, we divide the problem into a set of sub-problems and conquer them

accordingly. We propose an unsupervised fraud detection system called UNISIM targeting in particular prescription shoppers. UNISIM is comprised of a number of functional components namely feature extractor, cluster builder, model constructor and outlier detector. Each component is responsible for performing data mining tasks including feature selection, clustering, pattern recognition and outlier detection, into an cohesive system. Completing such an unsupervised system is not only technically more challenging but also more desirable in the practical data mining applications. We conduct extensive experiments using real-life medical claim data. The system can efficiently extract the hidden consumer patterns with respect to their temporal prescription behaviours. We also demonstrate its effectiveness on identifying potential prescription shoppers.

The remainder of the paper proceeds as follows. In section 2, we describe the available data and formally define the problem. Section 3 presents the design of UNISIM. Technical details of UNISIM are provided in section 4. UNISIM is extensively evaluated by using real-life datasets in section 5, and section 6 concludes the paper.

2 Preliminaries

2.1 Available Data

All Australian permanent residents and certain categories of overseas visitors have access to the Medicare Program. MA pays benefits to any eligible person to cover a set proportion of their incurred medical expenses. A consumer needs to lodge his/her medical bills in terms of claims through MA in order to get the relevant benefits. MA stores these claims in transactional databases. Each transaction record holds rich information including consumer details and medical provider details (e.g. name, the date and type of service provided). Additionally, there are reference databases which contain information about various MA services. Since we are mainly interested in identifying prescription shopping consumers (i.e. fraudulently gaining access PBS medications in excess of legitimate medical need), we focus ourselves on data pertaining to consumers and their respective PBS drug prescription details.

In general, MA stores consumer data in three different databases namely consumer directory for general information, MBS claims for medical services and consultations, and PBS claims for drug prescriptions. Linking both MBS and PBS databases would be beneficial for substantiating the genuine drug needs of a consumer. Unfortunately, we are only allowed to derive data from one of them due to the existing privacy legislation. Therefore, the PBS data is chosen for the purpose of countering prescription shoppers.

Each consumer, who obtains subsidised PBS drugs, is represented by at least one transaction in the PBS database. For an example, each transaction may take the following form:

$$(PhID, PrID, \{(Item, Cost)\}, Dos, Dop)$$

where *PhID* is the identifier of the pharmacy at which the drugs are supplied and *PrID* uniquely identifies the prescribing doctor or prescriber. The set of PBS items charged along with their respective costs to Medicare is encapsulated in $\{(Item, Cost)\}$. *Dos* and *Dop* are two time stamps recording the date of supply and the date of prescribing. Additional consumer information is available from the consumer directory including consumer ID, name, age, gender and address.

2.2 Problem Statement

In Ng *et al.* (2010), three classes of drugs were identified as being susceptible to abuse by prescription shoppers namely: opioids, benzodiazepines and psychostimulants. A list of drug names and their respective classes are given in table 1.

Name	Class
Alprazolam	Benzodiazepine
Clonazepam	Benzodiazepine
Diazepam	Benzodiazepine
Nitrazepam	Benzodiazepine
Olanzapine	Benzodiazepine
Oxazepam	Benzodiazepine
Quetiapine	Benzodiazepine
Temazepam	Benzodiazepine
Buprenorphine	Opioids
Codeine	Opioids
Fentanyl	Opioids
Hydromorphone	Opioids
Methadone	Opioids
Morphine	Opioids
Oxycodone	Opioids
Tramadol	Opioids
Dexamphenidate	Psychostimulant
Methylphenidate	Psychostimulant

Table 1: target drugs

Besides the above highly specialised knowledge, some fragmental and intuitive indicators about typical prescription shopping can be:

- Contradicting drug prescription (e.g. sleeping tablets versus stimulative tablets).
- Visiting a diversity of doctors for similar types of drugs.
- Excessive drug quantities over a set period.
- Sudden changes in prescription behaviours.
- Recurrent large temporal gaps after getting lots of drugs.

The main objective of this paper is to propose a workable fraud detection system. It is required to identify consumers with irregular prescription behaviours over a certain period of time (e.g. *1* to *4* years). Defining accurate notion of irregularity, to some extents, requires inputs from domain experts, which adds extra overheads. Instead, the system needs to autonomously derive and identify such patterns. Since the eventual users of the system are mainly non-technical and business-oriented, rendering interpretable results also plays a vital part. Overall, the resultant system needs be unsupervised and flexible due to the absence of labelled data and practicality issues.

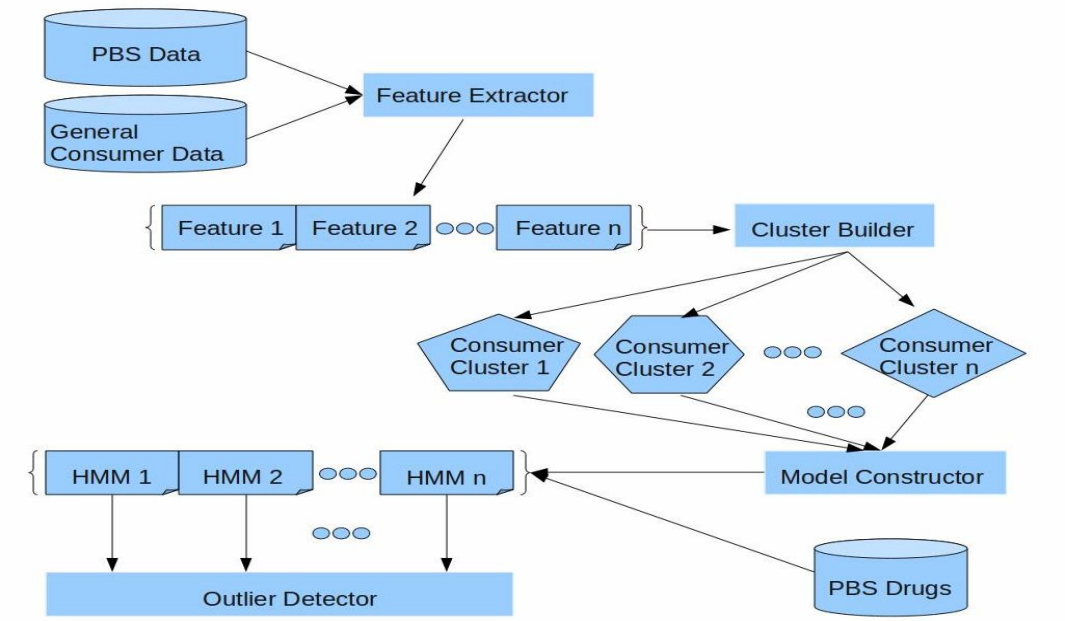


Figure 1: A high-level view of UNISIM

3 UNISIM - A Holistic View

The proposed system consists of several components as follows:

- **Feature extractor.** It harvests the PBS database and the general consumer directory for constructing and preparing featured consumer prescription data.
- **Cluster builder.** It examines the constructed consumer data and labels them on certain criteria (e.g. frequency of drug prescriptions or similarity of temporal prescription sequences).
- **Model constructor**. It learns hidden temporal patterns from the identified clusters of consumers and builds respective Hidden Markov Models (HMMs) (Rabiner 1989) for capturing consumers' implicit prescription patterns according to their cohorts.
- **Outlier detector.** It generates an *n*-dimensional score vector for each consumer then compares each consumer against his/her reachable peers to derive a final outlier-ness score.

Figure 1 depicts how these components logically fit into a common framework. Potentially we can utilise different methods or techniques for each component. Such a semi-open and modularised design maximises the flexibility of the system and changes to each component can easily be localised.

4 Technical Design

In this section, the intriguing details about each component are covered and discussed.

4.1 Feature Extractor

Feature extraction is a process attempting to filter out components of a data record which are irrelevant to the task at hand. Albeit the stored PBS data can be problematic (e.g. noise in terms of highly irregular patterns), each record is rich in features in terms of various types of attributes. Feature rich datasets have benefits and disadvantages. On one hand, they are beneficial for representing the underlying data with various characteristics and granularities. On the other hand, the curse of dimensionality can hamper the performance of existing data mining techniques (e.g. clustering and outlier detection). It is mainly because the projected data points become sparser with the increase of feature dimensionalities (database attributes). The earlier work (Ng *et al.* 2010 and Mendis *et al.* 2011) was conducted mainly on grouped consumer prescription data based on postcodes. This grouping, to a limited extent, captures the spatial correlation of consumers' behaviours with respect to their drug prescriptions. The benefit can drown in the pool of noisy data introduced by people with different demographics.

The overall aim of the feature extractor is to judiciously select a subset of feature attributes for representing the consumer prescription activities in a more compact way. These more compressed features can then facilitate more efficient data mining practices and lead to better results. Intuitively, consumers of similar ages may suffer similar types of illness resulting in demands for similar drugs. Likewise, the clinical functions of certain drugs may be specific to a particular gender. Therefore, both age and gender can serve as good discriminative features.

The temporal nature of prescription data carries essential information. As mentioned earlier, the transactional PBS data is in the form of (*PhID, PrID,* (*Item, Cost*)*, Dos, Dop*). Collectively, each consumer can have a sequence of drugs prescribed over a certain period of time (e.g. quarterly or yearly) by simply appending them together chronologically. Though such a flat structure alleviates some workload for the later processing steps, it can consequently cause the loss of temporal information. Therefore, we decide to concatenate each consumer's temporal drug prescriptions

into a multi-set. Formally, let $I = \{i_n \mid n > 0\}$ be a set of all the available prescription drug items. A subset of these items can be organised into an itemset $X = \{j_m \mid j_m \in I, 0 \leq m \leq n\}$. The itemset symbolises a transaction of prescribed drugs. A consumer can accumulate a sequence of transactions over a specified period, which is denoted as $S = <X_k \mid k > 0>$. Eventually, a *sequence* is constructed and attached to each consumer.

The feature extraction and dimensionality reduction is an important research field. Our approach relies on simple yet effective intuitive knowledge. There exist various more sophisticated methods such as Principle Component Analysis (Kirby and Sirovich 1990), Linear Discriminant Analysis (Swets and Weng 1996) and eigenvalues based analysis (Nguten and Gopalkrishnan 2010). Each of them has benefits and disadvantages. We are planning to investigate their applicability in the future.

4.2 Cluster Builder

Clustering techniques can be used to efficiently explore the data and reduce noise. Such an explorative approach can effectively group similar consumers based on their sequential prescription activities. Mining on sequential prescription patterns is challenging, thus contributing to the combinatorial nature of the problem. A density-based clustering algorithm called *ApproxMAP* (Kum *et al.* 2003) is adopted for accomplishing the task. It favours discovering approximate and long patterns over short and trivial ones. The hierarchical edit-distance is utilised for calculating the logical distances between prescription sequences of different consumers. An edit can be of type *insertion*, *deletion* or *replacement*. The *cost* is defined as the minimum editing operations required to change one sequence to the other. For example, changing (*p, e, t*) into (*p, e, t, e, r*) incurs two-unit of *cost* (e.g. two *insertion* operations). The cost associated with a *replacement* is assumed to be less than or equal to the aggregated cost of an *insertion* and a *deletion*. Formally, we denote *IND*() as the cost for either an *insertion* or a *deletion* and *REPL*() as a *replacement* cost. The eventual edit-distance $D(S_1, S_2)$, between two sequences $S_1 = <X_n \mid n > 0>$ and $S_2 = < Y_m \mid m > 0>$, can be computed by dynamic programming using a set of recurrence relations (Kum *et al.* 2003). We can then derive a normalised distance $dist(S_1, S_2)$ by dividing $D(S_1, S_2)$ by $max(\|S_1\|, \|S_2\|)$ (e.g. the length of the longer sequence). The calculation of *REPL*() is based on Sørensen coefficient (Sørensen 1957) reflecting the normalised set difference.

Given a database of sequences S, the density of each sequence S_i is calculated based on its k-nearest neighbours (KNN) as follows:

$$density\ (S_i) = \frac{l}{\| S \| \times d}$$

where $d = max\{d_1, \ldots, d_k\}$ is the largest distance amongst S_i's KNN and $l = \|\{S_j \in S \mid dist(S_j, S_j) < d\}\|$ is the number of reachable neighbours. The clustering algorithm called Uniform kernel KNN clustering consisting of three steps as follows (Kum *et al.* 2003):

- **Step 1.** Initialise every sequence as a cluster.
- **Step 2.** Merge nearest neighbours based on the density of sequences.
- **Step 3.** Merge based on the density of clusters.

The logical output is a set of cohorts so that consumers having similar prescription patterns over the specified period are organised into the same cluster.

4.3 Model Constructor

The main idea behind the model constructor is to model clustered prescription sequences by the stochastic process of an HMM (Rabiner 1989). The HMM is a double embedded stochastic process with a finite set of states governed by a set of transition probabilities. It is widely used in various applications including bioinformatics, speech recognition, and genomics (Smyth 1994 and Srivastava *et al.* 2008). In contrast to typical classification methods, the HMM requires no labelled data and is relatively robust in the presence of noisy data.

A typical HMM has the following characteristics (Rabiner 1989):

- N is the number of states in the model. A set of N states is denoted as $H = \{h_j \mid i = 1, 2, 3, \ldots, N\}$. q_t represents the state at time instant t.
- M represents the number of unique observation symbols per state, which corresponds to the physical output of the system being modelled. The set of symbols is denoted as $V = \{v_k \mid k = 1, 2, 3, \ldots, M\}$.
- The state transition probability matrix $A = [a_{ij}]$, where $a_{ij} = P(q_{t+1} = h_j \mid q_t = h_i)$, $1 \leq i, j \leq N$ and $t > 0$. For all i and j, we have $a_{ij} > 0$ indicating that any state can be reached by any other state in a single step.
- The observation symbol probability matrix B = $[b_j(k)]$, where $b_j(k) = P(v_k \mid h_j)$, $1 \leq j \leq N$, $1 \leq k \leq M$ and $\sum_{1 \leq k \leq M} b_j(k) = 1$, $1 \leq j \leq N$.
- The initial state probability distribution $\pi_i = P(q_1 = h_j), 1 \leq i \leq N$.
- A sequence of observations $O = \{o_l \mid l = 1, 2, \ldots, R\}$, where each observation o_l is one of the symbols from V. R is the number of observations from sequence O.

A complete specification of an HMM model requires two model parameters (N and M) and three probability measures (A, B and π), which can be denoted as $\lambda = (A, B, \pi)$.

To build HMMs for capturing the common prescription behaviours from consumer cohorts, we incorporate the consumer-visiting-prescriber pattern into various states denoted as $H = \{h_1, ..., h_N\}$, $N > 0$. Each sate indicates a consumer visit to a unique prescriber (e.g. the first time visit), otherwise the visit is not considered as unique and can be mapped accordingly based on the *PrID* and the respective state. Therefore, a temporal visiting pattern can be organised into a sequence of various states, for instance, (h_1, h_1, h_2, h_3, h_4, h_3). Considering the large number of registered prescribers, we decided to focus on the intra consumer and prescriber relations (visits) aspects of the problem. Each prescriber visit can incur certain drug prescriptions, and we can model them as physical outputs from the set V of all observable outputs (e.g. all PBS drugs stored in the PBS database). Modelling all PBS drugs can make the model cumbersome and even infeasible due to two factors. Firstly, it can increase the training time dramatically. Secondly, derived probabilities associated with rare drugs can be extremely small, thus hampering the performance of HMM. Therefore, a compressed list of drugs (observations) is utilised, which covers all targeted drugs in table 1. Additionally, we establish a generic drug observation for capturing all non-targeted ones. Based on these two sets, a sample HMM is shown in figure 2.

The state h_0 is a dummy state representing the start and the end of a consumer's temporal pattern. Likewise, the v_0 is an artificial observation symbol for the sake of model integrity and consistency. The inference of HMMs also requires tunning parameters for three probability distributions (e.g. A, B and π), which is essentially an optimisation problem. Given an observation sequence O = $\{o_1, o_2, ..., o_t\}$, the objective is to estimate $\lambda = (A, B, \pi)$ so that $P(O|\lambda)$ is maximised. We adopt the well-known Baum-Welch algorithm (Rabiner 1989), which can be described as follows:

- **Input:** O = $\{o_1, o_2, ..., o_t\}$ and Δ.
- **Output:** $\lambda = (A, B, \pi)$
- **Step 1:** Let initial model be λ_0.
- **Step 2:** Compute new model λ based on λ_0 and observation sequence O.
- **Step 3:** If $log(P(O|\lambda)) - log(P(O|\lambda_0)) < \Delta$ go to Step 5.
- **Step 4:** Else set $\lambda_0 \leftarrow \lambda$ and go to Step 2.
- **Step 5:** Stop.

where we consider a uniform distribution model for initializing λ_0.

For each of the consumer cluster, a profile HMM is inferred. These profiles then can be used to evaluate new consumer prescription patterns, namely given a model $\lambda_1 = (A_1, B_1, \pi_1)$ and a sequence of observations O_{new}, we can compute the probability (Pr) that the sequence is produced by the model. For each new consumer, we can examine his/her prescription patterns against each HMM, from which an n-dimensional scoring vector ($Pr_1, Pr_2, ..., Pr_n$) can be derived. The forward and backward algorithm (Rabiner 1989) is adopted to efficiently compute each probability based on the given HMM. The number n is tuneable based on the results from the cluster builder.

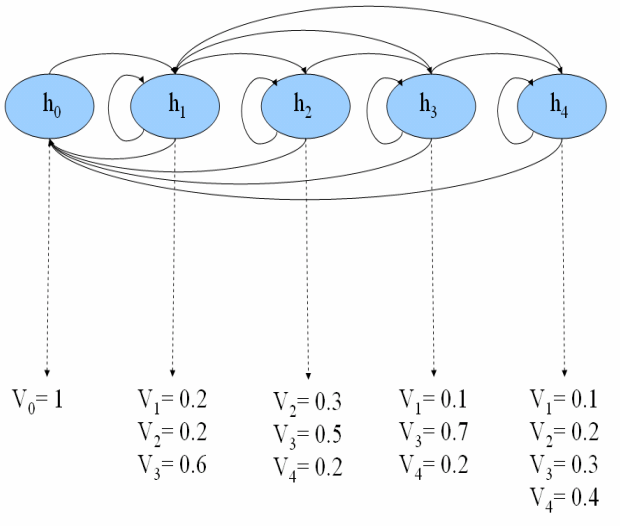


Figure 2: An auxiliary HMM

4.4 Outlier Detector

The model constructor provides a common ground for comparing prescription behaviors of different consumers. Medicare consumers, based on their temporal activities, are projected into an n-dimensional hyper-plane (e.g. n consumer cohorts and their respective HMMs), and each dimension implies a featured pattern encoded as an HMM. The spatial distance between any two consumers then can be computed. Various distance metrics (Cha 2007) are available for facilitating the task. Since each HMM score implicitly conveys the likelihood of a consumer being a member of the respective cohort, we adopt the City Block distance (Cha 2007) to augment the difference between two score vectors along all dimensions. Based on these spatial distances, we can adopt outlier detection techniques to automate the process of identifying potential fraudulent consumers. The LOCI (Local Correlation Integral) (Papadimitrious *et al.* 2003) is adopted as the underlying outlier detector, which produces outlier-ness scores rather than binary YES or NO answers. It is a density-based approach, which is effective on discovering micro-clusters (e.g. groups of outliers). Additionally, it uses statistical reasoning (such as standard deviation) to determine the outlier-ness.

We briefly describe some terms used in the LOCI and detailed algorithm description can be found in (Papadimitrious *et al.* 2003).

- r-neighbourhood of an object p_i: a set of objects within r distance of p_i.
- $n(p_i, \alpha r)$: the number of objects in the αr–neighbourhood of p_i.
- $\tilde{n}(p_i, r, \alpha)$: the average number of objects over all objects p in the r-neighbourhood of p_i.
- Multi-granularity deviation factor (MDEF) for p_i at radius r:

$$MDEF\ (p_i, r, \alpha) = 1 - \frac{n(p_i, \alpha r)}{\tilde{n}(p_i, r, \alpha)}$$

- Standard deviation of $n(p_i, \alpha r)$ over the r-neighbours:

$$\sigma_{\tilde{n}}(p_i, r, \alpha) \equiv \sqrt{\frac{\sum_{p \in N(p_i, r)} (n(p, \alpha r) - \tilde{n}(p_i, r, \alpha))^2}{n(p_i, r)}}$$

- Normalised deviation:

$$\sigma_{MDEF}(p_i, r, \alpha) = \frac{\sigma_{\tilde{n}}(p_i, r, \alpha)}{\tilde{n}(p_i, r, \alpha)}$$

The above terms essentially reflect the local integral correlation with respect to each projected data point (e.g. each consumer). Given a distance within $[r_{min}, r_{max}]$, we can compute the value of $MDEF(p_i, r, \alpha)$ with $\sigma_{MDEF}(p_i, r, \alpha)$ and evaluate how far they deviate from each other. Alternatively, both r_{min} and r_{max} can be replaced by the number of neighbours for comparison so that they can be dynamically identified.

The automated outlier detection process can efficiently narrow down the number of potential prescription shoppers, which enables more effective and targeted manual investigation by medical experts.

Parameter	Value Range
No. of clusters	3 to 5
No. of HMM states	50 to 100
No. of HMM observation symbols	42 to 100
No. of LOCI neighbours	10 to 20
Times of deviation	2 to 4

Table 2: parameter setting

5 Experimental Results

We have conducted extensive experimental studies to evaluate the performance of UNISIM. The system is implemented in C++ and OpenMP directives are inserted wherever possible to allow the parallelising of functional computations. A standalone workstation hosts the system, which has *8* CPU cores @ *2.66GHz* and *32 GB* of main memory. The underlying operating system is Ubuntu version *10.04*. A year worth of consumer data is extracted from the PBS and consumer directory databases for all eligible Australian residents. For the purpose of this paper, we further select consumers with certain demographics (e.g. male aged between *30* and *39*), which has a population of more than *300,000*. We randomly choose *1%* of the sampled consumers for building cluster cohorts and training the HMMs. The training dataset size is not only manageable but also sufficient for building consumer behavioural models. It is largely because we focus on extracting common prescription patterns. Additionally, HMMs are intrinsically robust to noisy data. Experimentally, the size of *1%* is reasonably well-balanced between under-training and over-fitting the HMMs. The training dataset is excluded from the proceeding testing.

Table 2 presents the set of parameters along with their value ranges required for UNISIM. Due to the departmental policy, we are restrained on revealing exact parameter values that have been used during our experiments. Instead, we briefly discuss how suitable values can be selected.

There is a trade-off for choosing the number of clusters. On one hand, more clusters can reflect finer-granularity of the underlying cohorts and HMMs. On the other hand, data is inevitably projected into higher-dimensional spaces, which can hamper the performance of the outlier detector. Additionally, building more clusters incurs more overheads in terms of time taken to train HMMs, data projection and spatial distance calculations. We find that a value between *3* and *5* is experimentally sufficient to produce promising results. The number of HMM states implies the number of unique prescribers that a consumer visits over a year. A value of *100* proves to be large enough, and it is almost impossible for a consumer to visit that many doctors over a year. It can be used as an upper bound for designing the number of HMM states. Throughout various experiments, a few consumers are identified as visiting a large number of unique doctors (e.g. *45*). Though these consumers represent an absolute minority, it is necessary to make the total number of allowable states large enough. As mentioned before, we utilise a compressed list of observation symbols (e.g. prescription drugs). All targeted drugs are uniquely denoted, whereas a generic observation symbol is defined to cover the rest of the drugs covered by the PBS. The list can be easily extended to target more drugs. The number of LOCI neighbors for comparison can be set to a range of values between *10* and *20*, which indicates coverage of *100* to *200* data objects. Finally, the value for times of deviation represents the tolerance towards considering an outlying data object. It can be tuned accordingly to facilitate varying investigation scope.

5.1 Detecting Known Outlying Consumers

Due to the unsupervised nature of UNISIM, we first examine its validity through a mock-up dataset. The dataset contains a sample of *1938* MA consumers along with a year worth of their prescription activities. These individuals are randomly picked from the studied demographics. Through previous studies (Ng *et al.* 2010 and Mendis *et al.* 2011), we obtain *20* identified outlying consumers resembling similar temporal behavioural patterns. They are deliberately injected into the sample dataset. Based on the HMM scores (e.g. the likelihoods of each consumer belonging to respective HMMs), we can treat each consumer as a data point in a *3*-dimensional hyper-plane. We plot these points in figure 3. For the sake of presentation, each HMM score is multiplied by *100,000*. As it can be seen, the majority of data points are crammed together posing challenges for visual analysis. Furthermore, outlying consumers are generally good at disguising themselves by emulating patterns of genuine consumers.

As it can be observed from table 3, all *20* outlying consumers are successfully detected by UNISIM. The outlier-ness score is calculated as the difference between *MDEF* and *3* times of σ_{MDEF}. The bigger the score, the more likely a data point can be classified as an outlier. A value of *0.57414* is big enough to indicate further investigation is warranted. Experimentally, we have found that the score is monotonically increasing. It is

worth noting that UNISIM is also capable of identifying a group of outliers. In this particular case, the group of known outlying consumers all have the same outlier-ness score, which is a very appealing feature. As table 3 shows, all *20* pre-identified fraudulent consumers belong to one group, which can be regarded as an outlier group. In terms of HMM scores, all these consumers are characterized with significantly small scores (e.g. close to *0*) implying that their behavioural patterns are highly irregular compared with common consumers (e.g. captured patterns during the HMMs training).

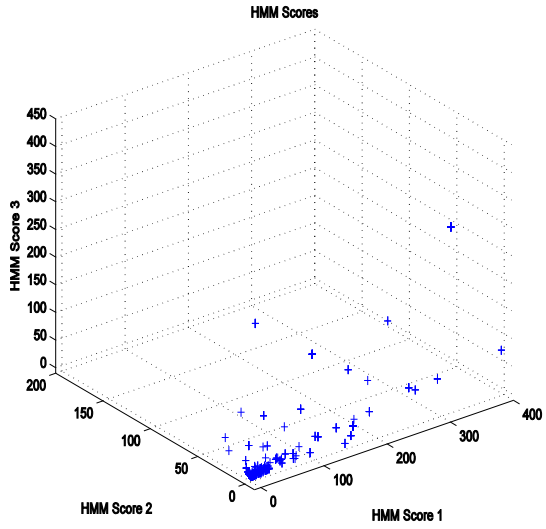


Figure 3: HMM score plot

De-identified Consumer ID	Outlier-ness Score
1	0.57414
2	0.57414
3	0.57414
4	0.57414
5	0.57414
6	0.57414
7	0.57414
8	0.57414
9	0.57414
10	0.57414
11	0.57414
12	0.57414
13	0.57414
14	0.57414
15	0.57414
16	0.57414
17	0.57414
18	0.57414
19	0.57414
20	0.57414

Table 3: known outlying consumers and their scores

5.2 Detecting Unknown Outlying Consumers

In this section, we study the generalised performance of UNISIM over unlabelled data. The dataset is compromised of temporal prescription data of *10,253* random consumers selected from the studied demographics sample. Considering the sheer quantity of all involved transactions, manual investigation is deemed to be infeasible.

Before delving into individual results comparison against each consumer, we focus on the group results generated by UNISIM so that we can observe some attractive features of it. Overall, there are *5,489* consumers identified by the system having a greater than *0* outlier-ness score. Such a large number shows that the real-life data is indeed very complex (e.g. irregular patterns from genuine consumers). Interestingly, some groups of outliers are amongst them (e.g. same score with similar prescription patterns). Table 4 shows *5* such groups along with their respective outlier-ness scores and number of members. Together they represent a population of *3,579* consumers or around *65%* of *5,489* consumers (outlier-ness score > *0*). By closely analysing the patterns in group *1*, we can observe that all its member consumers have one-off prescription over the chosen year. They can be easily filtered before further investigation. All four other groups have the similar properties. The capability of detecting micro-clusters of outliers allows us to quickly examine a group of consumers with similar temporal behaviours. Accordingly, we are able to effectively filter them out before further more costly investigation. For example, if we set the cut-off value to above *0.44901*, there is an instant reduction of *87%* leaving *693* potential suspects. It is very flexible to scope the investigation by the eventual business users.

Group ID	Outlier-ness Score	Number of Consumers
1	0.44901	1974
2	0.449009	656
3	0.159298	545
4	0.335087	270
5	0.155503	134

Table 4: representative outlier groups

De-identified Consumer ID	Outlier-ness Score
1	6.68674
2	6.65758
3	6.53518
4	6.46845
5	6.35029
6	5.97089
7	5.97089
8	5.71688
9	5.32089
10	5.27353

Table 5: top 10 individual outlying consumers

We further examine the top *10* individual consumers and their prescription patterns, which are included in table 5. On average, *4* different doctors have been visited by these consumers. The transaction records reveal that some extreme cases have multiple visits to different doctors on one day. By looking at their prescription

drugs, we can notice that the majority of them are targeted ones (i.e. listed in table 1 as suggested by subject matter experts). With such a combination, we can confidently classify the consumer as suspicious and pass the information onto the compliance division for further investigation.

6 Conclusion and Future Work

The main focus of the paper is unsupervised fraud detection particularly targeting MA claimants with potential prescription shopping behaviours. We propose a data-driven system, called UNISIM, for tackling the problem. UNISIM is comprised of comprehensive data mining components including feature extractor, cluster builder, model constructor and outlier detector for effective and efficient analysis of MA consumer data. Importantly, we provide effective HMM for encoding essential knowledge into UNISIM enabling it to automate the fraud detection process. We have demonstrated the effectiveness of UNISIM on detecting potential non-compliant consumers using real-life health care data. We need to emphasise that UNISIM itself serves as a complementary tool to assist with the subject matter experts. For consumers identified as obtaining large quantities of PBS medications, we are still reliant on the subject matter experts to decide if they have behaved fraudulently.

In the future, we are planning to experiment with different techniques or algorithms other than the ones that have been implemented. Currently, complex real-life interactions, either explicit or implicit, are not the focus of UNISIM. Capturing these coupled and intriguing relations are technically challenging yet can be beneficial especially for identifying more professional and organised fraud. The HMM can be built differently (e.g. to introduce contradictions). We expect to design and implement other stochastic models for evaluating consumer patterns.

7 Acknowledgements

The authors wish to thank Dr. David Jeacocke for his helpful clinical insights. We would also like to thank Leonie Greenwood, Thach Van, Alex Dolan, Rory King, and Paul Cowan for providing timely management support for this paper. Last but not least, we are grateful for invaluable comments from both reviewers for making any improvement on the paper possible.

References

Cha, S.H. (2007): Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions. *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 1, no. 4, pp300-307, 2007.

Kirby, M. and Sirovich, L. (1990): Application of the Karhunen-loeve procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp103-108, 1990.

Kum, H.C., Pei, J., Wang, W. and Duncan, D (2003): ApproxMAP: Approximate Mining of Consensus Sequential Patterns. *Proc. 3rd SIAM International Conference on Data Mining*, San Francisco, California, USA, pp311-315.

Medicare Australia (2011): Medicare Australia Annual Report 2009-2010. http://www.humanservices.gov.au/spw/corporate/publications-and-resources/annual-report/medicare/index.html. Accessed 29 July 2011.

Mendis, B.Sumudu.U., Murray, D.W., Sutinen, A., Tang, M.J. and Hu, Y.S. (2011): Enhancing the Identification of Anomalous Events in Medicare Consumer Data Through Classifier Combination. *Proc. 6th International Workshop on Chance Discovery*, Barcelona, Spain, pp39-44, Springer Press.

Ng, K.S., Shan, Y., Murray, D.W., Sutinen, A., Schwarz, B., Jeacocke, D. and Farrugia. J. (2010): Detecting Non-compliant Consumers in Spatial-Temporal Health Data: A Case Study from Medicare Australia. *Proc. IEEE International Conference on Data Mining Workshops*, Sydney, Australia, pp613-622, IEEE Press.

Nguyen, H.V. and Gopalkrishnan, V. (2010): Feature Extraction for Outlier Detection in High-Dimensional Spaces. *Proc. 4th Workshop on Feature Selection in Data Mining*, Hyderabad, India, pp64-73.

Papadimitriou, S., Kitagawa, H., Gibbons, P.B. and Faloutsos, C. (2003): LOCI: Fast Outlier Detection Using the Local Correlation Integral. *Proc. 19th International Conference on Data Engineering (ICDE'03)*, California, USA, pp.315, 2003

Rabiner, L.R. (1989): Investigating Hidden Markov Models Capabilities in Anomaly Detection. *Proc. IEEE,* vol. 77, no. 3, pp357-286, 1989.

Smyth, P. (1994): Markov Monitoring with Unknown States. *IEEE Journal on Selected Areas in Communications*, vol. 12, no. 9, pp1600-1612, 1994.

Sørensen, T. (1957): A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. *Biologiske Skrifter/Kongelige Danske Videnskabernes Selskab*, vol. 5, no. 4, pp1-34, 1957.

Srivastava, A., Kundu, A., Sural, S. and Majumdar, A.K. (2008): Credit Card Fraud Detection Using Hidden Markov Model. *IEEE Transactions on Dependable and Secure Computing*, vol. 5, no. 1, pp37-48, 2008.

Swets, D.L. and Weng, J.Y. (1996): Using Discriminant eigenfeatures for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp831-836, 1996.

Prescriber-Consumer Social Network Analysis for Risk Level Re-estimation based on an Asymmetrical Rating Exchange Model

Yingsong Hu D. Wayne Murray Yin Shan Alison Sutinen
B. Sumudu U. Mendis MingJian Tang

Strategic Data Mining Section
Department of Human Services Australia
134 Reed Street, Greenway, ACT 2900
Email: yingsong.hu@humanservices.gov.au or wayne.murray@humanservices.gov.au

Abstract

In this paper, we present a novel approach to re-estimate the risk level of prescribers and consumers (doctors and patients) that were previously evaluated by various independent Risk Analysis Systems (RAS). This is achieved by taking into consideration social network structure between prescribers and consumers. A mathematical model, called Asymmetrical Rating Exchange Model (AREM) is proposed to describe the mutual influences between prescribers and consumers from a social network perspective and based on this model an algorithm is derived to re-estimate the suspicion level of each entity in the community considering both the pre-evaluated rating and the network structure. Additionally, by comparing the pre-evaluated rating and the re-estimated risk level, under-rated entities can be identified and further assessed. Experimental results are also presented, showing that the proposed approach can effectively detect the entities that have strong connections to high risk entities, but previously been rated low suspicious by independent Risk Analysis Systems (RAS).

Keywords: Social network analysis, fraud detection, risk level estimation.

1 Introduction

Medicare Australia, a master program with the Department of Human Services, helps deliver health programs to Australians through the Medicare Benefit Scheme (MBS) and the Pharmaceutical Benefit Scheme (PBS). According to Medicare Australia 2009-10 annual report (Medicare Australia 2010), over 308 million MBS services and over 197 million PBS transactions were processed. Approximately A$40 billion was paid in benefits in the 2009-10 financial year. These programs benefit the Australian community as a whole. Thus, it is paramount to maintain the integrity and availability of these programs to ensure government funding is spent appropriately. According to the National Compliance Program 2010-2011 (Medicare Australia 2011), more than A$10.29 million of incorrect payments to providers, pharmacists and members of the public was identified in 2009-2010. The National Compliance program outlines the key activities the Department of Human Services will undertake to protect its programs from the occurrence of inappropriate practice and fraud. Several data mining systems have been developed either in-house or through external engagement at the Medicare Australia for identifying doctors, pharmacies and consumers that engage in fraudulent or inappropriate billing activities (Yamanishi et al. 2004, Pearson et al. 2006, Shan et al. 2008, 2009, Ng et al. 2010, Mendis et al. 2011). The final result for each system is a risk rating per entity which, on its own, reflects a certain perspective of the MBS and PBS programs. Each of these systems targets specific entities. For example, Prescriber RAS applies local outlier factor(Breunig et al. 2000) similar to the one presented in paper (Shan et al. 2009) to detect the prescribers who prescribe significantly different from their peers with the support of *Rlof* package (Hu et al. 2011), while Consumer RAS (Ng et al. 2010, Mendis et al. 2011) adopts a series of technologies and models, e.g. huff, markov and RFPOI, to target risky patients who are likely to have a behaviour of prescription shopping. These systems are individually designed and customised to specifically target one type of entities based on feature comparison or pattern recognition. Thus, it is beyond the scope of the systems to consider the connections and interactions between the entities. Intuitively, for example, a prescriber who is frequently visited by larger number of high risky consumers with high volume of prescriptions should be considered as a suspicious prescriber, but not all of these prescribers could be identified by Prescriber RAS due to the different focuses of these systems. This is the motivation for us to seek an appropriate approach to reestimating risk levels of entities based on both connections between entities and the suspiciousness originally evaluated by independent Risk Analysis Systems (RAS). In this paper, we focus on the analysis of the prescriber-consumer network, which can be mathematically modelled as a two-mode network or called bipartite network, and the input to this network are the outputs from the Prescriber RAS(Shan et al. 2009) and Consumer RAS(Ng et al. 2010, Mendis et al. 2011), two data mining systems developed and currently being used in Medicare Australia.

Social network analysis (SNA) takes the importance of relationships among interacting units into consideration. The social network perspective encompasses theories, models, and applications that are expressed in terms of relational concepts or processes(Wasserman & Faust 1994). Applying SNA to fraud detection and risk analysis areas has also attracted great interests from researchers as well as intelligent system developers. As fraudsters or high risk entities usually interact with each other and cer-

tain fraud behaviour could not be done without the interactions with other entities (e.g. a prescription shopper can not obtain the drugs in demand without visiting prescribers and filling the prescription in a pharmacy). Therefore, fraud detection, prediction or risk assessment could be achieved by analysing the networks of the suspicious entities. The idea of this paper is to bring these perspectives together and propose a holistic approach by considering the interactions between the entities of different perspectives. There have been a few papers previously published that try to identify various types of frauds using similar concepts and ideas. Under the observation that fraudsters seldom change their calling habits and are often closely linked to other fraudsters, link analysis is utilised for telecom fraud detection (Cortes et al. 2001). Neville et al. (2005) also used relational knowledge to discover misconduct among securities brokers. However these literatures focus on single-mode networks. i.e. the vertices (nodes) are intrinsically playing the same role in the network. For example, the nodes in the telecommunication network are all equally regarded as consumers. Within the prescriber-consumer network we discuss in this paper, however, each vertex is within a certain class, which implies that depending on the identification of the entities, every node plays a different role as a prescriber or consumer in the network, and hence the impacts from one class to another may be considered asymmetrical.

Considering the particularity of the prescriber-consumer network mentioned above, in Section 2 a model called Asymmetric Rating Exchange Model (AREM) is proposed to mathematically describe the influence between prescribers and consumers in terms of risk ratings. Section 3 presents the overall end-to-end approach to achieve the re-estimation of risk levels in the prescriber-consumer network based on the model proposed in Section 2. Experimental results are discussed in Section 4 demonstrating the effectiveness of the Asymmetrical Rating Exchange Model (AREM) and the algorithm proposed. Section 5 concludes this paper.

2 Mathematical Model and Solution

As mentioned in the previous section, two Risk Analysis Systems (called Prescriber RAS and Consumer RAS) adopting different technologies were independently developed to target suspicious prescribers and consumers, respectively. The outputs of these two system are the suspicious ratings of the entities assessed by the systems. Meanwhile, in Pharmaceutical Benefit Scheme (PBS) database maintained by Medicare Australia, every transaction records a prescriber and a consumer's identification as attributes, which builds up the links between prescribers and consumers from a network point of view. In the prescriber-consumer network, each vertex represents an entity which can be either a prescriber or a consumer, and the activity for a consumer to get prescriptions from a prescriber forms an edge. Considering the data we are using for the analysis is always for a certain time frame (e.g. a month or a quarter, etc.), we can simply use the total number of original prescriptions that a consumer gets from a prescriber within the time frame to quantify the frequency of interactions, which is here defined as the weight of the edge between the vertices of prescriber and consumer in the network.

The task of the new system called Unified RAS proposed in this paper is to take the outputs of two independent data mining systems as a pre-evaluation of the risk rating for every entity and further bring the interactions between prescribers and consumers together to work out a re-estimation of the suspicious levels from a network analysis perspective. To realise this idea, a straight forward thought is that, for example, the more often a prescriber prescribes for high risk consumers, the more suspicious the prescriber would be, because in this case, the prescriber would look like a supplier of prescriptions to a number of high risk consumers. On the other hand, the more frequently a consumer visits high risk doctors (especially for those who do not have primary family doctors(Ng et al. 2010)), the more chance that the consumer has a higher risk. We can also describe this fact in a way the entities in the network 'exchange' their suspicions depending on the frequency of interactions, which is the frequency of consultations in our application.

2.1 Asymmetrical Rating Exchange Model (AREM)

Based on the idea of risk exchange between neighbours in the network, assuming there are totally K vertices denoted as v_k $(k = 1, 2, ..., K)$ in the network, we may initially express the estimated rating $\hat{r}_k$ of entity v_k modulated by a linear factor λ_k as a weighted linear combination of the estimated rating of its neighbours in the network :

$$\lambda_k \cdot \hat{r}_k = \sum_{n \in \mathbf{N}(v_k)} w_{k,n} \cdot \hat{r}_n, (k = 1, 2, ..., K) \qquad (1)$$

where $\mathbf{N}(v_k)$ represents the set of the neighbours of vertex v_k and $w_{k,n}$ is the frequency of interactions between entity v_n and v_k. i.e. the number of original scripts obtained in a certain time frame as previously explained.

Further, as mentioned in Section 1, prescribers and consumers play very different roles in the network and the influence from one category to the other could be very different as well. The scalar λ_k introduced in Eq.(1) should vary with the category of entity v_k. i.e.

$$\lambda_k = \begin{cases} \lambda_c & v_k \in \mathcal{C} \\ \lambda_p & v_k \in \mathcal{P} \end{cases} \qquad (2)$$

where $\mathcal{C}$ denotes the set of consumers and $\mathcal{P}$ is the set of prescribers in the network.

Representing the model in a form of matrix, the re-estimated vector $\hat{\mathbf{r}} = (v_1, v_2, ..., v_K)^{\mathbf{T}}$ needed to satisfy following equation[1]:

$$\mathbf{W} \times \hat{\mathbf{r}} = \mathbf{\Lambda} \times \hat{\mathbf{r}} \qquad (3)$$

where $\mathbf{W}$ is the weighted adjacency matrix of the network. i.e.

$$\mathbf{W} = \left(w_{k,n} \right)_{m \times m} \qquad (4)$$

in which $m = |\mathcal{C}| + |\mathcal{P}|$, where $|\cdot|$ represents the cardinality of a set. $\mathbf{\Lambda}$ is a diagonal matrix whose k-th element on its diagonal, λ_k is defined in Eq.(2).

Without loss of generality, we can also assume that vertices $v_i \in \mathcal{C}$ when $1 \leq i \leq |\mathcal{C}|$ and $v_i \in \mathcal{P}$ when $|\mathcal{C}| < i \leq |\mathcal{C}| + |\mathcal{P}|$. In this case, the model could be more visually expressed as below:

$$\begin{pmatrix} 0 & \mathbf{A} \\ \mathbf{A^T} & 0 \end{pmatrix} \times \begin{pmatrix} \hat{\mathbf{r}}_\mathbf{c} \\ \hat{\mathbf{r}}_\mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{\Lambda_c} & 0 \\ 0 & \mathbf{\Lambda_p} \end{pmatrix} \times \begin{pmatrix} \hat{\mathbf{r}}_\mathbf{c} \\ \hat{\mathbf{r}}_\mathbf{p} \end{pmatrix} \qquad (5)$$

[1] $\mathbf{T}$ denotes transpose of a vector or matrix

where $\mathbf{A}$ is a $|\mathcal{C}|$ by $|\mathcal{P}|$ matrix, and $\hat{\mathbf{r_c}}$ and $\hat{\mathbf{r_p}}$ are column vectors representing the re-estimated rating for consumers and prescribers with the dimension of $|\mathcal{C}|$ and $|\mathcal{P}|$, respectively. Similarly, $\mathbf{\Lambda_c}$ and $\mathbf{\Lambda_p}$ are diagonal matrices of λ_c and λ_p with size of $|\mathcal{C}|$ by $|\mathcal{C}|$ and $|\mathcal{P}|$ by $|\mathcal{P}|$, respectively. Meanwhile, we need to consider the frequency of interactions between entities. i.e. the structure of the network, but also have to consider the pre-evaluated ratings output from the other two systems, because these two systems should perform well with a reasonable confidence based on the results reported (Ng et al. 2010, Mendis et al. 2011). This implies that we should find a re-estimated vector $\hat{\mathbf{r}}$ that is as 'close' to the pre-evaluated vector $\hat{\mathbf{p}}$ as possible. Similar to the way of dividing $\hat{\mathbf{r}}$ into $\hat{\mathbf{r}}_\mathbf{c}$ and $\hat{\mathbf{r}}_\mathbf{p}$, we use $\hat{\mathbf{p}}_\mathbf{c}$ and $\hat{\mathbf{p}}_\mathbf{p}$ that comprise $\mathbf{p}$ to denote the pre-evaluated ratings to consumers and prescribers, respectively. The initial task is then turned into an optimisation problem described as below:

$$\begin{aligned} &\textit{objective}: \\ &\mathbf{argmin}(||\hat{\mathbf{r_c}}|| - ||\mathbf{p_c}||)^2 + (||\hat{\mathbf{r_p}}|| - ||\mathbf{p_p}||)^2 \quad (6) \\ &\textit{subject to}: \\ &\mathbf{W} \times \hat{\mathbf{r}} = \mathbf{\Lambda} \times \hat{\mathbf{r}} \text{ and } \hat{\mathbf{r}} \geq 0 \quad (7) \end{aligned}$$

where $||\cdot||$ denotes the norm of a vector.

2.2 Solution to the problem

It would be very interesting to note that if we consider a special situation where $\lambda_c = \lambda_p$ in matrix $\mathbf{\Lambda}$, Eq.(3) will mathematically become a well-known eigenvector centrality problem first introduced by Bonacich (1972) as a novel centrality measure, after which the eigenvector centrality for bipartite graph is specifically discussed further(Bonacich 1991, Borgatti & Everett 1997). However, the major differences of our problem from the eigenvector centrality (EC) problem are at first, the eigenvector centrality generally discussed did not take λ as a variable for different classes of the vertex, which are introduced in our case to cater for asymmetrical impact between consumers and prescribers, as well as, to compensate the influence of different risk rating scales coming from two independent systems; secondly, we also have to take the pre-evaluated rating into our consideration for the re-estimation whereas EC problem only needs to look at the structure of the network.

Firstly, we only consider the model proposed in Eq.(5), which can be also expressed in following way:

$$\begin{aligned} \mathbf{A} \times \hat{\mathbf{r}}_\mathbf{p} &= \lambda_c \hat{\mathbf{r}}_\mathbf{c} \quad (8) \\ \mathbf{A}^\mathbf{T} \times \hat{\mathbf{r}}_\mathbf{c} &= \lambda_p \hat{\mathbf{r}}_\mathbf{p} \quad (9) \end{aligned}$$

As both λ_c and λ_p are scalars, by multiplying λ_p to both sides of Eq.(8), we can have:

$$\mathbf{A} \times (\lambda_p \hat{\mathbf{r}}_\mathbf{p}) = \lambda_c \lambda_p \hat{\mathbf{r}}_\mathbf{c} \quad (10)$$

and then substituting $\lambda_p \hat{\mathbf{r}}_\mathbf{p}$ with Eq.(9), we can get:

$$\mathbf{A} \times (\mathbf{A}^\mathbf{T} \times \hat{\mathbf{r}}_\mathbf{c}) = \lambda_c \lambda_p \hat{\mathbf{r}}_\mathbf{c} \quad (11)$$

i.e.

$$(\mathbf{A} \times \mathbf{A}^\mathbf{T}) \times \hat{\mathbf{r}}_\mathbf{c} = \lambda_c \lambda_p \hat{\mathbf{r}}_\mathbf{c} \quad (12)$$

similarly, we will also have:

$$(\mathbf{A}^\mathbf{T} \times \mathbf{A}) \times \hat{\mathbf{r}}_\mathbf{p} = \lambda_c \lambda_p \hat{\mathbf{r}}_\mathbf{p} \quad (13)$$

Because $\lambda_c \lambda_p$ is a scalar in Eq.(12), Eq.(13), $\hat{\mathbf{r}}_\mathbf{c}$ and $\hat{\mathbf{r}}_\mathbf{p}$ can be then resolved by using the approach to the typical eigenvector centrality problem, for which Bonacich (1991) pointed out that $\lambda_c \lambda_p$ should be the largest eigenvalue of matrix $\mathbf{A} \times \mathbf{A}^\mathbf{T}$ and $\mathbf{A}^\mathbf{T} \times \mathbf{A}$, while $\hat{\mathbf{r}}_\mathbf{c}$ and $\hat{\mathbf{r}}_\mathbf{p}$ would be the only positive eigenvectors corresponding to $\lambda_c \lambda_p$. Moreover, it is obvious that if vector $\hat{\mathbf{r}}_\mathbf{c}$ satisfies Eq.(12), then $\hat{\mathbf{r}}_\mathbf{c}$ multiplied by any scalar also satisfies the equation, which means that matrix $\mathbf{A}$ will only determine the angle of the vector $\hat{\mathbf{r}}_\mathbf{c}$ and $\hat{\mathbf{r}}_\mathbf{p}$, but the norm of them could be still arbitrary. Meanwhile, if we further investigate the objective function expressed by Eq.(6), we can see that the function could achieve a minimum value 0 if and only if $||\hat{\mathbf{r_c}}|| = ||\mathbf{p_c}||$ and $||\hat{\mathbf{r_p}}|| = ||\mathbf{p_p}||$. Therefore, the original optimisation problem becomes a definite problem, which is to find values of $\hat{\mathbf{r_c}}$ and $\hat{\mathbf{r_p}}$ that satisfy:

$$\begin{aligned} \mathbf{A} \times \hat{\mathbf{r}}_\mathbf{p} &= \lambda_c \hat{\mathbf{r}}_\mathbf{c} \quad (14) \\ \mathbf{A}^\mathbf{T} \times \hat{\mathbf{r}}_\mathbf{c} &= \lambda_p \hat{\mathbf{r}}_\mathbf{p} \quad (15) \\ ||\hat{\mathbf{r_c}}|| &= ||\mathbf{p_c}|| \quad (16) \\ ||\hat{\mathbf{r_p}}|| &= ||\mathbf{p_p}|| \quad (17) \\ \hat{\mathbf{r_c}} > 0 \quad &\text{and} \quad \hat{\mathbf{r_p}} > 0 \quad (18) \end{aligned}$$

and as discussed above, there is one and only one solution for the matrix $\mathbf{A}$, $\mathbf{p_c}$ and $\mathbf{p_p}$ given, which is more explicitly expressed as below:

$$\begin{aligned} \hat{\mathbf{r_c}} &= ||\mathbf{p_c}|| \cdot \mathbf{v}(\mathbf{A} \times \mathbf{A}^\mathbf{T}) \quad (19) \\ \hat{\mathbf{r_p}} &= ||\mathbf{p_p}|| \cdot \mathbf{v}(\mathbf{A}^\mathbf{T} \times \mathbf{A}) \quad (20) \end{aligned}$$

where $\mathbf{v}(\cdot)$ and $\lambda(\cdot)$ denotes the unit (normalised) principle eigenvector and principle eigenvalue of a matrix, respectively. To find out the solution of λ_c and λ_p, from Eq.14, we could get:

$$(\mathbf{A} \times \hat{\mathbf{r}}_\mathbf{p})^\mathbf{T} \times \mathbf{A} \times \hat{\mathbf{r}}_\mathbf{p} = (\lambda_c \hat{\mathbf{r}}_\mathbf{c})^\mathbf{T} \times \lambda_c \hat{\mathbf{r}}_\mathbf{c} \quad (21)$$

i.e.

$$\hat{\mathbf{r}}_\mathbf{p}^\mathbf{T} \times \mathbf{A}^\mathbf{T} \times \mathbf{A} \times \hat{\mathbf{r}}_\mathbf{p} = \lambda_c^2 (\hat{\mathbf{r}}_\mathbf{c}^\mathbf{T} \times \hat{\mathbf{r}}_\mathbf{c}) \quad (22)$$

Substituting Eq.(13) into this equation, we will have:

$$\lambda_c \lambda_p (\hat{\mathbf{r}}_\mathbf{p}^\mathbf{T} \times \hat{\mathbf{r}}_\mathbf{p}) = \lambda_c^2 (\hat{\mathbf{r}}_\mathbf{c}^\mathbf{T} \times \hat{\mathbf{r}}_\mathbf{c}) \quad (23)$$

Because elements in $\mathbf{A}$, $\hat{\mathbf{r_c}}$ and $\hat{\mathbf{r_p}}$ are non-negative, obviously λ_c and λ_p are positive, and hence, the equation above tells that:

$$\lambda_p ||\hat{\mathbf{r_p}}||^2 = \lambda_c ||\hat{\mathbf{r_c}}||^2 \quad (24)$$

From Eq.(19) and Eq.(20), we can see $||\hat{\mathbf{r_c}}|| = ||\hat{\mathbf{p_c}}||$ and $||\hat{\mathbf{r_p}}|| = ||\hat{\mathbf{p_p}}||$, then

$$\lambda_p ||\hat{\mathbf{p_p}}||^2 = \lambda_c ||\hat{\mathbf{p_c}}||^2 \quad (25)$$

Meanwhile, from Eq.(12) and Eq.(13), we have:

$$\lambda_c \lambda_p = \sqrt{\lambda(\mathbf{A} \times \mathbf{A}^\mathbf{T})} = \sqrt{\lambda(\mathbf{A}^\mathbf{T} \times \mathbf{A})} = \sqrt{\lambda} \quad (26)$$

where $\lambda(\cdot)$ represents the principle eigenvalue of a matrix. Thus, we can have the solution to λ_c and λ_p from Eq.(25) and Eq.(26):

$$\begin{aligned} \lambda_c &= \sqrt{\lambda} \cdot \frac{||\mathbf{p_p}||}{||\mathbf{p_c}||} \quad (27) \\ \lambda_p &= \sqrt{\lambda} \cdot \frac{||\mathbf{p_c}||}{||\mathbf{p_p}||} \quad (28) \end{aligned}$$

As a summary, the following equation is the derived solution to the problem stated in Eq.(7).

$$\begin{aligned} \hat{\mathbf{r_c}} &= ||\mathbf{p_c}|| \cdot \mathbf{v}(\mathbf{A} \times \mathbf{A^T}) \\ \hat{\mathbf{r_p}} &= ||\mathbf{p_p}|| \cdot \mathbf{v}(\mathbf{A^T} \times \mathbf{A}) \\ \lambda_c &= \sqrt{\lambda} \cdot \frac{||\mathbf{p_p}||}{||\mathbf{p_c}||} \\ \lambda_p &= \sqrt{\lambda} \cdot \frac{||\mathbf{p_c}||}{||\mathbf{p_p}||} \end{aligned} \tag{29}$$

2.3 Discussion of AREM

Before we move on to other practical issues we will need to deal with, we would like to discuss this model and solution a bit further. When $||\mathbf{p_c}|| = ||\mathbf{p_p}||$, the problem will become mathematically identical to the problem of eigenvector centrality for a two-mode network, which is in depth discussed by Borgatti & Everett (1997), i.e. the bipartite graph eigenvector centrality problem can be regarded as a special case when presuming that these two classes are symmatrically impacting each other, and hence the overall ratings of these two classes are the same. i.e. $||\mathbf{p_c}|| = ||\mathbf{p_p}||$. Without loss of generality when vector $\mathbf{p}$ is an unit vector ($||\mathbf{p}|| = 1$), we will have

$$||\mathbf{p}|| = \sqrt{||\mathbf{p_c}||^2 + ||\mathbf{p_p}||^2} = 1 \tag{30}$$

Thus,

$$||\mathbf{p_c}|| = ||\mathbf{p_p}|| = \frac{1}{2} \tag{31}$$

In an ideally balanced case where every node is equally centralised in the class the node belongs to, the centrality of each node in the class will be $\sqrt{\frac{1}{2n}}$, where n is the number of the vertices in the class. Because this figure is derived from the ideally balance situation, it could be regarded as an expectation of centralities, which theoretically explained why it can be used for the purpose of normalisation for bipartite network centrality scores proposed in Borgatti & Everett (1997). This also confirmed that mathematically the model proposed in this paper is actually a generalised asymmetrical version over the traditional two-mode network eigenvector centrality problem.

3 Practical Issue and Resolution

In the previous section, we mathematically discussed the solution to the rating re-estimation problem defined by Eq.(7). However, it is still almost impractical for us to work out the result directly from Eq.(29), due to the tremendous size of the network we are working on. If we consider the full prescriber-consumer network for 2009-2010 financial year, there would be 9,187,790 consumers and 89,812 prescribers in Australia. This implies that we need to at least work out the eigenvector for a matrix of 9,187,790 by 9,187,790 million, which is an impractical task for a workstation (Mac Pro v4.1 equipped with dual 2.66 GHz Quad-Core Intel Xeon and 16GB DDR3) to calculate in a timely manner. To reduce the size of the network for analysis, we constructed our network with only the consumers who had at least one flag from Consumer RAS (Ng et al. 2010, Mendis et al. 2011) and included only those prescribers who prescribed at least one original script to any of the consumers, i.e. we think it is a reasonable practice to ignore those consumers who are considered as zero risk by Consumer RAS and those providers who only ever prescribed for these very low risk consumers. This simplification successfully shrinks the entire network down to a network with 16,829 consumers, 20,319 prescribers and 102,719 edges. This simplified network is still a very large scale network, to which and our attempt to work out eigenvector of a 20,319 by 20,319 matrix by using R failed due to the limited memory in the workstation and the complexity of the eigenvector calculation. Meanwhile, if we consider the real situation of the network, some of the consumers might be travelling and hence these consumers may occasionally visit some prescribers in other states, but these occasional connections should not be considered as a suspicious behaviour and hence could be regarded as a noise in practice. i.e. from network analysis perspective, it would be better to consider the communities in the prescribers-consumers' network instead of the entire graph. In other words, based on both the real situation and the technical considerations, we would like to further break the suspicious network down to communities and perform re-estimation at the community level.

Communities in a graph are subsets of vertices within which vertex-vertex connections are dense but between which connections are less dense(Girvan & Newman 2002). If we divide the entire network into communities with the edges among the community members retained, each community can be expressed as a subgraph with relatively dense connection within it. Let's assume the overall network G can be divided into m communities expressed by subgraph $G_j, j = 1, 2, ...m$. With each subgraph G_j, we can certainly apply Eq.(29) to obtain solutions of $\hat{\mathbf{r_{cj}}}$, $\hat{\mathbf{r_{pj}}}$, λ_{cj} and λ_{pj}, where footnote j denotes the results for community j. As mentioned before, the removed edges during community detection processes should be those unfrequent visits. i.e. edges with low weights, and hence the relative distribution (the angle) of re-estimated rating vector should be very close to the theoretical value worked out directly from the entire network, i.e. $angle(\hat{\mathbf{r_{cj}}}) \approx angle((\hat{\mathbf{r_c}})_\mathbf{j})$ and $angle(\hat{\mathbf{r_{pj}}}) \approx angle((\hat{\mathbf{r_p}})_\mathbf{j})$ ($angle(\cdot)$ denotes the angle of a vector). However, because the overall structures of communities are very diverse across subgraphs and different from the original network G, we have to somehow work out a way to determine the norm of $\hat{\mathbf{r_{cj}}}$ and $\hat{\mathbf{r_{cj}}}$ for each community to make the final re-estimated result consistent across subgraphs.

We still remember the initial intention of λ_c and λ_p is to deal with asymmetrical influence from one class to the other. Hence, the contrast between λ_p and λ_c .i.e. $\frac{\lambda_p}{\lambda_c}$ should be consistent across the whole network in our application. Let's define:

$$q = \frac{\lambda_p}{\lambda_c} = \frac{||\mathbf{p_c}||^2}{||\mathbf{p_p}||^2} \tag{32}$$

then we would like to see all the communities have the same q for consistency. Therefore, the λ_{cj} and λ_{pj} for community j can be calculated as below:

$$\lambda_{cj} = \sqrt{\frac{\lambda(\mathbf{A_j} \times \mathbf{A_i^T})}{q}} \tag{33}$$

$$\lambda_{pj} = \sqrt{q \cdot \lambda(\mathbf{A_j} \times \mathbf{A_i^T})} \tag{34}$$

Meanwhile, each community should also hold the relationship expressed in Eq.(24). i.e.

$$\lambda_{pj}||\hat{\mathbf{r_{pj}}}||^2 = \lambda_{cj}||\hat{\mathbf{r_{cj}}}||^2 \tag{35}$$

which is

$$\frac{||\hat{\mathbf{r_{cj}}}||}{||\hat{\mathbf{r_{pj}}}||} = \sqrt{\frac{\lambda_{pj}}{\lambda_{cj}}} = \sqrt{q} \quad (36)$$

As mentioned before, we would like to have a values of $||\hat{\mathbf{r_{cj}}}||$ that is as close to $||\hat{\mathbf{p_{cj}}}||$ as possible, and the same for $||\hat{\mathbf{r_{pj}}}||$ and $||\hat{\mathbf{p_{pj}}}||$, under the relationship expressed in Eq.(36). That means that we want to minimise following objective function:

$$f_{obj} = (||\hat{\mathbf{r_{cj}}}|| - ||\hat{\mathbf{p_{cj}}}||)^2 + (||\hat{\mathbf{r_{pj}}}|| - ||\hat{\mathbf{p_{pj}}}||)^2 \quad (37)$$

Substituting with Eq.(36), the objective function could be expressed as:

$$\begin{aligned} f_{obj} &= (\sqrt{q}||\hat{\mathbf{r_{pj}}}|| - ||\hat{\mathbf{p_{cj}}}||)^2 + (||\hat{\mathbf{r_{pj}}}|| - ||\hat{\mathbf{p_{pj}}}||)^2 \\ &= (1+q)||\hat{\mathbf{r_{pj}}}||^2 - 2(\sqrt{q}||\hat{\mathbf{p_{cj}}}|| + ||\hat{\mathbf{p_{pj}}}||)||\hat{\mathbf{r_{pj}}}|| \\ &\quad + ||\hat{\mathbf{p_{cj}}}||^2 + ||\hat{\mathbf{p_{pj}}}||^2 \quad (38) \end{aligned}$$

which is a very typical quadratic function minimisation problem. i.e. the minimum value of the function can be achieved when:

$$||\hat{\mathbf{r_{pj}}}|| = \frac{\sqrt{q}||\hat{\mathbf{p_{cj}}}|| + ||\hat{\mathbf{p_{pj}}}||}{1+q} \quad (39)$$

and thus

$$||\hat{\mathbf{r_{cj}}}|| = \frac{q||\hat{\mathbf{p_{cj}}}|| + \sqrt{q}||\hat{\mathbf{p_{pj}}}||}{1+q} \quad (40)$$

Therefore, the solution to rating re-estimation of community j can be expressed as:

$$q = \frac{||\mathbf{p_c}||^2}{||\mathbf{p_p}||^2} \quad (41)$$

$$\hat{\mathbf{r_{cj}}} = \frac{q||\hat{\mathbf{p_{cj}}}|| + \sqrt{q}||\hat{\mathbf{p_{pj}}}||}{1+q} \cdot \mathbf{v}(\mathbf{A} \times \mathbf{A^T}) \quad (42)$$

$$\hat{\mathbf{r_{pj}}} = \frac{\sqrt{q}||\hat{\mathbf{p_{cj}}}|| + ||\hat{\mathbf{p_{pj}}}||}{1+q} \cdot \mathbf{v}(\mathbf{A^T} \times \mathbf{A}) \quad (43)$$

We also note, that using the approach derived above, the calculation for each community can be fully paralleled, which enables us to make good use of the multiple cores available in our workstation and significantly reduce the computation time for large scale networks.

4 Experiment and Results

Based on the Asymmetrical Rating Exchange Model (AREM) proposed in Section 2 and the solution derived in Section 3, a system called Unified RAS is implemented in R(R Development Core Team 2011), with the support of package *igraph*(Csardi & Nepusz 2006) to handle the graph data structure and a proportion of visualisation work.

In our application, we focused on a dataset extracted for 2009-2010 financial year. The nodes in the network are the consumers who had at least one flag assessed by Consumer RAS(Ng et al. 2010, Mendis et al. 2011), and the prescribers who prescribed scripts for these consumers during the time period analysed, which end up with 37,148 nodes composed of 16,829 consumers and 20,319 prescribers, and 102,719 weighted edges.

As mentioned in Section 3, the first step of our proposed approach is to divide the entire large network into communities, where we adopt the fast algorithm for large scale networks proposed in (Clauset et al. 2004), which has been made available in the R package *igraph*(Csardi & Nepusz 2006). It only took approximately 27 seconds to divide the whole network into 4,363 communities, and 3 seconds for the calculation expressed from Eq.(41) to Eq.(43) with paralleled implementation for these 4,363 subgraphs.

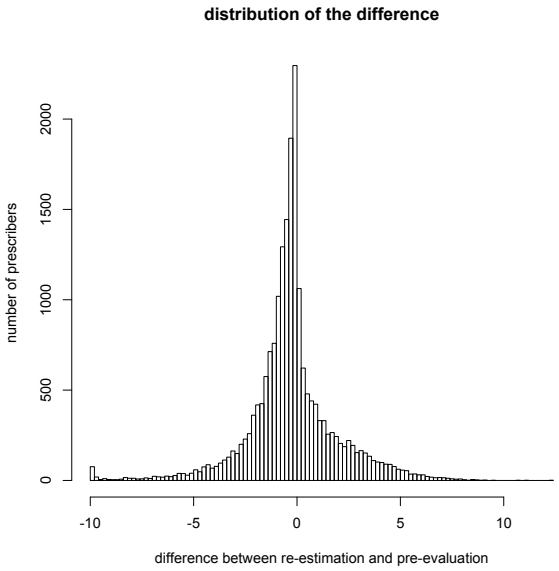


Figure 1: Distribution of difference between re-estimation and pre-evaluation for prescribers

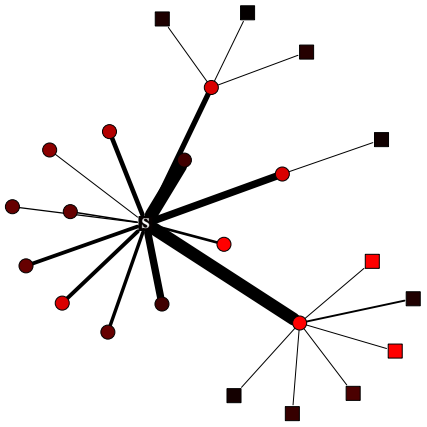
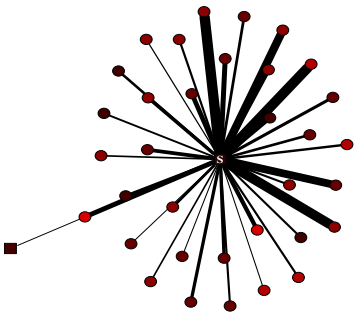
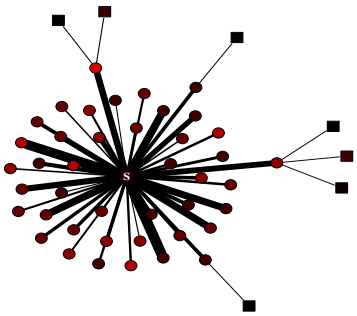


Figure 2: An example of the targeted prescriber and its community structure. ■: prescribers, ●: consumers, **S** over a ■: prescriber of interest

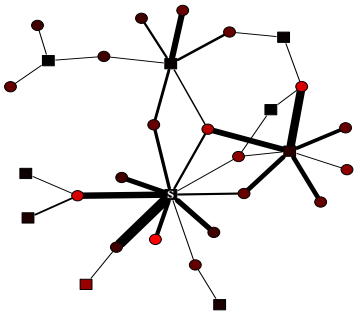
After working out the re-estimation of the ratings $\hat{\mathbf{r}}$, we can compare the re-estimated results against the pre-evaluated values $\hat{\mathbf{p}}$ and identify those entities that were significantly under-rated by independent systems without considering network interactions. Even though we are trying to find under-rated



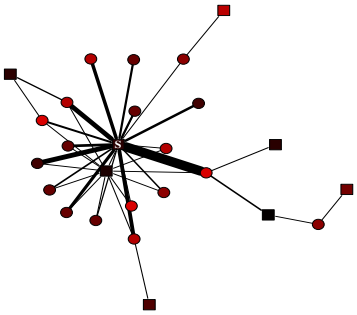
(a) Community 1



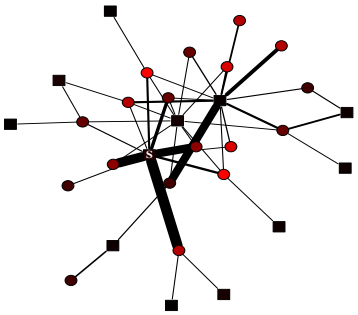
(b) Community 2



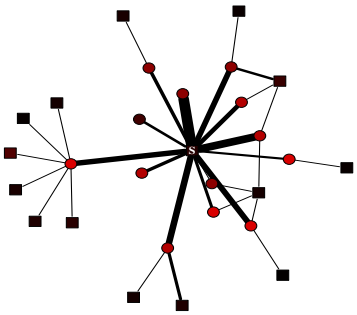
(c) Community 3



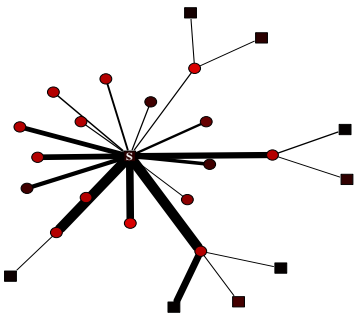
(d) Community 4



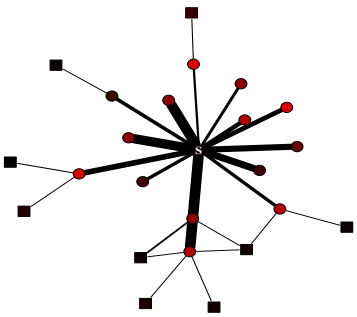
(e) Community 5



(f) Community 6



(g) Community 7



(h) Community 8

Figure 3: Eight detected communities

prescribers here, we should still expect that there would not be a big difference for most of the prescribers, under the assumption that the independent systems are relatively accurate. Taking prescribers as examples, the distribution of the differences between the re-estimation and pre-evaluation are shown in Fig.(1).

We can see that the difference is close to zero for most of the entities. This means that the ratings for most of the entities assessed by independent RASes are reasonably close to the results based on network structures, which also confirms the effectiveness of both independent RASes and the Unified RAS. Meanwhile, we can also see that there are a few entities sitting on the right hand side of the figure, which means they are significantly under-rated. By applying a threshold. e.g. a cut-off value, these under-rated entities could be easily targeted.

As an example, Fig.2 shows one of the prescriber picked up by the system and its community. Square and circle nodes denote prescribers and consumers in the network, respectively. The redder a node colour is, the higher pre-evaluated risk level for the entity. The width of the edges represents the frequency with which a consumer obtains original prescriptions from a prescriber, and the wider, the more often. The vertex marked with white 'S' is the prescriber considered as significantly under-rated based on its community structure. From Fig.2, we can also visually explain why this prescriber is picked up by the Unified RAS. Many high risk (very red) nodes around are visiting this prescriber quite often, especially, the very red circle at the right bottom corner visited many prescribers including two high risk ones, and also obtained very large number of prescriptions (more than 200 original scripts for sensitive drugs, listed in previous papers regarding to Consumers RAS(Ng et al. 2010, Mendis et al. 2011)) within the financial year from the prescriber selected. Considering these factors from a network perspective, even though this prescriber was originally rated with LOF (local outlier factor) value equal to 1.02, which was a very normal value from Prescriber RAS, we still need to highlight this prescriber as a high risk figure.

Another eight of the detected communities where prescribers are considered under-rated based on network connections are plotted in Fig.3(a) - 3(h), It can be clearly seen that these prescribers who originally had a low risk ratings from Prescriber RAS have a very strong connections with high risk entities and behave like prescription suppliers to suspicious consumers in their communities. Based on the proposed Asymmetrical Rating Exchange Model and derived approach described in Section 3, we are able to automatically target them from a large scale network with more than 30,000 nodes and 100,000 edges. The overall algorithm implemented in R takes less than 5 minutes to work out the identified entities, which proved to be effective and efficient for our applications.

5 Conclusion

In this paper, we proposed an Asymmetrical Rating Exchange Model (AREM) to describe the risk behaviour influences between prescribers and consumers, which is in theory a mathematical generalisation to the eigenvector centrality problem for two-mode networks. Based on the model, we also derived an algorithm to re-estimate the risk level of entities based on both network structures and pre-evaluated ratings from Prescriber RAS and Consumer RAS. Comparing the re-estimated value with pre-evaluated rating, under-rated entities can be successfully detected by our Unified RAS. Experimental result further proved that mathematical model and derived algorithm are effective and efficient to identify suspicious targets that have strong interactions in high risk communities.

In the future, the analysis of the entities that have been identified using the technique described in this paper will be undertaken by subject matter experts to evaluate the accuracy, i.e. hitting rate from medical and pharmaceutical practice compliance perspective. There are a number of other entity types that interact with Medicare Australia and be suitable for analysis using the technique outlined in this paper. The network presented in this paper could be extended by including the connections to pharmacies. Similarly the relationships between consumers, providers and referred service providers can be modelled using this technique. This analysis could be modified to include the relationships between entities of the same type. e.g. providers who render services in a practice with other providers or ownership partnerships between pharmacies. For providers who render services in several practices, the services can be separated out to gain a practice centric view of the interactions.

Acknowledgements

The authors would like to acknowledge the input from Dr. David Jeacocke for his insightful comments during this research, and also special thanks to Leonie Greenwood, Rory King and Paul Cowan for their support from management.

References

Bonacich, P. (1972), 'Factoring and weighting approaches to status scores and clique identification', *Journal of Mathematical Sociology* **2**, 113–120.

Bonacich, P. (1991), 'Simultaneous group and individual centralities', *Social Networks* **13**(2), 155–168.

Borgatti, S. P. & Everett, M. G. (1997), 'Network analysis of 2-mode data', *Social Networks* **19**, 243–269.

Breunig, M., Kriegel, H., Ng, R. & Sander, J. (2000), 'LOF: identifying density-based local outliers', *Sigmod Record* **29**(2), 93–104.

Clauset, A., Newman, M. E. & Moore, C. (2004), 'Finding community structure in very large networks', *Physical Review E*.

Cortes, C., Pregibon, D. & Volinsky, C. (2001), 'Communities of interest', Lecture Notes in Computer Science.

Csardi, G. & Nepusz, T. (2006), 'The igraph software package for complex network research', *InterJournal* p. 1695.

Girvan, M. & Newman, M. E. (2002), 'Community structure in social and biological networks', *Proceedings of the National Academy of Sciences* **99**(12), 7821–7826.

Hu, Y., Murray, W. & Shan, Y. (2011), *Rlof: R parallel implementation of Local Outlier Factor(LOF)*, Strategic Data Mining Team, Department of Human Services, Australia. R package version 1.0.0.

Medicare Australia (2010), 'Medicare Australia Annual Report 2009-10', http://www.medicareaustralia.gov.au/about/governance/reports/09-10/index.jsp.

Medicare Australia (2011), 'Medicare Australia's National Compliance Program 2010-2011', http://www.medicareaustralia.gov.au/provider/business/audits/files/3013-national-compliance-program2010-2011.pdf.

Mendis, S., Murray, W., Sutinen, A., Tang, M. & Hu, Y. (2011), Enhancing the identification of anomalous events in medicare consumer data through classifier combination, *in* '6th International Workshop on Chance Discovery', Springer Press, Barcelona, pp. 39–44.

Neville, J., Simsek, O., Jensen, D., Komoroske, J., Palmer, K. & Goldberg, H. (2005), Using relational knowledge discovery to prevent securities fraud, *in* 'ACM SIGKDD Conference on Knowledge Discovery and Data Minig', pp. 449–458.

Ng, K., Shan, Y., Murray, D., Sutinen, A., Schwarz, B., Jeacocke, D. & Farrugia, J. (2010), Detecting Non-compliant Consumers in Spatio-Temporal Health Data: A Case Study from Medicare Australia, *in* '2010 IEEE International Conference on Data Mining Workshops', pp. 613–622.

Pearson, R., Murray, D. & Mettenmeyer, T. (2006), 'Finding anomalies in medicare', *Electronic Journal of Health Informatics*.

R Development Core Team (2011), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Shan, Y., Jeacocke, D., Murray, D. & Sutinen, A. (2008), Mining medical specialist billing patterns for health service management, *in* 'Proceedings of the conferences in Research and Practice in Information Technology'.

Shan, Y., Murray, D. & Sutinen, A. (2009), Discovering inappropriate billings with local density based outlier method, *in* 'Proceedings of the conferences in Research and Practice in Information Technology', pp. 105–110.

Wasserman, S. & Faust, K. (1994), *Social Network Analysis: Methods and Applications*, Cambridge University Press.

Yamanishi, K., ichi Takeuchi, J., Williams, G. & Milne, P. (2004), 'On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms', *Data Mining and Knowledge Discovery* **8**(3), 275–300.

Model Selection Strategy for Customer Attrition Risk Prediction in Retail Banking

Fan Li [1], Juan Lei [2], Ying Tian [3], Sakuna Punyapatthanakul [4] and Yanbo J. Wang [1, 5]

[1] Information Management Center, China Minsheng Banking Corp., Ltd.
[2] Department of Retail Banking, China Minsheng Banking Corp., Ltd.
[3] Beijing Dongdan Sub-branch, China Minsheng Banking Corp., Ltd.
No. 2, Fuxingmennei Avenue, Xicheng District, Beijing 100031, China

[4] Consumer Segment Management Department, Retail Business Division, KASIKORNBANK
1 Soi Rat Burana 27/1, Rat Burana Road, Bangkok 10140, Thailand

[5] Institute of Finance and Banking, Chinese Academy of Social Sciences
No. 5, Jianguomennei Dajie, Dongcheng District, Beijing 100732, China

{lifan,leijuan2,tianying,wangyanbo}@cmbc.com.cn
sakuna.p@kasikornbank.com

Abstract

Nowadays customer attrition is increasingly serious in commercial banks, particularly, *high-valued* customers in retail banking. Hence, it is encouraged to develop a prediction mechanism and identify such customers who might be at risk of attrition. This prediction mechanism can be considered to be a classifier. In particular, the problem of predicting risk of customer attrition can be prototyped as a *binary* classification task in data mining. In previous studies, a number of techniques have been introduced in (*binary*) classification study, i.e. artificial-based model, Bayesian-based model, case-based model, tree-based model, regression-based model, rule-based model, etc. With regards to a particular application — predicting customer attrition risk for retail banking, this paper presents four principles in (classification) model selection. To support this model selection study, a set of experiments were run, based on a collection of *real* customer data in retail banking. These results and consequent recommendations are given in this paper.

Keywords: Classification Prediction, Commercial Banks, Customer Attrition Risk, Model Selection, Retail Banking.

1 Introduction

With increased competition within the domestic banking industry, customer churn/attrition is increasingly serious in commercial banks, particularly, *high-valued* customers in retail banking. Nowadays, more and more commercial banks start to pay attention to CRM (Customer Relationship Management), especially the investigation of retaining existing customers. In the work (Luck, 2009), the author clearly states that "*retaining customers is more profitable than building new relationships*". Kandampully and Duddy (1999) even attempt to clarify that attracting a new customer is about five times more costly than retaining an existing customer. Hence, "*the retention of existing customers has become a priority for businesses to survive and prosper*" (Luck, 2009). Consequently, accurately identifying those customers who might be at risk of attrition has become an essential problem. For commercial banks in general, it is suggested to produce a prediction mechanism that can be used to classify whether an existing customer will churn in the near future (in the next business/observation period).

The rest of this paper is organised as follows. The following section indicates the link between our problem of study and the data mining classification task and summarises some related works. Section 3 presents the strategy of model selection, which mainly consists of four principles. Experimental results, based on the collected VIP customer data from a *real* retail banking environment are shown in Section 4. During the experiments, a number of classification models were compared and the most suitable one was selected for each of the principles. Finally, conclusions and direction for future work are given at the end of this paper.

2 Related Work

2.1 Classification Models

The customer attrition risk identification problem can be prototyped as a *binary* classification task in data mining — *binary* classification, also referred to as 2-*class* classification, "*learns from both positive and negative data samples, and assigns either a predefined category* (*class-label*) *or the complement of this category to each 'unseen' instance*" (Wang *et al.*, 2011). In past decades, many models/techniques have been proposed in the study of (2-*class*) classification that include: Artificial-Based Classification (ABC), Bayesian-Based Classification (BBC), Case-Based Classification (CBC), Tree-Based Classification (TBC), Regression-Based Classification (REBC), Rule-Based Classification (RUBC), etc.

- **ABC Model** aims to solve the classification problem by using *AI* (*Artificial Intelligence*) techniques. One typical approach is the Artificial Neural Network (ANN). Knowledge on ANN classification can be found in the study (Berson and Smith, 1997, 375-406).

- **BBC Model** aims to solve the classification problem using the *Bayesian* theory. One well known approach is Naïve Bayes (NB). In the work (Wang, 2007, 24-25), the author depicts the general idea of NB classification.
- **CBC Model** aims to solve the classification problem by *lazily* utilising the training examples/ cases. One typical approach is using *k*–Nearest Neighbours (*k*–NN). Knowledge relating to *k*–NN classification has been contributed by Cunningham and Delany (2007).
- **TBC Model**'s approach to solving the classification problem is based on a *greedy* algorithm. The result of classifier construction using TBC is to build a *Decision Tree* (*DT*). C4.5/C5.0 (Quinlan, 1993) is the best known *DT* classification approach.
- **REBC Model** aims to solve the classification problem using the statistical *regression* study. The approach is namely Logistic Regression (LR). Study as related to LR classification can be found in the work (Witten and Frank, 2005, 121-125).
- **RUBC Model** aims to solve the classification problem by generating a set of "*IF–THEN*" patterns/rules, where each rule is expressed in the form of "*attribute*(*s*) ⇒ *category*". The generated set of (human readable and understandable) rules represents the (constructed) classifier and presents to the end users why and how the classification predictions have been made. One typical mechanism in this school is namely RIPPER (Cohen, 1995).

2.2 Previous Work

In the previous studies of customer attrition risk identification, Khan, Jamwal and Sepehri (2010) indicate that TBC DT, REBC LR, and ABC ANN can be almost equally well applied in the Internet Service Provider (ISP) industry. By running experiments on real data collected from the ISP industry, Khan, Jamwal and Sepehri (2010) point out that ABC ANN slightly outperforms the other two models/approaches on "overall accuracy"; and REBC LR slightly outperforms the other two models/ approaches on "churner hit rate" (also referred to as "recall of attrition"). In the work (Li, 2009), the author suggests to use TBC DT in banking customer attrition risk prediction. However Li's work does not show the experimental results in model performance evaluation.

3 Model Selection Strategy

With regards to a particular application — predicting customer attrition risk for retail banking, our study proposes four principles that can be applied strategically in classification model selection.

3.1 Principle of Good Performance

Broadly speaking, we say a classification model demonstrates good performance if the model shows good classification accuracy. Simply speaking, classification accuracy is the fraction of correctly predicted "instance–category" mappings, which is calculated as the number of correctly classified instances divided by the total number of instances to be classified.

Other measures that have been used in classification performance evaluation, especially in *binary* classification (Zheng and Srihari, 2003), include: recall, precision, the F1 measure, micro-averaging, macro-averaging, etc. With respect to the application of Text Categorization (TC), Sebastiani (2005) explains the reason for applying these evaluation measures in *binary* classification rather than using accuracy alone: "*in binary TC application the two categories c and c are usually unbalanced, i.e. one contains for more members than the other*". Therefore, "*building a classifier that has high accuracy is trivial*", i.e. a classifier could directly assign the majority class-label found in the *training* dataset for all *test* instances and reach a high classification accuracy without any (serious/ intelligent) computation to be involved.

In our study, the data collected from a *real* retail banking environment is not (necessarily) class-*balanced* too — usually the number of customers who are going to churn is less than the number of customers who will stay. However, this is not the basis for recommending utilising recall and precision in classification model evaluation/ selection. In a banking context, accurately/correctly identifying a customer who is at risk of attrition is more valuable than finding a customer who will truly stay. Hence, the involvement of "recall of attrition" is strongly recommended in model evaluation and in our study, this measure in some cases stands for the execute-ability of customer retention. The "recall of attrition" is calculated as the number of truly churned customers who have already been correctly predicted divided by the total number of truly churned customers.

In the extreme situation of 100% "recall of attrition", all customers are going to churn. In this case, all truly churned customers can definitely be identified in the prediction phase, but obviously this is not a good prediction model. Consequently, the "precision of attrition" measure is of central concern in our study. The "precision of attrition" is calculated as the number of truly churned customers who have already been correctly predicted divided by the total number of customers who have been predicted to churn. Actually, this measure represents the cost-level of customer retention — i.e. a high "precision of attrition" means that most of the churn-predicted customers are truly churned customers. This results in the most efficient management of customer retention funds.

In our study, we use the "overall accuracy" and the "recall of attrition" plus "precision of attrition" to measure whether a classification model satisfies the principle of good performance.

3.2 Principle of Efficiency

Simply speaking, efficiency is a measure of time and can be used to demonstrate how fast an information system can be processed. The most straightforward way to evaluate an information system's efficiency is to count the system's running time in seconds. In our study, we count both the "classifier building time" and the "overall running time" to determine whether a classification model satisfies the principle of efficiency.

3.3 Principle of Instance Ranking

In a *real* banking environment, especially in retail banking, the number of existing customers (shown as the collected customer data-instances) can be as large as some ten millions. We can assume that at least 10% of these customers are predicted/going to churn, which means there are more than one million customers who require us to implement customer-care (for customer retention). Obviously, this number is too large to be handled by a commercial bank at one time. Hence, it is necessary to distinguish customers with a high probability to churn from the ordinary ones. Further, it is encouraged to rank all customer-instances in descending order, based on their churn probabilities, so that the bank can easily select a suitable size of predicted (the most probable) churn-customers to implement customer-care. In our study, we use both the "smoothness of probability distribution" and the "slope of probability distribution" to measure whether a classification model satisfies the principle of instance ranking.

3.4 Principle of Rule Generation

In a general context, classification models can be separated into two schools — (*i*) classification without rule generation *vs.* (*ii*) classification with rule generation (and presentation). In the early stage of classification investigation (1960's ~ 1980's), most of the models/ approaches were proposed by the first school. In the past two decades (1990's ~ present), the tree-based and rule-based classification techniques were introduced. These aim to generate human-readable and human-understandable patterns/rules while classifying "unseen" data-instances and present to the end users why and how the classification predictions have been made. In our study, the rules generated to demonstrate why and how a set of customers are at risk of attrition are clearly stated and are essential to the design of an *effective* customer-care program for customer retention.

4 Experimental Results

In this section, we present four groups of evaluations for our proposed model selection strategy — one for each principle, using a set of collected VIP customer data from a *real* retail banking environment. All evaluations were obtained using the WEKA software [1] (Witten and Frank, 2000; Witten and Frank, 2005; Witten, Frank and Hall, 2011). The experiments were run on a 3.00 GHz Pentium(R) Dual-Core CPU with 1.96 GB of RAM running under the x86 Windows Operating System.

4.1 Description of Data

From a *real* commercial bank's EDW (Enterprise Data Warehouse), we collect a set of retail banking VIP customer (attrition) data across nine *continuous* months. The class-labels (also noted as the category-attribute values) of this dataset are "*attrition*" *vs.* "*non-attrition*", which represents the churn status of each (VIP) customer in the seventh month to the ninth month. The features (also noted as the data-attributes) in this dataset are grouped into customer's "*geo-demographical*", "*financial*", "*product*", "*transaction*" and "*loan*" information, which together, clearly depicts each customer in the first month to the sixth month.

After the data cleansing process — data-instances with missing and/or noisy values are eliminated. We then randomly select 2000 "*attrition*" plus 2000 "*non-attrition*" data-instances to create a class-*balanced* dataset. In feature selection, we choose only 12 simple data-attributes, i.e. customer's age, number of products holding, time to stay with the bank, etc., based on suggestions given by the bank's financial managers.

4.2 Description of Models and Approaches

In the WEKA software version 3.6.4, the implementation of ABC ANN approach is namely MultilayerPerceptron; the implementation of BBC NB is namely NaiveBayes; the CBC *k*–NN approach is namely IBk, and in our study *k* was set to be 5; the TBC DT (C4.5/C5.0) is namely J48; the REBC LR is namely Logistic; and the RUBC RIPPER approach is namely JRip. In our experiments, we ran these implemented methods using our prepared data on the WEKA platform.

4.3 Description of Results

First of all, the six classification models/approaches (as introduced above) are evaluated by "overall accuracy", "recall of attrition" and "precision of attrition" (see Table 1). The experimental results were obtained using the Ten-fold Cross Validation (TCV) setting. From the evaluation, the BBC NB model/approach (as highlighted in Table 1) only shows 61.8% "overall accuracy". Although it recognises 94.3% of the customers who are truly going to churn, the cost of obtaining this "recall of attrition" is very high (the "precision of attrition" is only 57.2%). In this case, we suggest abandoning BBC NB. Results from other models/approaches are valued at the same level.

Models Approaches	Recall of Attrition	Precision of Attrition	Overall Accuracy
ABC ANN	78.9%	87.8%	83.950%
BBC NB	94.3%	57.2%	61.800%
CBC *k*–NN	79.2%	83.1%	81.525%
TBC DT	80.3%	87.6%	84.425%
REBC LR	79.3%	79.3%	79.325%
RUBC RIPPER	79.9%	88.3%	84.675%

Table 1: Experimental results for the principle of good performance

Secondly, the "classifier building time" and the "overall running time" for the six classification models/approaches were counted in seconds and the results are shown and compared as follows (see Table 2). In general, the "classifier building time" is less than 0.7 seconds and the "overall running time" is less than 8 seconds, except ABC ANN (as highlighted in Table 2). It can be evaluated that the run-time efficiency of ABC ANN is at least 15 times higher than the efficiency of other

[1] The well known WEKA software, a Data Mining and Machine Learning Software in Java, may be obtained from http://www.cs.waikato.ac.nz/~ml/weka/.

models/approaches (calculated as 12.55 √ 0.7 ≈ 17.9 and 122 √ 8 ≈ 15.6). Note that in our experiments, the prepared dataset involves only 13 data-attributes (including the category-attribute) and contains only 4000 data-instances. If it were to handle a very large data collection in our study, the run-time efficiency of ABC ANN may not demonstrate equal endurance. Hence, it is suggested to abandon the ABC ANN model/approach.

Models Approaches	Classifier Building Time (in Sec.)	Overall Running Time (in Sec.)
ABC ANN	12.55	122
BBC NB	0.02	1
CBC *k*–NN	0.0001	7
TBC DT	0.2	2
REBC LR	0.14	2
RUBC RIPPER	0.61	7

Table 2: Experimental results for the principle of efficiency

The results of the third evaluation (for the principle of instance ranking) are shown as follows (see Table 3, 4 and Figures 1 ~ 6). From the experiments, the six produced classifiers (with regards to the issue of *anti-overfitting*), based on the six introduced classification models/ approaches were employed to assign a score of churn/ attrition probability (between 0 and 1) to each of the 4000 originally given customers/data-instances. Note that this operation can be done easily in WEKA by simply selecting the "Output predictions" box after clicking the "More options… (Classifier evaluation options)" button under the "Weka Explorer — Classify" subtitle.

We draw the customer attrition probability distributions generated by each of the six approaches/classifiers into graphs (see Figures 1~6). From these figures, we see that the Figures 5, 2 and 1 show better graph smoothness ("smoothness of probability distribution") than the Figures 4, 6 and 3. In fact, the probability (score) values of ABC ANN, BBC NB and REBC LR are *continuous*, whereas the probability values of CBC *k*–NN, TBC DT and RUBC RIPPER are *discrete*. It can be argued that the usability of *discrete* valued probability distribution is weak, e.g. there are only 9 *discrete* values through the customer attrition probability distribution of CBC *k*–NN and RUBC RIPPER (see Table 4), so that it is difficult to catch the top 400 and later on the next 500 customers who are the most probable to churn, like the result provided by REBC LR as follows (see Table 3).

Models Approaches	Number of Instances (based on the attribution probability score)					Value Desc. of Probability Distribution
	≥ 0.9	≥ 0.8	≥ 0.7	≥ 0.6	≥ 0.5	
ABC ANN	1475	1552	1739	1793	1883	Continuous
BBC NB	2076	3002	3189	3253	3301	Continuous
CBC *k*–NN	1018	1543	1543	1895	1896	Discrete
TBC DT	1472	1641	1777	1783	1783	Discrete
REBC LR	407	578	915	1486	2007	Continuous
RUBC RIPPER	1213	1576	1600	1807	1807	Discrete

Table 3: Experimental results for the principle of instance ranking

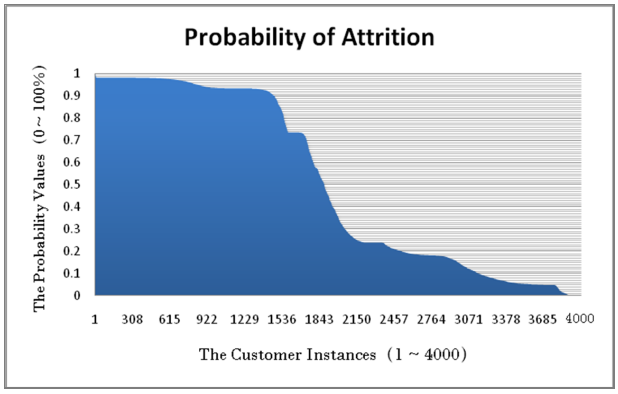


Figure 1: The probability distribution graph of customer attrition by ABC ANN

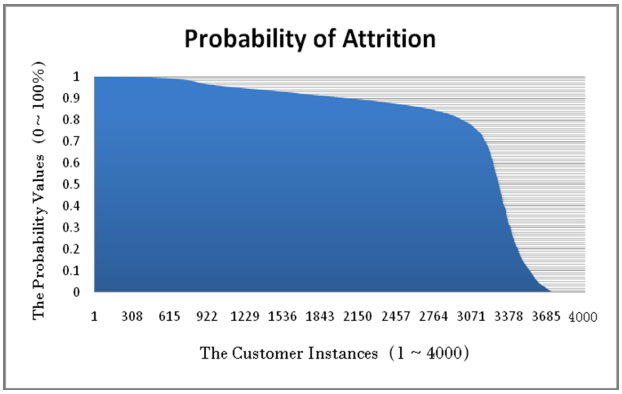


Figure 2: The probability distribution graph of customer attrition by BBC NB

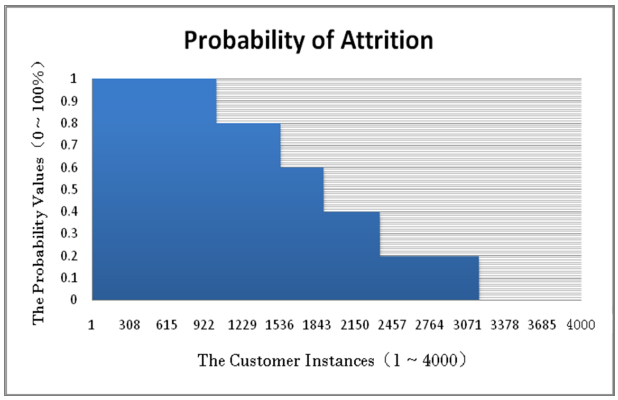


Figure 3: The probability distribution graph of customer attrition by CBC *k*–NN

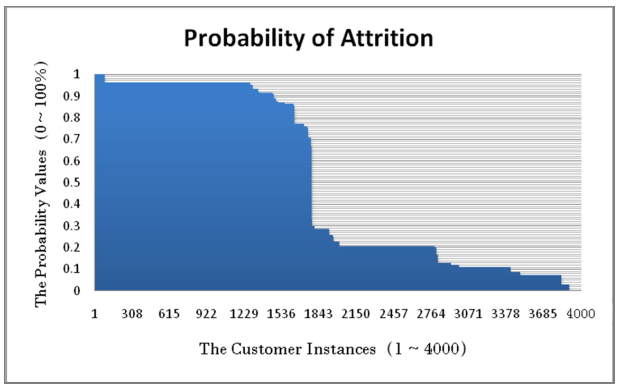


Figure 4: The probability distribution graph of customer attrition by TBC DT

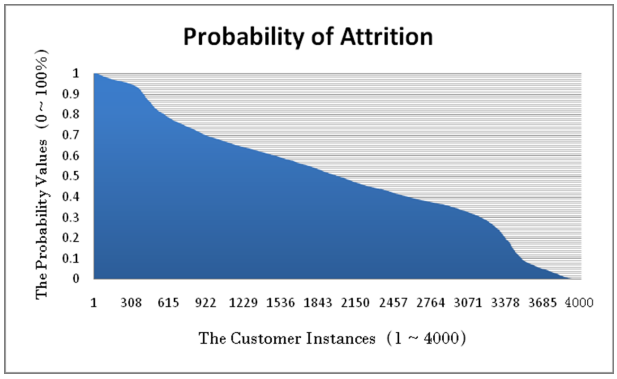


Figure 5: The probability distribution graph of customer attrition by REBC LR

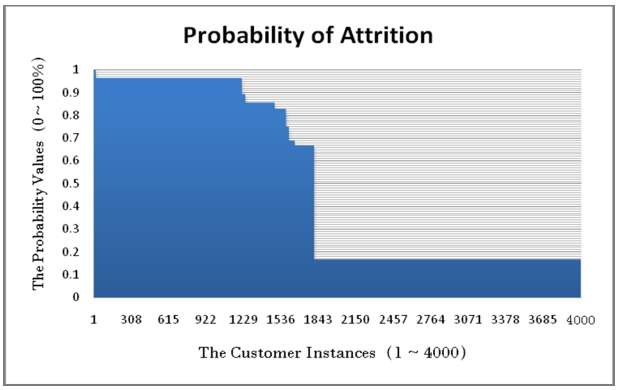


Figure 6: The probability distribution graph of customer attrition by RUBC RIPPER

The # of Discrete Values	CBC *k*–NN		RUBC RIPPER	
	The Discrete Values (in ↓ order)	Accumulated Number of Customers	The Discrete Values (in ↓ order)	Accumulated Number of Customers
1	1	1018	1	13
2	0.8	1543	0.963	1213
3	0.6	1895	0.893	1241
4	0.5	1896	0.855	1483
5	0.4	2355	0.828	1576
6	0.3	2356	0.75	1600
7	0.2	3166	0.688	1648
8	0.1	3167	0.667	1807
9	0	4000	0.167	4000

Table 4: Detailed description of the customer attrition probability distribution for CBC *k*–NN & RUBC RIPPER

From the "slope of probability distribution", we see that the Figures 1, 2, 4 and 6 show a flatter curve slope than the Figures 3 and 5 for the first 1500 customers. The flatter curve slope represents the difficulty of identifying significant churn-predicted customers from ordinary customers. From Table 3, we see that ABC ANN, BBC NB, TBC DT and RUBC RIPPER catch respectively 1475, 2076, 1472 and 1213 customers that are predicted with higher than 90% probability to churn. Note that we only have 2000 truly churned customers in total.

By adopting both the "smoothness of probability distribution" and the "slope of probability distribution", the only model/approach that satisfies the principle of instance ranking is REBC LR. It is advisable to abandon other models/approaches (as highlighted in Table 3).

Finally, we look into the principle of rule generation. A set of experiments were run on our prepared dataset. The experimental results are shown as follows (see Table 5), where only TBC DT and RUBC RIPPER are able to generate classification rules and present to the end users why and how the customer attrition predictions have been made. Based on the TBC DT approach, a tree classifier was constructed that contains 82 leaf nodes (classification rules). Figure 7 partially shows the tree classifier. Moreover, the RUBC RIPPER approach generates 9 classification rules; Figure 8 partially lists the rule classifier. The quality of these generated rules, in terms of the extent to which they correlate with a *priori* knowledge, can be confirmed by experienced financial managers.

Models Approaches	Rule Generation	Number of Rules
ABC ANN	×	—
BBC NB	×	—
CBC *k*–NN	×	—
TBC DT	√	82
REBC LR	×	—
RUBC RIPPER	√	9

Table 5: Experimental results for the principle of rule generation

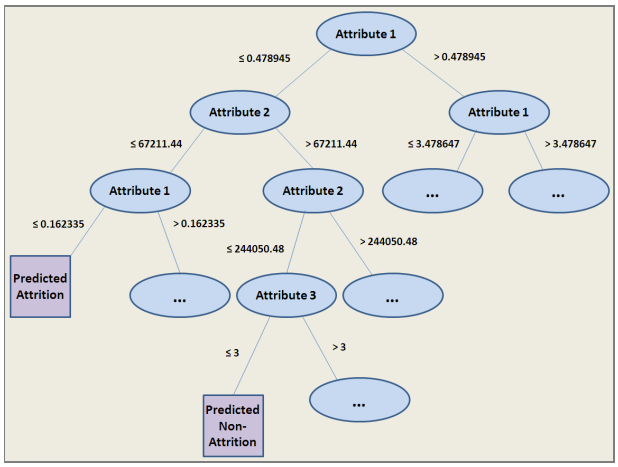


Figure 7: Rule generation by TBC DT
(The J48 decision tree classifier is shown partially)

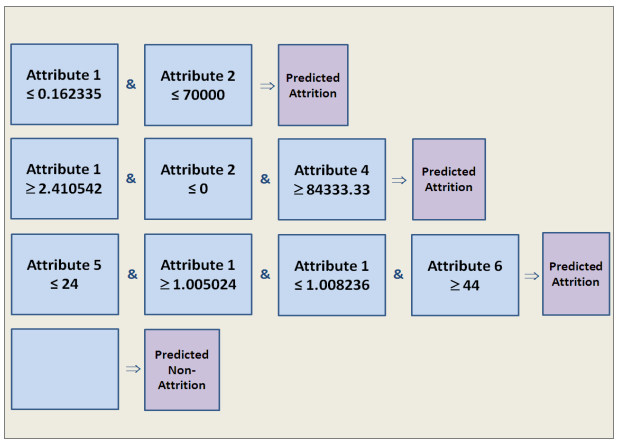


Figure 8: Rule generation by RUBC RIPPER
(The JRip rule classifier is shown partially)

4.4 Summary

Four groups of experiments were run — one for each of the proposed model selection principles. From the experimental results, it can be summarised that the ABC ANN, BBC NB and CBC *k*–NN models/approaches should be abandoned due to the following reasons:

- **ABC ANN** does not demonstrate acceptable performance in terms of "overall accuracy" and "precision of attrition".
- **BBC NB** does not show an acceptable level of efficiency in terms of "classifier building time" and "overall running time".
- **CBC *k*–NN** satisfies neither the principle of instance ranking nor the principle of rule generation.

With ability to satisfy both the principles of performance and efficiency, we suggest to adopt REBC LR since this classification model/approach satisfies the principle of instance ranking well. Furthermore, we recommend also adopting the TBC DT and RUBC RIPPER models/ approaches since both demonstrate equal capacity to satisfy the principle of rule generation.

In a banking context, it is preferable to provide explanation of why and how a customer is predicted to be at risk to churn when adopting the REBC LR classifier. The Voice Of Customer (VOC) analysis (by questionnaire) that investigates the reason of customer attrition, is definitely suggested. The result of VOC analysis, as a substitute of the generated rule list/set, can be used to design the customer-care program for customer retention.

On the other hand, if the TBC DT or RUBC RIPPER classifier were adopted, it would be encouraged to develop a better churn/attrition probability scoring mechanism, which identifies the top-most probable *f*-% customers to churn (in the near future). Here *f* can be any number between 0 and 100. Again, the VOC analysis should be utilised, as it ranks all or a fraction of the churn-predicted customers in descending order, based on their churn probability.

5 Conclusions

Today, customer attrition, especially for *high-valued* customers in retail banking, has become more and more serious in commercial banks. Hence, customer retention, and consequently predictions of customer attrition have become a priority issue for the survival and prosperity of commercial banks. In this paper, we investigated the customer attrition prediction problem based on a collection of customer (attrition) data from a *real* retail banking environment. By prototyping our study into a data mining *binary* classification problem, we listed a number of classification models/approaches that can be selected. We further proposed four model selection principles, and a set of experiments were run based on our prepared dataset. The experimental results show that although none of the models are perfect, the REBC LR, TBC DT and RUBC RIPPER models/approaches are recommended for adoption for customer attrition prediction in retail banking. Further research is suggested to produce an improved classification model/approach that satisfies all our proposed model selection principles simultaneously.

6 Acknowledgements

The authors would like to thank Dr. Jiongyu Li from the China Minsheng Banking Corp., Ltd., Pipit Aneaknithi and Lei Xiao from KASIKORNBANK (Thailand), Jiangtao Lai from IWT Solutions Ltd. (KXEN Exclusive Distributor in China and Hong Kong), Haixia Pan from the College of Software at Beihang University, and Karen Zhang from Work Place Safety and Insurance Board (Ontario, Canada) for their support with respect to the work described here.

7 References

Berson, A. and Smith, S.J. (1997): *Data warehousing, data mining, and OLAP*. New York, NY, McGraw-Hill Companies, Inc.

Cohen, W.W. (1995): Fast effective rule induction. *Proc. of the 12th International Conference on Machine Learning*, Tahoe City, CA, 115-123, Morgan Kaufmann Publishers.

Cunningham, P. and Delany, S.J. (2007): *k*–nearest neighbour classifiers. Technical report (UCD-CSI- 2007-4). University Colledge Dublin, Ireland.

Khan, A.A., Jamwal, S. and Sepehri, M.M. (2010): Applying data mining to customer churn prediction in an internet service provider. *International Journal of Computer Applications* **9**(7): 8-14.

Li, X. (2009): ID3 applying to loss of bank clients. *Computer Technology and Development* **19**(3): 158-167.

Luck, D. (2009): The importance of data within contemporary CRM. In the book *Data Mining Applications for Empowering Knowledge Societies*. 96-109. Rahman, H. (ed). Hershey, PA, IGI Global.

Kandampully, J. and Duddy, R. (1999): Relationship marketing: a concept beyond primary relationship. *Marketing Intelligence and Planning* **17**(7): 315-323.

Quinlan, J.R. (1993): *C4.5: programs for machine learning*. San Francisco, CA, Morgan Kaufmann Publishers.

Sebastiani, F. (2005): Text categorization. In the book *Text Mining and Its Applications to Intelligence, CRM and Knowledge Management (Advances in Management Information)*. 109-129. Zanasi, A. (ed). Southampton, UK, WIT Press.

Wang, W., Wang, Y.J., Xin, Q., Bañares-Alcántara, R., Coenen, F. and Cui, Z. (2011): A comparative study of associative classifiers in mesenchymal stem cell differentiation analysis. In the book *Knowledge Discovery Practices and Emerging Applications of Data Mining: Trends and New Domains*. 223-243. Kumar, A.V.S. (ed). Hershey, PA, IGI Global.

Wang, Y.J. (2007): Language-independent pre-processing of large documentbases for text classification. Ph.D. thesis. University of Liverpool, UK.

Witten, I.H. and Frank, E. (2000): *Data mining: practical machine learning tools and techniques with java implementations*. San Francisco, CA, Morgan Kaufmann Publishers.

Witten, I.H. and Frank, E. (2005): *Data mining: practical machine learning tools and techniques (second edition)*. San Francisco, CA, Morgan Kaufmann Publishers.

Witten, I.H., Frank, E. and Hall, M.A. (2011): *Data mining: practical machine learning tools and techniques (third edition)*. Burlington, MA, Morgan Kaufmann Publishers.

Zheng, Z. and Srihari, R. (2003): Optimally combining positive and negative features for text categorization. *Proc. of the 2003 ICML Workshop on Learning from Imbalanced Data Sets II*, Washington DC.

An Efficient Two-Party Protocol for Approximate Matching in Private Record Linkage

Dinusha Vatsalan[1], Peter Christen[1], and Vassilios S. Verykios[2]

[1] Research School of Computer Science, College of Engineering and Computer Science
The Australian National University, Canberra ACT 0200, Australia
Email: dinusha.vatsalan@anu.edu.au, peter.christen@anu.edu.au

[2] School of Science and Technology
Hellenic Open University, Patras, Greece
Email: verykios@eap.gr

Abstract

The task of linking multiple databases with the aim to identify records that refer to the same entity is occurring increasingly in many application areas. If unique identifiers for the entities are not available in all the databases to be linked, techniques that calculate approximate similarities between records must be used for the identification of matching pairs of records. Often, the records to be linked contain personal information such as names and addresses. In many applications, the exchange of attribute values that contain such personal details between organisations is not allowed due to privacy concerns. The linking of records between databases without revealing the actual attribute values in these records is the research problem known as 'privacy-preserving record linkage' (PPRL). While various approaches have been proposed to deal with privacy within the record linkage process, a viable solution that is well applicable to real-world conditions needs to address the major aspect of scalability of linking very large databases while preserving security and linkage quality.

We propose a novel two-party protocol for PPRL that addresses scalability, security and quality/accuracy. The protocol is based on (1) the use of reference values that are available to both database owners, and allows them to individually calculate the similarities between their attribute values and the reference values; and (2) the binning of these calculated similarity values to allow their secure exchange between the two database owners. Experiments on a real-world database with nearly two million records yield linkage results that have a linear scalability to large databases and high linkage accuracy, allowing for approximate matching in the privacy-preserving context. Since the protocol has a low computational burden and allows quality approximate matching while still preserving the privacy of the databases that are matched, the protocol can be useful for many real-world applications requiring PPRL.

Keywords: Entity resolution, privacy technologies, scalability, approximate matching, similarity measure, binning, two-party protocol.

1 Introduction

In computer science, a long line of research has been conducted in probabilistic record linkage, based on the theoretical foundation provided by Fellegi & Sunter (1969). In today's world many organisations are collecting, storing, processing, analysing and mining fast-growing data sets that contain hundreds or even thousands of millions of records. Analysing such large data sets often requires information from multiple data sources to be aggregated in order to enable more detailed analysis. The process of matching and aggregating records that relate to the same entity from one or more data sets is known as 'record linkage', 'data matching' or 'entity resolution' (Elmagarmid et al. 2007, Herzog et al. 2007). Today, record linkage not only faces computational and operational challenges due to the increasing size of data collections, but also privacy and confidentiality challenges due to growing privacy concerns.

The problem of finding records that represent the same individuals in separate databases without revealing identifying details of these individuals is called the 'privacy-preserving record linkage' (PPRL), 'blind data linkage', or the 'private record linkage' problem (Churches & Christen 2004, Durham et al. 2011, Hall & Fienberg 2010, Verykios et al. 2009).

In the absence of a unique identifier for the entities stored in databases, exact or approximate similarity comparison techniques are applied to the common identifiers (which can contain personal information such as name, address and date of birth) for the identification of matching record pairs (Winkler 2006). Linking records by comparing the encrypted attribute values with a standard cryptographic technique in a three-party protocol seems to be a well-understood solution for PPRL (Churches & Christen 2004, O'Keefe et al. 2004). The attribute values match exactly if the corresponding encrypted values match, and the third party can link records without knowing the actual attribute values. However, a limitation of these methods is that only exact comparisons of values are possible. A small variation in an attribute value results in a completely different encrypted value. In practical applications, the exact matching of identifiers is not always possible due to variations or typographical and other types of errors in real-world data (Hernandez & Stolfo 1995). Applying approximate matching techniques overcomes this problem because these techniques do not rely on exact matches only. Therefore, an approach for approximate matching of identifiers in PPRL is required.

There have been several approaches proposed for approximate matching of identifiers in PPRL (Trepetin 2008). They can be classified into approaches that do or do not require a trusted third party for linkage, which are known as three-party and two-party protocols, respectively (Verykios et al. 2009). The advantages of two-party protocols over three-party protocols are that the former is more secure (because there is no possibility of collusion between two parties), and two-party protocols often have lower communication costs.

A practical PPRL application should address three main factors, which are scalability to large databases, linkage quality, and security. The aim of our paper is to propose a new two-party protocol for approximate matching that is practical in real-world PPRL applications by addressing these three factors. The paper also presents an evaluation of the proposed approach with regard to scalability, quality and security.

In the following section we provide an overview of related work in PPRL. In Section 3, we define the core problem of PPRL which is the focus of this paper, and we propose a scalable two-party solution using binning techniques. Section 4 presents our protocol in detail using small example data sets for illustration, and we analyse the security and the complexity of the protocol. We provide the results of the empirical evaluation of our work using large real-world data sets in Section 5. Finally, we conclude the paper with an outlook to future work in Section 6.

2 Related Work

Various methods for approximate matching in PPRL have been proposed, and several surveys have recently been conducted (Christen 2006*b*, Durham et al. 2011, Karakasidis & Verykios 2010, Trepetin 2008, Verykios et al. 2009). These methods can be classified into those that require a third party for performing the linkage and those that do not.

2.1 Three-party Protocols

A protocol is proposed by Song et al. (2000) that addresses the problem of approximate matching by calculating enciphered permutations for private approximate record matching. However, it is practically impossible to predict all possible permutations in real-world applications. Du et al. (2000) suggested a similar solution for private approximate record matching by pre-computing all possible permutations which are then enciphered. However, such an approach requires an unrealisable amount of storage and is susceptible to known plain text attacks.

A blindfolded multi-party approach is suggested by Churches & Christen (2004) that uses n-gram hash digests to achieve approximate private linkage. All matching hash values are compared using extra information such as the number of n-grams contained in each subset and the number of total n-grams comprising the original value. Though the cost is relatively low, compared to the enciphered permutations and pre-computed scores approaches, this is still a costly approach, because of the power set generation and computation it requires.

The work presented by Al-Lawati et al. (2005) introduces a multi-party secure token blocking protocol for achieving high performance private record linkage by using secure hash signatures for computing secure TF-IDF distances. In their work, three methods are explored which are known as simple blocking, record-aware blocking, and frugal third party blocking. These methods provide a trade-off between privacy and computation and communication cost.

Pang et al. (2009) suggested a protocol based on a set of reference strings common to the database owners. The database owners compute the distances between the reference strings and their attribute values and send the results to a third party that sums these distances and finds the minimum of this sum. If this minimum lies below a threshold the two original attribute values are classified as a match. The performance of the protocol depends crucially on the set of reference strings. In our protocol, we also use reference lists for securely calculating the similarities between attribute values in two databases, which are then compared without sending the actual attribute values to a third party.

A three-party protocol that provides privacy for both data and schema without revealing any information is presented by Scannapieco et al. (2007). This approach transforms records into an embedding metric space while preserving the distance between record values. It is assumed that the third party also holds a global schema to which the schemas of two database owners are mapped. A greedy re-sampling heuristic based on SparseMap is used to map values into a vector space at lower computational costs. However, the experimental results presented by Scannapieco et al. (2007) indicate that the linkage quality is affected by the greedy heuristic re-sampling method.

A hybrid approach that combines anonymisation techniques and cryptographic techniques to solve the private record linkage problem is proposed by Inan et al. (2008). This method uses value generalisation hierarchies in the blocking step, and the record pairs that cannot be blocked are compared in a computationally expensive secure multi-party computation (SMC) step using cryptographic techniques.

Using the one-to-many property of phonetic codes, an approach is proposed by Karakasidis & Verykios (2009) for performing approximate matching in PPRL. The attribute values are encoded using a phonetic encoding algorithm such as Soundex (Christen 2006*a*) and the resulting phonetic codes are mixed with randomly generated phonetic codes and sent to a third party to perform matching. The approach is secure and efficient for approximate matching but is not suitable for linking records based on numerical attributes, since phonetic codes are not suitable for numerical values.

2.2 Two-party Protocols

A two-party protocol is suggested by Atallah et al. (2003) that allows the parties to compute the distance (such as edit-distance) between strings without exchanging them. Due to the large amount of necessary communication required to compute the distance this protocol is unsuited for tasks involving large databases.

Ravikumar et al. (2004) use a secure set intersection protocol which requires extensive computations and is impractical for linking large databases. Yakout et al. (2009) present an approach that is based on transforming the data into vectors as described by Scannapieco et al. (2007) and comparing them without sending them to a third party. Complex numbers are calculated to create a complex plain and in the first step the likely matched pairs are computed by moving an adjustable width slab within the complex plain. The second step computes the actually matching pairs using a scalar product protocol based on randomised vectors.

3 Problem Statement and Proposed Solution

PPRL is the problem of how to identify matching records in different databases more effectively and faster without compromising privacy and security. In practice, the matching of two records can often be determined by similarity functions. Assume *Alice* and *Bob* are the two owners of their respective databases $\mathbf{D}^A$ and $\mathbf{D}^B$. They wish to determine which of their records $R_i^A \in \mathbf{D}^A$ and $R_j^B \in \mathbf{D}^B$ have an overall similarity $sim(R_i^A, R_j^B) \geq s_t$ according to some similarity function sim and minimum similarity threshold s_t. These record pairs are classified as matches. *Alice* and *Bob* agreed to disclose the actual values of some selected attributes of the record pairs that are classified as matches with each other. However, they do not wish to reveal the records that are not matches to each other or to any other party.

Due to the frequency of typographical variations and other errors (Hernandez & Stolfo 1995, Christen 2006*a*), record linkage algorithms have to employ string similarity functions for approximate matching of identifying attribute values such as names and addresses. Similarity functions, such as edit-distance or the Jaro-Winkler algorithm (Christen 2006*a*), are used to calculate the numerical similarity value of two attribute values. The core problem of a PPRL protocol is the calculation of the similarity of two records without revealing the attribute values to any other party.

The use of a public reference table which is common to both database owners has previously been proposed for PPRL in a three-party framework (Pang et al. 2009). Such public reference tables are available to both database owners and can be constructed either with random faked values or from a telephone directory, for example, by extracting all unique names, postcodes, and suburb names. The public reference table is used by the database owners to calculate the similarities between their attribute values and the reference values. These similarities are then sent to a third party that can link the records based on the triangular inequality property of the distance metrics, see Equation 1. Reference values are the values that are known to both database owners *Alice* and *Bob*, while the attribute values are the values that are only known to the corresponding database owner.

$$\begin{aligned} dist(v_i, r) + dist(v_j, r) &\geq dist(v_i, v_j) \\ (1 - sim(v_i, r)) + (1 - sim(v_j, r)) &\geq (1 - sim(v_i, v_j)) \\ 1 - sim(v_i, r) - sim(v_j, r) &\geq -sim(v_i, v_j) \\ sim(v_i, r) + sim(v_j, r) - 1 &\leq sim(v_i, v_j) \end{aligned} \quad (1)$$

Assume $dist(v_i, v_j)$ is the metric distance between two objects v_i and v_j, and $sim(v_i, v_j) = 1.0 - dist(v_i, v_j)$ is the corresponding similarity between the two objects. Similarity values are assumed to be normalised, such that $0 \leq sim(v_i, v_j) \leq 1$. For an exact match of the two objects the similarity function results in $sim(v_i, v_j) = 1.0$ and for two totally different objects $sim(v_i, v_j) = 0.0$. A distance-based similarity function mainly holds three properties: positivity ($dist(v_i, v_j) \geq 0$), symmetry and triangular inequality. A distance function is symmetric if $dist(v_i, v_j) = dist(v_j, v_i)$. The triangular inequality property states that the direct distance between two objects v_i and v_j is always less than or equal to the combined distance when going through a third object r: $dist(v_i, v_j) \leq dist(v_i, r) + dist(v_i, r)$. Reference values can be used as a third object (r) to calculate the similarity between the actual attribute values (v_i and v_j). Any similarity function that fulfils the conditions of a distance function can be used in this approach.

The similarity between attribute values and reference values $(sim(v_i, r), sim(v_j, r))$ can be calculated by the database owners individually and sent to a third party that can calculate the left hand side (LHS) of Equation 1 by calculating the combined similarity value $(sim(v_i, r) + sim(v_j, r) - 1)$. The third party then classifies all the record pairs as matches that have $sim(v_i, r) + sim(v_j, r) - 1 \geq s_t$, where s_t is a threshold value. If the LHS of Equation 1 is greater than s_t, then obviously the right hand side (RHS) of the equation, that is the actual similarity value $sim(v_i, v_j)$ between the two string values v_i and v_j, is also greater than s_t and therefore the pair (v_i,v_j) can be classified as a match. However, the results of an empirical evaluation of this approach conducted by Bachteler et al. (2010) shows inadequate linkage quality in terms of precision and recall. Increasing the size of the reference table improves the linkage quality to some extent but is impractical since this leads to very long run times.

We use the reverse triangular inequality of the distance metric, which is explained by Equation 2, to privately calculate the similarity of two values without exchanging the values.

$$\begin{aligned} |dist(v_i, r) - dist(v_j, r)| &\leq dist(v_i, v_j) \\ |(1 - sim(v_i, r)) - (1 - sim(v_j, r))| &\leq (1 - sim(v_i, v_j)) \\ |-sim(v_i, r) + sim(v_j, r)| &\leq (1 - sim(v_i, v_j)) \\ 1 - |sim(v_j, r) - sim(v_i, r)| &\geq sim(v_i, v_j) \end{aligned} \quad (2)$$

From the reverse inequality property of the similarity function, we can see that the value for $sim(v_i, v_j)$ (RHS) becomes higher and gets closer to 1.0 if and only if the values for $sim(v_i, r)$ and $sim(v_j, r)$ (LHS) become equal to each other, with r being an object from the reference table. This implies that if the difference between the similarity values of two objects with an object from the reference table is small, then they should be similar to each other.

The scalability factor of the record linkage process can be addressed by blocking the databases using indexing techniques. In large databases comparing all pairs of records is not feasible. The aim of indexing is to reduce this large number of potential comparisons by removing as many pairs of records as possible that correspond to non-matches. A recent survey (Christen 2011) presents detailed reviews of the indexing techniques that can be used in non-privacy-preserving record linkage applications. There have also been several approaches proposed towards this direction within a privacy-preserving setting (Al-Lawati et al. 2005, Inan et al. 2008, Yakout et al. 2009, Inan et al. 2010).

Our protocol indexes the data sets first by blocking the records based on a (phonetic) encoding function such as Soundex (Christen 2006*a*), and then uses public reference lists to generate one or several reference values for each block. These reference values are then used by the database owners to calculate the similarity between their attribute values and the reference values in each block. The similarity of each attribute value in a block is calculated by comparing the value only with the list of reference values that are in it's corresponding block.

Once the similarities are calculated we can perform the linkage by using a third party that links the records based on the triangular inequality of these similarities, as was done by Pang et al. (2009). Since

Table 1: Example Bins of Similarity Range

Bin Label	Start Range	End Range
A	0.5	0.625
B	0.626	0.750
C	0.751	0.875
D	0.876	1.0

Table 2: Matching Bin Combinations (MBC)

Match ID	Attribute 1 (Given Name)	Attribute 2 (Surname)
1	A,B	A,B
2	A,**B**	B,**C**
3	A,**B**	**C**,D
4	B,C	A,B
5	**B**,C	B,**C**
6	**B,C**	**C,D**
7	C,D	A,B
8	C,D	B,C
9	**C**,D	C,**D**

Table 3: Example Records and Reference Values

	Database Owner 1		Database Owner 2	
	Given Name	Surname	Given Name	Surname
Attribute Values	'millar'	'ameile'	'miller'	'amelia'
Reference Values	'myler'	'amalia'	'myler'	'amalia'
Similarity Values	0.7	0.8	0.8	0.9
Bin Labels	B	C	C	D
Match IDs	{1,2,3,4,5,6}	{2,3,5,6,8,9}	{4,5,6,7,8,9}	{3,6,9}
	{2,3,5,6}		{6,9}	

the similarities are pre-calculated this will reduce the run times for linkage and it will be scalable to large databases. This approach provides a scalable three-party solution for approximate matching in a PPRL scenario that can achieve high matching quality. As with other three-party protocols, security is however the major drawback in this three-party protocol. If one of the database owners colludes with the third party they can learn about the other database owner's private data.

Our aim is to develop a two-party protocol by using public reference lists and the reverse of triangular inequality property of distance metrics to measure similarities. If there is a way to exchange the calculated similarities between the database owners without revealing any information, we can simply eliminate the need of a third party for the linkage. Since both database owners know the public reference list values, sending the calculated similarity values can leak some information about the identifiers. We propose a two-party solution for this problem by binning the actual similarity values.

We split the similarity range into a number of bins k ($k > 1$), and each database owner stores the similarities between their attribute values and the reference values as bin labels into which the calculated similarity values fall into. The similarity values have a possible range from 0.0 to 1.0. Since we compare the attribute values only with the reference values that are in their corresponding block, the minimum similarity value will be larger than 0.0, and so we only need to bin similarities in an interval $[s_m, 1.0]$, with $s_m > 0.0$ selected by the user. Binning the similarity range from 0.5 to 1.0 into 4 bins, for example, is shown in Table 1. We will explain this example in detail further below.

We then calculate the Matching Bin Combinations (MBC) based on the binning distance d. The binning distance determines the maximum number of bin differences we allow for each attribute for the approximate matching of attribute values. For example, if the binning distance is $d = 1$ for each attribute and we use 2 attributes for the matching (and thus a total binning distance of $d = 2$), the MBC would be the ones that are given in Table 2.

Table 4: Subsets of bin combinations for the $(A, B/C, D)$ combination - Match ID 3 from Table 2

Database owner 1		Database owner 2		Total binning distance (d)
Given name	Surname	Given name	Surname	
A	C	A	C	(0+0) = 0
A	C	A	D	(0+1) = 1
A	C	B	C	(1+0) = 1
A	C	B	D	(1+1) = 2
A	D	A	C	(0+1) = 1
A	D	A	D	(0+0) = 0
A	D	B	C	(1+1) = 2
A	D	B	D	(1+0) = 1
B	C	A	C	(1+0) = 1
B	C	A	D	(1+1) = 2
B	C	B	C	(0+0) = 0
B	C	B	D	(0+1) = 1
B	D	A	C	(1+1) = 2
B	D	A	D	(1+0) = 1
B	D	B	C	(0+1) = 1
B	D	B	D	(0+0) = 0

Every candidate for the Matching Bin Combination is given a unique Match ID. Based on the MBC, each database owner calculates the set of Match IDs to which each of the records in their database corresponds to. Then these Match IDs are exchanged. Computing the intersection set of the Match IDs and then exchanging the records that are corresponding to those Match IDs between the database owners provides a two-party solution for our problem.

To illustrate our approach, assume we have two entities in two different databases with the values for the attributes 'Surname' and 'Given Name' as ('millar','ameile') and ('miller','amelia'), as shown in Table 3. Applying the Soundex (Christen 2006*a*) phonetic encoding to these values results in the two blocks 'm460' and 'a540'. Assume that the reference list contains one value for each of these blocks, and they are 'myler' for 'm460' and 'amalia' for 'a540'.

Comparing the attribute values with the corresponding block reference values gives us the similarity values of (0.7,0.8) for ('millar','ameile') and (0.8,0.9) for ('miller','amelia'), which result in the bin combinations (B,C) and (C,D), respectively (see Table 1 for the bin ranges). According to the MBC in Table 2, the corresponding matches would be Match IDs $\{2, 3, 5, 6\}$ and Match IDs $\{6, 9\}$, because the bin combination of B for attribute 'Surname' and C for 'Given Name' appears in Match IDs $2, 3, 5$ and 6, whereas the combination of C and D appears in Match IDs 6 and 9 only.

The intersection of these two sets results in set $\{6\}$, which is considered to be a match combination for these two example records, and so the two entities can be classified as a match. If the intersection list is empty the entities do not match.

The MBC calculated here are supersets of all the subsets of bin combinations. For example, if we consider the bin combination of Match ID 3, the subsets of bin combinations would be as shown in Table 4. As shown in this table, if the combination $(A, B/C, D)$ is a match then all subsets of this combination are also matches, since they all have 2 or less than 2 binning distance. This improves the security factor of our approach because there can be many possible matching combinations (16 in this example) for one Match ID.

This parametric solution requires the number of bins k to be determined before the linkage. The selection of k is crucial for the performance of the protocol

Table 5: Notation used in this paper

$\mathbf{D^A}$, $\mathbf{D^B}$	Databases held by database owners Alice and Bob, respectively
$R_i^{\mathbf{A}}$, $R_j^{\mathbf{B}}$	A record in $\mathbf{D^A}$ or $\mathbf{D^B}$, respectively
A, a	Attributes common to $\mathbf{D^A}$ and $\mathbf{D^B}$ that are used for linking, an attribute $a \in A$
v, v_i, v_j	An individual attribute value
r, r_i, r_j	A reference value
$block_a(\cdot)$	Function used to block/index attribute a
b	A blocking key value (BKV): $b = block_a(\cdot)$
c	A compound BKV (CBKV): $c = [block_{a_1}(\cdot), \ldots block_{a_{\lvert A \rvert}}(\cdot)]$
$sim_a(\cdot)$	Function used to calculate similarities between values in attribute a
s_m	Minimum similarity threshold to determine the similarity range $[s_m - 1.0]$
k, d	Number of bins, Maximum number of bin differences to find the matching bin combinations
$enc(\cdot, h)$	Function and key used to hash-encode values
BI, $\mathbf{BI}_a$, **BLI**	Block Index, Block Index for attribute $a \in A$, Block List Index
RLI, $\mathbf{RLI}_a$	Reference List Index for attribute $a \in A$
MBC, **MLI**	Matching Bin Combinations, Match ID List Index

as the three major factors of PPRL protocols, security, scalability and linkage quality, depend on this parameter. The larger the number of bins the smaller the range of each bin is, which results in higher accuracy of the protocol. But the smaller the number of bins the smaller the computational complexity is, as the number of candidates of matching bin combinations is reduced, and the more secure the protocol is due to the higher range of bins. So the number of bins must be carefully chosen. We will experimentally investigate how these three factors are affected by the number of bins in Section 5.

4 A Two-Party Protocol for Scalable and Approximate Secure Matching

In this section we will illustrate the steps (S1 to S9) of our protocol in detail using an example consisting of two small databases with given names and surnames used as the linkage attributes. The notation used throughout the paper is summarised in Table 5.

S1: Alice and Bob agree upon (a) a list of attributes A to be used for the linkage and their priority order such as primary attribute, secondary attribute, etc; (b) one blocking function (phonetic) $block_a(\cdot)$ for each attribute $a \in A$, used to generate blocking key values (BKV) b; (c) a similarity function $sim_a(v, r)$, used to calculate the numerical similarity for a pair of values v and r, where v is an attribute value and r is a reference value, such that for an exact match ($v = r$) $sim_a(v, r) = 1$ and for two totally different values $sim_a(v, r) = 0$; (d) a minimum similarity threshold s_m, which determines the start range of the first similarity bin; (e) the number of bins k to be used; (f) a binning distance d used for finding the candidates of Matching Bin Combinations (MBC) for each attribute; (g) a hash-encoding function $enc(\cdot, h)$ and a corresponding hash key h, used to encode the Compound BKVs (CBKVs), reference lists and finally matching records before they are being exchanged between the database owners. This hash-encoding function can for example be the HMAC (Hashed Message Authentication Code) function (Krawczyk et al. 1997), which encodes a plain-text string into a unique hash-code such that having access to a hash-code only makes it impossible with the current computing techniques to find the plain-text string in a reasonable amount of time. To simplify the illustration we do not apply any hash-encoding function in the example.

S2: Alice and Bob each read their databases (example databases are shown in Figure 1) and independently build their local Block Index (BI) data structures for each linkage attribute, and a Block List Index (BLI) data structure by indexing their databases using the blocking function $block_a(\cdot)$, as is illustrated in Figures 2 and 3. The BI data structures are implemented as an inverted index (Witten et al. 1999). The index keys are the unique encodings of a linkage attribute (the BKVs), and the corresponding lists contain the actual attribute values in a block. The BLI data structure is implemented as a nested inverted index where the keys are the unique encodings of the primary linkage attribute and the values are again inverted indexes with keys being the unique encodings of the secondary linkage attribute and values being the list of unique encodings of the third linkage attribute, for example if the number of linkage attributes is three. The nested inverted indexes for two linkage attributes are shown in Figure 3.

S3: Alice and Bob exchange their BLI data structure with each other. This communication is encrypted, for example using public key encryption (Schneier 1995), such that only Alice and Bob can decrypt each others values. Once the BLI is exchanged, Alice and Bob can generate an intersection list of BLIs, as is illustrated in Figure 3. Exchanging the BLIs to find out the intersection list, which is the list of compound blocks c (individual blocks b for each linkage attribute are grouped to generate the compound block) that are common to both databases, might leak some information about each others data. In order to overcome this, a secure set intersection protocol can be used that enables to find the intersection list of BLI lists securely (Agrawal et al. 2003, Kissner & Song 2005). This is discussed in detail in Section 4.2. Alice and Bob then sort the intersection BLI and find the common individual blocks for each linkage attribute separately, as is illustrated in Figure 4.

S4: The next step is to generate the Reference List Index (RLI) which contains lists of reference values, one for each individual block in the intersection list of BLI. The RLI can be generated by both parties together, for example one could generate reference lists for odd blocks and the other for even blocks, or one for primary attribute blocks and the other for secondary attribute blocks. This is shown in Figure 5. In our example, we assume the number of reference values generated for each block is 1.

S5: Alice and Bob then build their Similarity Index (SI). For each unique individual block b in the intersection BLI, they calculate the similarity of

D^A

RecID	Surname	GivenName
RA1	millar	robert
RA2	myler	amelia
RA3	millar	gail
RA4	peter	robart
RA5	peterra	gail
RA6	smyth	amelie

D^B

RecID	Surname	GivenName
RB1	millar	amelia
RB2	miller	roberto
RB3	petera	gayle
RB4	smitth	amilia
RB5	smeth	amelie
RB6	smeetth	rupert

Figure 1: Example databases held by Alice ($\mathbf{D^A}$) and Bob ($\mathbf{D^B}$) with Surname and Given name attributes, used to illustrate the protocol described in Section 4.

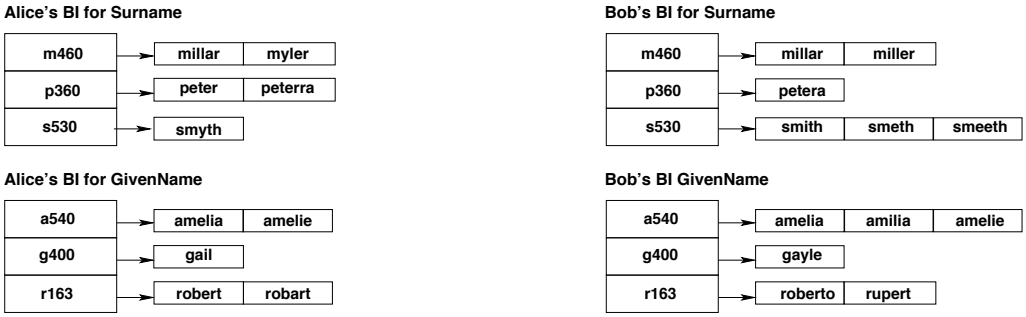


Figure 2: The Block index (BI) of Alice and Bob for the Surname and Given name attributes. The BI is generated in S2 of the protocol as the databases are loaded, and is used in S5 to build the similarity index.

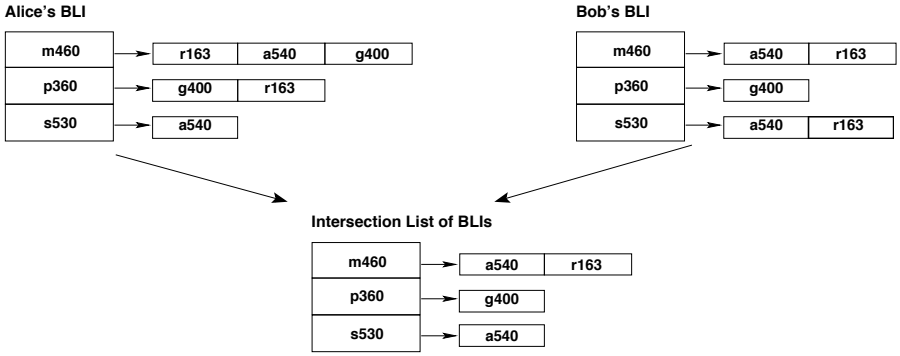


Figure 3: The Block List Index (BLI) of Alice and Bob and the Intersection list of BLIs. Exchanging the BLI in order to calculate the intersection list of the BLIs can reveal some information about the other party's data. This is discussed in detail in Section 4.2. The BLI is generated in S2 of the protocol.

Figure 4: The compound blocks c in the sorted intersection list of the BLIs and the individual blocks b for each linkage attribute. The intersection list of BLIs is sorted and the individual blocks are found in S3 of the protocol. The compound blocks are sorted and given index numbers which will be needed in S7 of the protocol.

Figure 5: The reference lists for primary BKVs (Surname attribute) and secondary BKVs (Given name attribute). In the example, we use one reference value per list. These lists are generated in S4 of the protocol.

each unique attribute value in that block (which is stored in their BI as is generated in S2) with the list of reference values of that block, which is retrieved from the RLI. Figure 6 illustrates this for the running example.

S6: In the next step the database owners build the bins with their similarity ranges and the Matching Bin Combinations (MBC), as is illustrated in Figure 7. The bins are stored as an inverted index data structure where the keys are the bin

Alice's SI

Block 'm460' (Surname)

	millar	myler
malar	D (0.8)	C (0.7)

Block 'a540' (GName)

	amelia	amelie
amilia	D (0.8)	C (0.7)

Block 'p360' (Surname)

	peter	peterra
peter	E (1.0)	D (0.8)

Block 'g400' (GName)

	gail
gail	E (1.0)

Block 's530' (Surname)

	smyth
smith	E (0.9)

Block 'r163' (GName)

	robart	robert
robert	E (0.9)	E (1.0)

Bob's SI

Block 'm460' (Surname)

	millar	miller
malar	D (0.8)	C (0.7)

Block 'a540' (GName)

	amelia	amilia	amelie
amilia	D (0.8)	E (1.0)	C (0.7)

Block 'p360' (Surname)

	petera
peter	E (0.9)

Block 'g400' (GName)

	gayle
gail	D (0.8)

Block 's530' (Surname)

	smitth	smeth	smeeth
smith	E (0.9)	E (0.9)	D (0.8)

Block 'r163' (GName)

	roberto	rupert
robert	E (0.9)	D (0.8)

Figure 6: The Similarity Index (SI) which contains the similarities between attribute values and their corresponding reference values calculated using the Jaro-Winkler (Christen 2006*a*) approximate string comparison function, rounded to one digit, along with their corresponding bins. The SI is generated in S5 of the protocol.

Bin Intervals

Label	Range
A	0.5 – 0.59
B	0.6 – 0.69
C	0.7 – 0.79
D	0.8 – 0.89
E	0.9 – 1.0

Matching Bin Combinations

Match ID	Surname	GName
1	A–B	A–B
2	A–B	B–C
3	A–B	C–D
4	A–B	D–E
5	B–C	A–B
6	B–C	B–C
7	B–C	C–D
8	B–C	D–E
9	C–D	A–B
10	C–D	B–C
11	C–D	C–D
12	C–D	D–E
13	D–E	A–B
14	D–E	B–C
15	D–E	C–D
16	D–E	D–E

Figure 7: The bins of similarity and the Matching Bin Combinations (MBC). The bins and their ranges are agreed upon by the database owners in S1 of the protocol. In this example, the number of bins is $k = 5$ and the similarity range is 0.5 to 1.0. The MBC are calculated based on the bins and the binning distance d. In this example, $d = 1$. The bin combinations are generated only for one compound block. Using this, the Match IDs can be calculated using Equation 3 for bin combinations in other blocks as well.

labels/indices and the values are the lists of starting value and ending value of the ranges of each bin. The similarity range between s_m and 1.0 is split into k bins. Based on the bins and the binning distance d, the MBCs are generated with the corresponding Match IDs for each candidate.

S7: Alice and Bob go through their database and build their local Matching Bins of Records (MBR) data structure, as is shown in Figure 8. The MBR data structure contains unique tuples of encoding values of linkage attribute values (CBKVs) and for each unique CBKV, c, it contains an inverted index with keys being the unique tuple of attribute values in that compound block, and values being the lists that contain (a) a list of bin labels for each of the attribute value (which is retrieved from SI, as is generated in S5), (b) a list of Match IDs that correspond to this combination of bin labels (which is retrieved from MBC, as is generated in S6) by using Equation 3, and (c) a list of record IDs that contain this unique tuple of attribute values.

$$\begin{aligned} MatchID &= (compound\_block\_index\_number \\ &\quad \times number\_of\_candidates\_in\_MBC) \\ &\quad + Match\_ID\_in\_MBC \end{aligned} \tag{3}$$

It is important to note that the MBC data structure is calculated only for one compound block because all the compound blocks will have the same set of candidates of matching bin combinations. The compound blocks in the intersection list of the BLIs is sorted in S3 to find the index numbers of these sorted compound blocks. For example, consider the compound block of $c = [`p360', `g400']$ in Figure 4. It is in the 3rd position in the sorted intersection BLI list. We use a zero-based index in our example, therefore the compound block index number would be 2. The number of candidates in the MBC is 16 in our example. A record with the bin combination of 'D' for the Surname and 'E' for the Given name attribute corresponds to Match IDs 12 and 16 in the MBC (Figure 7). Using Equation 3, the actual Match IDs for this record would be calculated as $(2 \times 16 + 12)$ and $(2 \times 16 + 16)$ which are equal to Match IDs of 44 and 48, the Match IDs for RA5 in Alice's MBR.

S8: Once the MBRs are generated, Alice and Bob retrieve the list of unique Match IDs from their MBR. They then exchange their list of Match IDs with each other and find the intersection of Match IDs, which contains the Match IDs that are common to both database owners. This step is illustrated in Figure 9.

S9: In the final step, as is illustrated in Figures 10 and 11, both Alice and Bob exchange the records (record identifiers) of the matches with each other that are corresponding to the Match IDs in the intersection list of Match IDs. The accumulator is built for storing these matching records, as is shown in Figure 11.

Alice's MBR

BKV Tuple	Surname	GName	Surname Bin	GName Bin	Match IDs	Rec ID
(m460,r163)	millar	robert	D	E	12,16	RA1
(m460,a540)	myler	amelia	C	D	23,24,27,28	RA2
(m460,g400)	millar	gail	D	E	---	RA3
(p360,r163)	peter	robart	E	E	---	RA4
(p360,g400)	peterra	gail	D	E	44,48	RA5
(s530,a540)	smyth	amelie	E	C	63	RA6

Bob's MBR

BKV Tuple	Surname	GName	Surname Bin	GName Bin	Match IDs	Rec ID
(m460,a540)	millar	amelia	D	D	27,31	RB1
(m460,r163)	miller	roberto	C	E	12,16	RB2
(p360,g400)	petera	gayle	E	D	47	RB3
(s530,a540)	smitth	amilia	D	E	60,64	RB4
(s530,a540)	smeth	amelie	D	C	58,59,62,63	RB5
(s530,r163)	smeeth	rupert	D	D	---	RB6

Figure 8: The Matching Bins of Records (MBR) of Alice and Bob. For each of the unique tuple of encoding values (BKVs), it contains the combination of surname and given name attribute values with their corresponding bin labels, a list of Match IDs, and a list of record identifiers that contain the combination. The MBRs are generated in S7 of the protocol. The Match IDs are calculated only for the records that belong to the compound blocks that are in the intersection list of BLIs. The records RA3, RA4 and RB6 in this example belong to the compound blocks of ['m460','g400'], ['p360','r163'], and ['s530','r163'], respectively, which are not in the intersection list of BLIs in Figure 4. In other words, these compound blocks are not common in both databases.

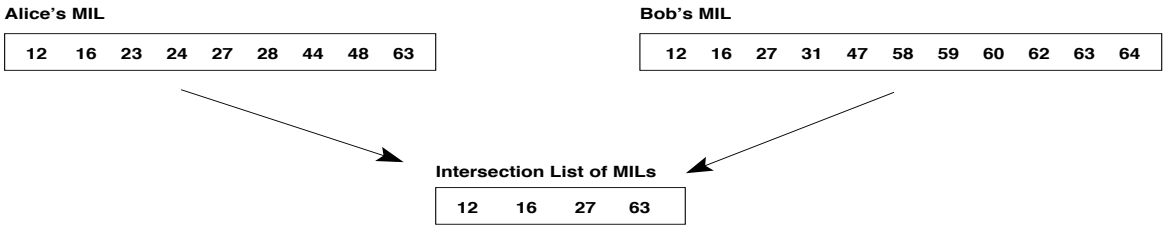


Figure 9: The Match ID List (MIL) of Alice and Bob which contains a list of Match IDs found in their records and the intersection list of MILs. The MILs are generated in S8 of the protocol.

Alice's Matches

Match ID	Rec ID	Surname	GName
12	RA1	millar	robert
16	RA1	millar	robert
27	RA2	myler	amelia
63	RA6	smyth	amelie

Bob's Matches

Match ID	Rec ID	Surname	GName
12	RB2	miller	roberto
16	RB2	miller	roberto
27	RB1	millar	amelia
63	RB5	smeth	amelie

Figure 10: The matches of Alice and Bob for the corresponding Match IDs in the Intersection list of MILs, which are generated and exchanged in S9 of the protocol.

Accumulator

Alice	Bob
RA1	RB2
RA2	RB1
RA6	RB5

Figure 11: The Accumulator generated by Alice and Bob which contains the actual matching record pairs. The accumulator is generated in S9 of the protocol.

4.1 Complexity Analysis

In this section we analyse the computation and communication complexity of our two-party protocol. We assume both databases contain N records, $I = |A|$ attributes that are used for linking the records, and each linkage attribute contains M unique values. Each attribute generates B blocks by applying the $block_a(\cdot)$ functions to index their databases. It is obvious that for large databases it commonly holds that $I \ll B \leq M \ll N$.

In step S1, the agreement of the required functions between Alice and Bob has a constant communication complexity. Reading the databases in step S2 and building the local BI data structures and the BLI data structure requires $O(N)$ of computational complexity if I, B and M are very small compared to N, because building the BI and BLI data structures are $O(I \times M)$ and $O(I \times B)$, respectively.

The exchange of the BLI in step S3 requires the communication of $I \times B$ values for each party, and with I, the number of linkage attributes, being comparatively a very small constant this results in an $O(B)$ communication complexity. Assuming each BLI contains B compound blocks ($I \times B$ individual blocks) calculating the intersection of the two BLIs takes $B \times log(B)$, which results in $O(B\, log(B))$ computation complexity.

We assume the number of reference values used for each individual block in the intersection list of the BLIs is on average R. In step S4, the number of reference values to be generated and exchanged is $I \times B \times R$. With R and I being very small compared to B, this step requires a computation and communication complexity of $O(B)$.

In step S5, assuming each list in the BI that was generated in step S1 contains on average M/B attribute values, each of the $I \times B$ individual blocks requires $(M/B) \times R$ similarity calculations, and thus a total of $I \times M \times R$. Again with I and R being very small, the computation complexity of step S5 is $O(M)$.

Candidates of Matching Bin Combinations are calculated for one compound block based on the number of bins k, the similarity range (which includes the minimum similarity value s_m and the maximum similarity value 1.0) and the binning distance d for each attribute. For each of the candidates a unique Match ID is given. This can be used to calculate the Match IDs for any bin combination in any compound block using Equation 3. The number of candidates is given by $(k-d)^I$ for one compound block and thus the computation complexity is $O((k-d)^I)$.

In step S7, building the MBR by reading the N records requires a total of $O(N)$ computation complexity. In the next step (S8) Alice and Bob exchange the unique Match IDs that are corresponding to the matching combinations found in their records. This is of size $(k-d)^I \times B$, because a maximum of $(k-d)^I$ candidates are calculated for one compound block and each candidate has a corresponding unique Match ID. With the total number of compound blocks being B, this step results in $O((k-d)^I \times B)$ communication complexity. Finding the intersection of these two lists requires $O((k-d)^I B\, log((k-d)^I B))$.

Finally, the generation of the accumulator to store the matches using the Match IDs in the intersection list requires a computation complexity of $O((k-d)^I \times B)$, because a maximum of $(k-d)^I \times B$ Match IDs can be found in the intersection list.

Overall, the communication and computation complexities of our protocol are linear in the size of the databases $O(N)$ and the number of blocks $O(B)$, but is of exponential complexity in the number of attributes I and bins k, $O(k^I)$. The complexity of our protocol greatly depends on the value of k.

4.2 Security Analysis

The protocol assumes that both parties follow the 'honest but curious' behaviour (HBC) (Hall & Fienberg 2010). Both parties are curious, in that they try to find out as much as possible about the other party's inputs while following the protocol. The protocol is secure in the HBC perspective if and only if both parties have no new knowledge at the end of the protocol above what they would have learned from the output of the matched record pairs. We analyse the security of our protocol by discussing what the two parties learn from the data they communicate with each other during the protocol.

There are mainly two steps where we have to consider the security factor in our protocol. One is the exchange of the BLI (that contains compound blocks) which might leak some information regarding the combination of block values in each party's database to other party. Using a secure set intersection (SSI) protocol to find out the intersection set of the compound blocks in each database (without revealing any additional information to either party) will solve this problem. There are two major types of SSI protocols that are commutative encryption (Agrawal et al. 2003) and homomorphic encryption (Kissner & Song 2005).The encryptions of both types of SSI protocols have a linear communication complexity. Since the exchange of the BLI is of $O(B)$ size (see Figure 12), using a SSI protocol for this step is feasible.

The second security issue in our protocol is at the step of exchanging the Match IDs to find the intersection list that contains the Match IDs of matching bin combinations that are common to both databases. This depends on the number of bins. If the number of bins is high then the range of each bin is low and thus the average number of unique attribute values that fall in each bin will be smaller. This results in less overlap in the Matching Bin Combinations (which means some of the candidate Matching Bin Combinations will not be found in any of their records) and allows inferring the combinations of bins in which no records exist when exchanging the Match IDs. If the number of bins is high then the probability of getting bins which do not have any attribute values in them is also high. So the lower the value for the number of bins the higher the security of our protocol.

4.3 Accuracy Analysis

Evaluating the accuracy of our protocol is crucial since we use the bins of similarity values instead of the actual similarity values for the approximate matching of attribute values. In this section we analyse the accuracy of our protocol in terms of the metrics such as precision, recall and f-measure, which are commonly used in Information Retrieval (Raghavan et al. 1989, Manning & Schütze 1999). Based on the classification for true positive (TP), false positive (FP), false negative (FN) and true negative (TN) pairs, the accuracy measures are defined as shown in Equation 4.

$$\begin{aligned} precision &= \frac{\sum TP}{\sum TP + \sum FP} \\ recall &= \frac{\sum TP}{\sum TP + \sum FN} \\ f\text{-}measure &= 2\left(\frac{precision \times recall}{precision + recall}\right) \end{aligned} \tag{4}$$

Accuracy depends on the number of bins. If the number of bins is high then the range of each bin is small which results in more specific ranges of similarity values, and thus the number of FPs and FNs will be less, resulting in higher precision and recall. However, if we increase the number of bins and thus decrease the range of each bin above a certain value then the number of FNs will start to increase, because the number of matches missed as non-matches (FNs) increases. But still the precision is high with an increasing number of bins.

As a result, the f-measure which is the harmonic mean of precision and recall, is expected to increase with the number of bins up-to a certain value. The larger the number of bins used the higher the accuracy of our protocol should be.

5 Experimental Evaluation and Discussion

In this section we present the results of experiments conducted using a real Australian telephone database containing 6,917,514 records. We extracted four attributes commonly used for record linkage: Given name (with 78,336 unique values), Surname (with 404,651 unique values), Suburb (town) name (13,109 unique values), and Postcode (2,632 unique values). These four attributes allowed us to evaluate how the number of attributes used for linkage affects the performance of our protocol. We generated data sets of various sizes by sampling 0.01%, 0.1%, 1%, 10% and 100% of records in the full database twice each in such a way that we obtained pairs of data sets that had an overlap where 25%, 50%, or 75% of records appeared in both the sampled data sets. Table 6 provides an overview of the generated data sets.

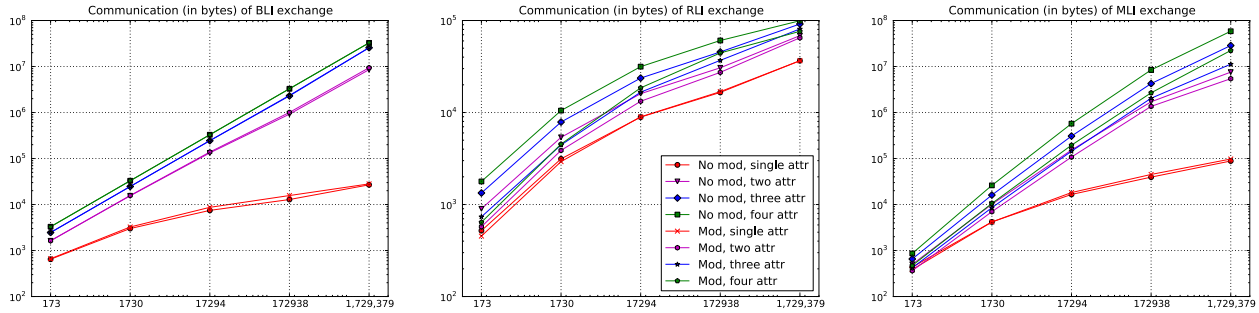


Figure 12: Amount of communication required for different number of linkage attributes, averaged over the results of both database owners over all data set variations as described in Section 5.

Table 6: The number of records in the data sets used for experiments, and the number of records that overlap (i.e. occur in both data sets of a pair). This is considered as the number of true matches.

Data set sizes	25% overlap	50% overlap	75% overlap
173 / 173	38	86	130
1730 / 1730	446	897	1310
17,290 / 17,290	4365	8611	12,973
172,938 / 172,938	42,980	86,363	129,542
1,729,379 / 1,729,379	432,538	86,487	1,297,029

All records that occur in both the data sets are exact matches, which are labelled as 'No mod' (for no modification) in the results figures. To evaluate the performance of approximate matching in the context of 'dirty data' (where attribute values contain typographical errors and variations), we generated another set of data sets (labelled as 'Mod') where we modified two attribute values in each record by applying one randomly selected character edit operation (insert, delete, substitute or transposition).

As encoding functions for blocking the data sets we used Soundex (Christen 2006*a*) for the Given name, Surname and Suburb attributes, while for the Postcode attribute we took the first three digits of the value as the blocking key. The Jaro-Winkler (Christen 2006*a*) string comparison function was used for Given name, Surname, and Suburb name values, while Edit-distance (Christen 2006*a*) was used as a comparison function for Postcode values. The similarity range was set between $s_m = 0.4$ and 1.0 and the binning distance (number of bin differences) for the matching for each linkage attribute was set to $d = 1$ and with 4 attributes to $d = 4$.

We implemented a prototype to evaluate the performance of our protocol using the Python programming language (version 2.7.1). We simulated communication between the parties by creating a directory for each party and writing the communicated data into a file in the receiver's directory. We did not apply any encryption to the communicated data for easy inspection of the files written. All tests were run on a 64-bit Intel Core i7 (2.7 GHz), 8 GBytes of main memory computer running the Ubuntu 11.04 OS platform. The prototype and test data sets are available from the authors.

5.1 Discussion

Figures 12 and 13 shows the scalability of our protocol. Computation complexity is assessed as the total run time required for the linkage, and communication complexity is assessed by the size of the files into which the communicated data are written. All variations of the data sets were used with all the combination of all four attributes. The similarity range was set as $s_m = 0.5$ to 1.0. The value for the number of bins k was set to 5 for these experiments. The results of both the exact and the approximate matching ('No Mod' and 'Mod') are shown in the figures.

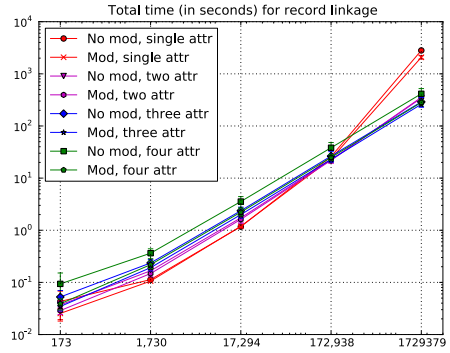


Figure 13: Total run time required for the linkage for different number of linkage attributes, averaged over the results of both database owners over all data set variations as described in Section 5.

As can be seen from Figure 12, the communication complexity of our protocol is linear or sub-linear in the size of the data sets. It increases with the number of attributes I used for linkage. With a smaller number of attributes used, the communication complexity tends to be more sub-linear while with all four attributes used it becomes linear in the size of the data sets.

As expected, the computation complexity of our protocol is linear in the size of the data sets, and it increases with the number of attributes I used for the linkage. Most of the steps in our protocol depend on the number of linkage attributes I and the number of bins k used. However, the linkage performed with one attribute takes longer than with two, three and even four attributes, especially for larger data sets. All the steps performed after the step of calculating the intersection list of the BLIs (step S3) are dependent on this intersection list. With only one linkage attribute there may be many values that exist in both databases and thus the intersection list of the BLIs will be larger than when more than one attribute is used. As a result, the calculation of similarities of these attribute values, the generation of the Matching Bins of Records and building the accumulator takes more time with one attribute only than performing the linkage with several attributes.

Figure 14 presents the experimental results of the complexity, accuracy and security of the protocol for different number of bins k. In these three experiments, all variations of the data sets were used with

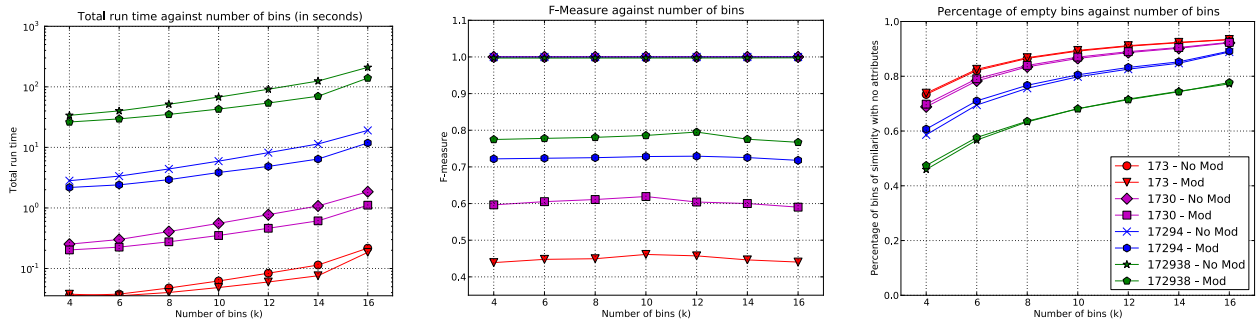


Figure 14: Complexity (left), accuracy (middle), and security (right) of the linkage for different number of bins k, averaged over the results of both database owners over all variations of each data set.

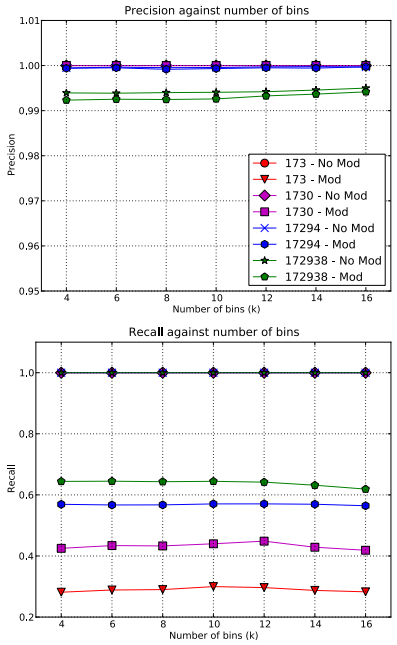


Figure 15: Precision and recall of the linkage, averaged over the results of both database owners over all data set variations as described in Section 5.

'Given name', 'Surname' and 'Postcode' attributes used for the linkage.

The run time is calculated for different number of bin values to evaluate the complexity of the protocol and how it is influenced by the value for k.

Accuracy is measured by running the protocol with different values for the number of bins k and measuring the f-measure as indicated in Equation 4. A pair of records is considered as a true match in our experiments if they have the same record ID. As the results show, our protocol can achieve high accuracy by tuning the parameter k to an optimal value. To analyse the accuracy of our protocol in more detail, we also provide the results of accuracy in terms of precision and recall when k was set to 5 in Figure 15. As discussed in Section 4.3, although the precision increases with k, recall starts decreasing at some point since the number of true matches missed as non-matches starts increasing.

Security is assessed by the percentage of bins that do not have any attribute values when running the protocol with different values for the number of bins, k. As we discussed in Section 4.2, the number of bins used determines the security of our protocol. When exchanging the Match IDs with each other the bin combinations that do not have any records might leak some information. So the percentage of empty bins can be used as a measure to evaluate the security of our protocol. Even with the knowledge of empty bins it is difficult to infer the actual attribute values and also there can be many possible subsets of combinations for a single matching combination (see Table 4).

As we discussed in Sections 4.1 to 4.3, the accuracy and complexity of our protocol increases with the number of bins while the security decreases. This explains that the choice of value for the number of bins, k, is crucial for our protocol.

6 Conclusion

In this paper, we have presented a novel two-party protocol for scalable approximate matching for privacy-preserving record linkage by using reference values and binning the similarity ranges for secure calculation of the similarities between attribute values. Our protocol is linear in the size of the databases to be linked which allows scalability to large databases. This has been validated in our experimental evaluation where we performed the linkage on data sets of up to a size of nearly two million records. However, our protocol is a parametric solution which depends on the number of bins.

As shown in the experimental evaluation the number of bins plays a major role in our protocol in determining the three main factors of the privacy-preserving record linkage protocol, which are security, scalability and accuracy. We aim to tackle this problem of finding the optimal value for the number of bins in our future work. Specifically, we will investigate both analytically and empirically the combinations of security, complexity and accuracy of our protocol with the values for the number of bins.

In our current implementation, we used the Jaro-Winkler string comparison function to measure the similarity between two strings. Another extension to our current work is to compare the performances of the protocol when different approximate string comparison functions are used. We will also investigate how parallelism can improve the performance and security of our protocol. Upon the best determination of the value for the number of bins, our two-party protocol performs well in real-world privacy-preserving record linkage applications.

References

Agrawal, R., Evfimievski, A. & Srikant, R. (2003), Information sharing across private databases, *in* 'ACM SIGMOD', ACM, pp. 86–97.

Al-Lawati, A., Lee, D. & McDaniel, P. (2005), Blocking-aware private record linkage, *in* 'International Workshop on Information Quality in Information Systems', pp. 59–68.

Atallah, M., Kerschbaum, F. & Du, W. (2003), Secure and private sequence comparisons, *in* 'ACM workshop on Privacy in the Electronic Society', pp. 39–44.

Bachteler, T., Schnell, R. & Reiher, J. (2010), An empirical comparison of approaches to approximate string matching in private record linkage, *in* 'Proceedings of Statistics Canada Symposium 2010. Social Statistics: The Interplay among Censuses, Surveys and Administrative Data'.

Christen, P. (2006*a*), A comparison of personal name matching: Techniques and practical issues, *in* 'Workshop on Mining Complex Data, held at IEEE ICDM'06', Hong Kong.

Christen, P. (2006*b*), Privacy-preserving data linkage and geocoding: Current approaches and research directions, *in* 'Workshop on Privacy Aspects of Data Mining, held at IEEE ICDM'06', Hong Kong, pp. 497–501.

Christen, P. (2011), 'A survey of indexing techniques for scalable record linkage and deduplication', *IEEE Transactions on Knowledge and Data Engineering* .

Churches, T. & Christen, P. (2004), 'Some methods for blindfolded record linkage', *BioMed Central Medical Informatics and Decision Making* **4**(9).

Du, W., Atallah, M. & Kerschbaum, F. (2000), Protocols for secure remote database access with approximate matching, *in* 'Proceedings of the 1st ACM Workshop on Security and Privacy in E-Commerce'.

Durham, E., Xue, Y., Kantarcioglu, M. & Malin, B. (2011), 'Quantifying the correctness, computational complexity, and security of privacy-preserving string comparators for record linkage', *Information Fusion* **In Press**.

Elmagarmid, A. K., Ipeirotis, P. G. & Verykios, V. S. (2007), 'Duplicate record detection: A survey', *IEEE Transactions on Knowledge and Data Engineering* **19**(1), 1–16.

Fellegi, I. P. & Sunter, A. B. (1969), 'A theory for record linkage', *Journal of the American Statistical Society* **64**(328).

Hall, R. & Fienberg, S. (2010), Privacy-preserving record linkage, *in* 'Privacy in Statistical Databases, Springer LNCS 6344', Corfu, Greece, pp. 269–283.

Hernandez, M. A. & Stolfo, S. J. (1995), The merge/purge problem for large databases, *in* 'ACM SIGMOD'95', San Jose.

Herzog, T., Scheuren, F. & Winkler, W. (2007), *Data quality and record linkage techniques*, Springer Verlag.

Inan, A., Kantarcioglu, M., Bertino, E. & Scannapieco, M. (2008), A hybrid approach to private record linkage, *in* 'IEEE ICDE', pp. 496–505.

Inan, A., Kantarcioglu, M., Ghinita, G. & Bertino, E. (2010), Private record matching using differential privacy, *in* 'International Conference on Extending Database Technology'.

Karakasidis, A. & Verykios, V. (2009), Privacy preserving record linkage using phonetic codes, *in* 'Fourth Balkan Conference in Informatics', IEEE, pp. 101–106.

Karakasidis, A. & Verykios, V. (2010), Advances in privacy preserving record linkage, *in* 'E-activity and Innovative Technology, Advances in Applied Intelligence Technologies Book Series', IGI Global, pp. 22–34.

Kissner, L. & Song, D. (2005), Private and threshold set-intersection, Technical report, Carnegie Mellon University.

Krawczyk, H., Bellare, M. & Canetti, R. (1997), HMAC: Keyed-hashing for message authentication, *in* 'Internet RFCs'.

Manning, C. & Schütze, H. (1999), *Foundations of statistical natural language processing*, Vol. 59, MIT Press.

O'Keefe, C., Yung, M., Gu, L. & Baxter, R. (2004), Privacy-preserving data linkage protocols, *in* 'Proceedings of the 2004 ACM workshop on Privacy in the electronic society', pp. 94–102.

Pang, C., Gu, L., Hansen, D. & Maeder, A. (2009), 'Privacy-preserving fuzzy matching using a public reference table', *Intelligent Patient Management* pp. 71–89.

Raghavan, V., Bollmann, P. & Jung, G. (1989), 'A critical investigation of recall and precision as measures of retrieval system performance', *ACM Transactions on Information Systems (TOIS)* **7**(3), 205–229.

Ravikumar, P., Cohen, W. & Fienberg, S. (2004), A secure protocol for computing string distance metrics, *in* 'Workshop on Privacy and Security Aspects of Data Mining held at IEEE ICDM'04', pp. 40–46.

Scannapieco, M., Figotin, I., Bertino, E. & Elmagarmid, A. (2007), Privacy preserving schema and data matching, *in* 'ACM SIGMOD', pp. 653–664.

Schneier, B. (1995), *Applied cryptography: Protocols, algorithms, and source code in C, 2nd Edition*, John Wiley & Sons, Inc., New York.

Song, D., Wagner, D. & Perrig, A. (2000), 'Practical techniques for searches on encrypted data', *Security and Privacy, IEEE Symposium on* p. 44.

Trepetin, S. (2008), 'Privacy-preserving string comparisons in record linkage systems: a review', *Information Security Journal: A Global Perspective* **17**(5), 253–266.

Verykios, V., Karakasidis, A. & Mitrogiannis, V. (2009), 'Privacy preserving record linkage approaches', *Int. J. of Data Mining, Modelling and Management* **1**(2), 206–221.

Winkler, W. E. (2006), Overview of record linkage and current research directions, Technical Report RR2006/02, US Bureau of the Census.

Witten, I. H., Moffat, A. & Bell, T. C. (1999), *Managing Gigabytes, 2nd Edition*, Morgan Kaufmann.

Yakout, M., Atallah, M. & Elmagarmid, A. (2009), Efficient private record linkage, *in* 'IEEE ICDE', pp. 1283–1286.

Bands of Privacy Preserving Objectives: Classification of PPDM Strategies

Rui Li **Denise de Vries** **John Roddick**

School of Computer Science, Engineering and Mathematics
Flinders University,
PO Box 2100, Adelaide, South Australia 5001,
Email: {Rui.Li, Denise.deVries, John.Roddick}@flinders.edu.au

Abstract

At present, data mining algorithms are largely the domain of governments, large organisations and academia where they provide useful insight into the data. However, without the ability to assure privacy protection, the availability of datasets for research purposes may be impaired. Moreover, privacy-preservation is essential if data mining is to be permitted widespread use in government and commercial contexts. Indeed, as data mining algorithms become more widespread, even the datasets currently made available under limited release now may become more restricted. In addition, the ambiguous definitions currently in use hinder the assessment of the quality of the privacy preservation.

This paper categorises the protection objectives during the data mining process into bands and then presents a reconceptualization of privacy-preserving data mining algorithms from the viewpoint of these bands. Existing algorithms from eight protection strategies are selected as examples to explain the six bands. Significantly, gaps are revealed in the Privacy Preserving Data Mining literature that indicate areas for future research.

Keywords: privacy-preserving data mining, bands of privacy preserving objectives, privacy management;

1 Introduction

There is broad acknowledgement that, if abused, the process of knowledge discovery can violate both an individual's privacy and data owner's intellectual property by revealing critical information about the individual or organisation. To counter this a new research direction, Privacy Preserving Data Mining (PPDM) has developed concerned with ensuring that existing and future data mining algorithms assess and manage the risk to privacy. Since the publication of the two major papers defining the PPDM problem (Lindell & Pinkas 2000, Agrawal & Srikant 2000), PPDM has attracted a number of research contributions, including algorithmic development and assessment mechanisms.

This paper is motivated by the observation that, in the main, privacy issues have not been addressed sufficiently from the perspective of the sensitive information to be protected. In particular, consider the following three aspects.

Firstly, assessing the quality of privacy protection needs a declaration of the objects to be protected before applying criteria to quantify the privacy level and hiding failure of a PPDM algorithm. Assessment methods, which identify suitable evaluation criteria and develop related benchmarks, have been widely studied. The assessment criteria proposed by Bertino et al. (2005) encompass five dimensions: *efficiency*, *scalability*, *data quality*, *hiding failure* and *privacy level*. By comparison, Verykios et al. (2004*b*) suggest four factors as evaluation criteria, *performance*, *data utility*, *level of uncertainty* and *resistance*. Factors specifically applied when assessing data mining algorithms with privacy concerns are *hiding failure*, *privacy level* and *endurance of resistance*. However, uncertainty in the privacy protection target restricts the ability to quantify the degree of sensitive information protection. With a clear protection target, we are able to calculate *hiding failure* and *privacy levels* compared with the original protection target.

Secondly, in situations where the data miners are not the data owners, collaborative decisions must be made regarding what sensitive information ought to be protected and which strategies are available to meet the privacy protection requirements. Generally, the sensitivity of information is determined by data owners, whereas data miners select the mining strategies. For better outcomes, there is a need that both data owners and data miners acknowledge each others knowledge domains without violating the privacy agreement.

Ultimately, ambiguous and inexplicit definitions of the protected target hinder the effectiveness of performing PPDM projects, resulting in high computation overheads. Research efforts often opt to sacrifice performance to prevent privacy leakage. For example, utilizing *zero-knowledge proofs* to force a malicious party to follow their protocols. Unfortunately, inherent security problems still cannot be avoided (Han & Ng 2008). Current research upon PPDM acknowledges that no PPDM system guarantees zero privacy leakage. On the other hand, many PPDM strategies are privacy protection tractable, but they show a computationally intractable problem in many realistic cases because of the high computing cost over large volumes of data. Most PPDM strategies assume a protection of overarching information, including mining results and the original datasets. More research is required to balance the quality of privacy preservation with the computational cost caused by privacy

protection processes.

To illustrate this point, consider two banks, A and B, which collaboratively conduct a data mining exercise over their data under a malicious model. The requirements for both parties are that the only information their counterpart is able to learn is the mining result, while their own datasets should be hidden from each other. However, the agreement between two banks leads to three issues.

- Firstly, adversaries still impose a potential loss to Bank A even without knowledge of the whole dataset. Suppose Bank B accidentally queried the financial statements of several VIP customers of Bank A. Bank A will suffer a serious profit and credit loss if the leaked statement information is used for Bank B's marketing strategies competing with Bank A. Therefore, the banks should assign the profiles of VIP customers a high privacy protection. Such a strategy assists the banks to minimize their loss.
- Secondly, due to insufficient knowledge of the protection objectives, the banks normally require their whole datasets to be confidential. However, in most cases, reducing the number of protection objectives also reduces information loss.
- Finally, hiding all datasets from other parties is not a sensible decision. The strategy of protecting more information by sacrificing the performance is not necessary as the greater the privacy protection, the greater the information loss.

Generally PPDM algorithms broadly claim that the goal of the algorithm is to protect the attributes, the whole dataset or any information which lead to an individual. This claim is not clear enough. This ambiguity leads to hiding data which does need to be hidden, resulting in missing potentially interesting mining results. Moreover, in order to calculate the *privacy level* (Bertino et al. 2005) of a PPDM algorithm most assessment strategies leverage information entropy to compute a quantitative value. However, the parameters in these calculations need to be modified if the scenario of the dataset and the protection target of the PPDM strategy are changed. Therefore, clearly identifying the exact information that has to be protected is important.

Research on the classification of the objectives of privacy protection and the usage of various PPDM algorithms is necessary yet lacking in the current literature. This paper aims to redress the insufficiency by presenting a classification of PPDM algorithms that claim to protect privacy during the data mining process.

In Section 2, we define the Bands of Privacy Preserving Objectives (BPPO), in Section 3 we assign PPDM strategies into the Bands of Privacy Preserving Objectives for centralized dataset and in Section 4 for multi-party dataset. We then present our observations in Section 5 followed by our conclusions in Section 6.

2 Defining Bands of Privacy Preserving Objectives

A thorough analysis of privacy requirements reveal the **Bands of Privacy Preserving Objectives** presented in Figure 1. The bands have two groups: 1) data protection for original datasets, including Sensitive Attribute Instances, Sensitive Attributes, Tuples, and Datasets; 2) protection for the information from the mining results, including Intermediate Mining Results and Final Mining Results. This classification of PPDM enables domain experts to more easily select the appropriate data mining algorithm based on the nature of the dataset.

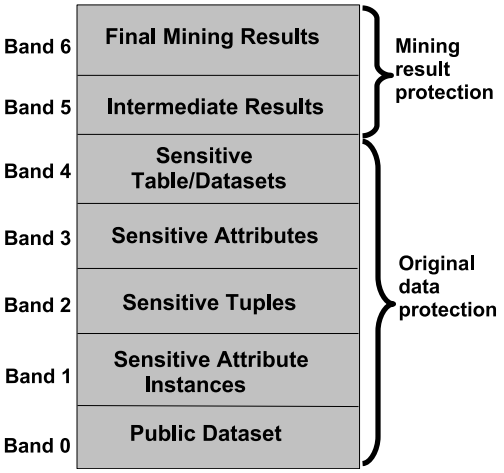


Figure 1: Bands of Privacy Preserving Objectives (BPPO)

The following bands classify the objectives of privacy protection for PPDM algorithms.

Band 0: The Public Dataset

Centralized data mining algorithms which are not concerned with privacy issues fall into this band.

Band 1: Sensitive Attribute Instances Protection

PPDM algorithms should protect the sensitive values of attributes in particular cells. Normal privacy preserving techniques, i.e. modifying data, deleting data, swapping data, can be adopted to sanitize these values. An example case is as follows:

> An institute owns a dataset, consisting of the environmental quality factors in horizontal attributes and the dates within a month in columns. The values above a particular threshold value in a certain attribute column are confidential. Therefore, before performing data mining process over this dataset, these confidential values should be protected through some privacy protection strategy.

Band 2: Sensitive Tuples Protection

The protected objectives in this band are sensitive records or data in row. In circumstances where the number of records and the names of attributes are not deemed as sensitive information by data owners, the PPDM strategies applied on a horizontally partitioned dataset are considerable. Typically, randomization techniques, distributed dataset techniques and mathematical transformation approaches are used to sanitize data in this band. However, geometric transformation strategies do not apply to privacy protection in this band because of the diversity of data types in a record. An example of this is as follows:

A medical company is obliged to stop any trace of individual profiles of the patients who have *lung cancer*. This obligation requires the hiding of all records for these patients when the value in the *disease* attribute is equal to *lung cancer*.

Band 3: Sensitive Attributes Protection

The protection objective in this band is the sensitive column or attribute. In multi-party schemes, when the dataset is vertically partitioned over several data owners, data owners are willing to protect whole sensitive attributes they hold. An example case (Yang & Huang 2007) is:

A commercial bank has its record of customers including attributes for date of birth, income, deposit and credit. It is possible for adversaries to obtain a sort on attributes in descending order. It is also possible for adversaries to acquire valuable information by launching aggregated queries, e.g. SUM, MAX. The leakage risk should be eliminated by protecting the sensitive column. As there is the same data type for the entire column, most traditional geometric transformation methods are good solutions for protecting objectives fitting this band.

Band 4: Sensitive Table or Dataset Protection

In this band the whole table or whole dataset requires protection. As the privacy protection tasks on this band normally involve a distributed dataset, the PPDM algorithms using only modification or transformation techniques do not satisfy the privacy preserving requirements. Thus, from this to Band 6, Secure Multi-party Computation (SMC)-based and query-based PPDM algorithms are introduced. In some cases, to protect the sensitive information in this band, the partition of the dataset could be very complex. For example, the instances for one attribute could be held by several data owners, A, B and C. While the instances for another attribute are held by data owners, D, E and F. Namely, *no* data owner holds *all* instances for *one* attribute and *no* data owner holds an entire record with values for *all* attributes.

An example case is as follows:

A company has a dataset storing 10 years of financial information. For security reason, the dataset is shared between different departments, A, B and C. Each department only holds part of the dataset, so that no department knows all attributes of the dataset nor the total number of the entire records. The company is interested in exploring new knowledge by mining over their entire dataset, but with the constraint of no disclosure of the dataset to other departments and the external data mining company.

The dataset D (Figure 2) is partitioned into four parts, D_A, D_B, D_{C1} and D_{C2}, owned by three data owners A, B and C respectively.

D has n attributes $\{A_1, A_2, \ldots, A_n\}$ with a total of m records $\{R_1, R_2, \ldots, R_m\}$
After partitioning, A gets dataset D_A comprising s records $\{R_1, \ldots, R_{1+s}\}$ of t attributes $\{A_{n-t}, \ldots, A_n\}$; B gets dataset D_B comprising u records $\{R_{m-u}, \ldots, R_m\}$ of n-t attributes$\{A_1, \ldots, A_{n-t}\}$; C gets dataset D_{C1} comprising m-u records $\{R_1, \ldots, R_{m-u}\}$ of n-t attributes$\{A_1, \ldots, A_{n-t}\}$ and dataset D_{C2} comprising m-s records $\{R_{m-s}, \ldots, R_m\}$ of t attributes$\{A_{n-t}, \ldots, A_t\}$. The data mining tasks will be applied over the union of datasets from the three data owners, while each dataset, D_A,D_B and $(D_{C1}+D_{C1})$ must be kept confidential from other two data owners.

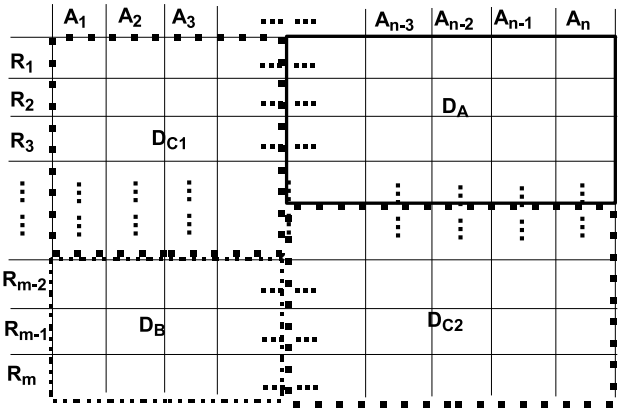


Figure 2: Partition of an United Dataset in Band 4 Protection

There are two options to perform the mining tasks: providing the data mining company with a modified dataset or implementing a SMC protocol. However, as the records of one attribute are owned by more than one department, the attribute protection strategies cannot be adopted in this case. Likewise, the records with all attributes in the united dataset are managed by more than one department. The partition of one record restrains the application of the tuple protection strategies. Hence, cases in this band are more complicated than in bands 2 and 3. The cases in this band are not able to be dealt with by the strategies for horizontally partitioned or vertically partitioned datasets.

Band 5: Intermediate Result Protection

The protection objectives in this band are the sensitive intermediate results, that is, the frequent itemset from association rule mining or the intermediate clustering core points from clustering. We separate this band from the Band 6 as, in some cases, the intermediate results are sufficient for adversaries to trace related private data.

Band 6: Final Mining Result Protection

The aim in this band is to protect the rules or clusters generated from the private datasets. Here the information that should be prevented from disclosure is:

Any clusters or patterns from the sensitive datasets which may lead to sensitive information.

Using mining results with either public or private data inferences may be made to identify individuals.

A widely cited study by Sweeney (2000) and revisited by Golle (2006) showed how using publicly available information (sex, ZIP code and date of birth) with census data allows for unique identification of individuals. Calandrino et al. (2011) use a moderate amount

of auxiliary information about a customer and infer this customer's transactions from temporal changes in the public outputs of a recommender system.

3 Categorising PPDM Strategies for Centralized Datasets

As the majority of privacy preserving strategies in data mining focus on the protection of mining results, including knowledge hiding (Johnsten & Raghavan 2002), rule hiding (Wang & Jafari 2005), cluster hiding (Oliveira & Zaïane 2003*b*), in Table 1, we present a summary of the Bands of Privacy Preserving Objectives and representative algorithms for the PPDM strategies for centralized datasets found in the literature.

3.1 Targeted and Sacrificed Information

In advance, two types of information separately used for bands and strategies should be clarified. The information defined in bands is *Targeted Information* while the information modified by the strategy is *Sacrificed Information*. In PPDM, these different types of information have two kinds of relationships:

- *Targeted Information* and *Sacrificed Information* are the same. For example, the age values of patients, which are *Targeted Information*, are confidential. Thus the strategy of simple modification modifies the age values, which are also *Sacrificed Information*, to protect *Targeted Information*.
- *Targeted Information* and *Sacrificed Information* are different. For instance, values in the original database are modified, which are *Sacrificed Information*, to protect mining results, *Targeted Information*. Another example, a few sensitive rules or mining results are hidden, which are *Sacrificed Information*, to protect a group of values in the original dataset, *Targeted Information*.

3.2 Overview of Algorithms for Centralized Datasets

- *Simple Modification*: This strategy uses removing from or adding to original data items or pattern items to mask sensitive targeted information. Simple Modification of data is a form of cell suppression, a technique applied on statistical databases. Normally, this modification has no rule to obey, they randomly and intuitively modify values. This method, however, has proven privacy leaks (Oliveira & Zaïane 2003*b*). For instance, although the identifiers are removed in the first phase of privacy protection, the released data, after removing identifiers may still contain other information that can be linked with other datasets to re-identify individuals or entities. Therefore, complex hiding strategies have been introduced to protect sensitive information hidden between public data. Nevertheless, because of the simplicity and efficiency in some circumstances, this strategy is still widely used in industrial applications.
- *Blocking*: Instead of providing misleading data or information by using the former strategy, the data blocking strategy provides incomplete information by replacing the sensitive items with an uncertain value, such as "?".
- *Probability Distribution*: This strategy also modifies the sacrificed information to protect the sensitive targeted information as the simple modification strategies do. The difference between these two is that the latter leverages the rules of the probability distribution to make the modification complicated. However, there are two issues caused by this strategy. First, once the rules applied to the sacrificed information are discovered by adversaries, this strategy fails to protect the sensitive targeted information. Second, different data types or a dataset with its own properties may need a different probability distribution strategy.
- *Grouping*: The grouping strategy maps the sacrificed information into a number of groups with a certain degree of privacy. Currently, widely used grouping strategies include K-anonymous (generalization and suppression)(Ciriani et al. 2008), condensation, and ℓ-diversity.
- *Reconstructing Dataset*: Instead of modifying the original dataset to hide sensitive values, the dataset reconstruction strategy creates a new dataset as the shared dataset in terms of sensitive patterns determined by data owner. External data miners conduct general mining algorithms over the shared dataset.
- *Geometric Transformation*: Geometric transformations such as translation, scaling, and rotation are applied to the sacrificed information to protect the targeted information.
- *Sampling Data*: Because of the computation limitation and privacy concerns over the large volume of data, it is applicable that a group of sample data with similar statistics or distribution properties are extracted from the original dataset.
- *Result Hiding and Others*: Recently, privacy protection in the original dataset during the data mining process has explored. Instead of focusing on strategies to protect mining results, some research strategies aim to hide the targeted information in the original dataset by using the mining result. As the sacrificed information, the mining result could be clustering, association rules, classifications and intermediate results, frequent itemset, or core points of the clusters. *Others* may be any novel strategies which are covered by the above-mentioned strategies, such as random response.

3.3 Example Algorithms for Centralized Datasets

3.3.1 Simple Modification of Data

Band 2: Randomization Operators: Evfimievski et al. (2002) note that it is feasible to recover association rules and preserve privacy using a straightforward "uniform" randomization, yet still possible that these discovered rules may be used for privacy breaches. They propose a framework for mining association rules from transactions consisting of categorical items in which the data have been randomized to preserve privacy of individual transac-

Table 1: Privacy Protection Strategies for Centralized Dataset

Strategies	Band1	Band2	Band3	Band4	Band5	Band6
Simple Modification	⎵	Randomization Operators	Data Swapping	⎵	Integer Programming and Border Based	Threshold Filters
Blocking	⎵	⎵	⎵	⎵	Unknowns on Given Items	Unknowns on Support Values
Probability Distribution	Expectation Maximization	Amplification on Itemset Randomization	Simulating Correlation Distribution in Original Data	MASK Mining	Itemset Randomization	⎵
Grouping	⎵	Condensation	ℓ-diversity and Data Utility Preservation	$O(k)$-approximation	K-anonymous Patterns	⎵
Reconstructing Dataset	⎵	⎵	⎵	⎵	⎵	Reconstruction on Frequent Itemset
Geometric Transformation	⎵	⎵	Rotation-based Perturbation	Weighted SVD-based Method	Sanitization Matrix	Geometric Data Transformation Methods
Data Sampling	⎵	⎵	⎵	⎵	Indexing	Limiting Exposure by Sample Size
Result Hiding & Others	Multivariate Data Disguise	⎵	Rule Hiding for Attribute Protection	Meta Privacy Protection ⋄	Rule Hiding for Itemsets	⎵

⋄:the strategy may also fit in other bands.
⎵: no examples were found in the literature

tions. In this approach, some items in each transaction are replaced by perturbed items. To do so, the sensitive real information is removed and the false information is introduced. In their paper, a class of randomization operators is proposed, which is incorporated into mining algorithms, to reduce privacy breaches. Compared with uniform randomization, those formulae comprising the proposed randomization operators are more effective in limiting the breaches.

Band 3: Data Swapping: In order to deal with the trade-off between statistical precision and security level, Estivill-Castro & Brankovic (1999) adopt a data swapping technique to protect the values in confidential attributes and the substantial information about a confidential value without disclosing it exactly. They introduce a method for ensuring partial disclosure while allowing a miner to explore detailed data. They first build a local decision tree over the original data, and then swap values amongst records in a leaf node of the tree to generate randomized training data. The swapping is only performed over the confidential attributes if the confidential attributes are class labels. They claim that this approach balances the statistical precision against the security level by choosing to perform the swapping in the internal nodes, rather than in the leaves of the decision tree: the closer to the root, the higher the security but lower the precision.

Band 5: Integer Programming and Border-Based : An integer programming approach (Menon et al. 2005) and a border-based approach (Sun & Yu 2005) may be taken to hide the frequent items in association rules.

The integer programming approach maximizes the accuracy of the altered database by minimizing the number of transactions to be modified. Additionally, the border-based approach greedily calculates the **weight** of a border item B in each step of the hiding process. Rather than considering each non-sensitive itemset individually, the algorithm aims to preserve the quality of the resulting border.

By combining the advantages of the above two approaches, Gkoulalas-Divanis & Verykios (2006) propose another globally optimal approach.

Although the detailed formulations in these three approaches are delicately designed with a certain degree of complexity, the purpose of the strategies is to delete or remove the sensitive patterns from the original dataset.

Band 6: Threshold Filters: There has been much research in protecting mining results, the *Targeted Information* in association rule mining by using the simple modification strategy.

The basic concept of protecting rules in these example algorithms is to delete or add items to the sensitive transactions in the original dataset in order to decrease the values of *support* and *confidence* to a specified threshold. This basic concept drives the algorithms to design the *threshold filters* by following these five typical steps:

1. perform the mining tasks over the original dataset to obtain the association rules;
2. apply heuristic sensitive rules to select the candidate transactions to be modified. (The features of heuristic PPDM algorithms can be found in the work of Bertino et al. (2005));
3. from these candidate transactions, balance the *disclosure threshold* (Oliveira & Zaïane 2002) and the *conflict degree* (Oliveira & Zaïane 2003*a*) to determine the victim items (Oliveira & Zaïane 2002) to be modified;
4. modify these victim items;
5. repeat steps 1 to 4 until the the values of *support* and *confidence* are down to the user-defined safe values.

By doing so, sensitive rules are not able to appear in the result of mining the shared datasets.

Details of the algorithms applying *conflict degree* and *disclosure threshold* associated with this category are be found elsewhere (Dasseni et al. 2001, Verykios et

al. 2004*a*, Oliveira & Zaïane 2002, 2003*a,c*). Example algorithms include Naive Algorithm, Round Robin Algorithm (RRA) and Sliding Window Algorithm (SWA).

3.3.2 Blocking

Band 5: Unknowns on Given Items: To hide specific rules, many data altering techniques need to execute the entire data mining process. However, for some applications, only certain sensitive predictive rules that contain given items have to be hidden. Therefore, Wang & Jafari (2005) assume that only sensitive items are given and propose two algorithms, **I**ncrease **S**upport of **L**eft Hand Side (ISL) and **De**crease **S**upport of **R**ight Hand Side (DSR), to replace data by unknowns (specifically, question marks "?" to represent the unknowns) in the database. Thus, they expect to reach the goal that sensitive predictive rules containing specified items on the left hand side of rule cannot be inferred through association rule mining.

Band 6: Unknowns on Support Values: By the introduction of *Unknowns*, the data blocking strategy turns the single value for the support of an itemset A into a *support interval*, and the *confidence* of the rule into a *confidence interval*.

When there are no unknown values (e.g. "?"), then minima and maxima for the *support* and *confidence* are correspondingly identical. During the sanitization process of placing the *Unknowns*, the minima and maxima will start to set apart, and in this way, the degree of uncertainty for the rule, will increase. When the *minsup(A)* is less than Minimum Support Threshold (MST), or the *minconf(*$A \longrightarrow B$*)* is less than Minimum Confidence Threshold (MCT), the rules are considered to be hidden successfully.

When applying a data blocking strategy, three approaches are universally adopted to hide the sensitive rules(Saygin et al. 2001):

1. decrease the value of $minsup(A)$,
2. decrease the value of $minconf(A \longrightarrow B)$,
3. increase the value of $minconf(A \longrightarrow B)$.

Detailed algorithms are in (Saygin et al. 2001, Hintoglu et al. 2005). Also Bertino et al. (2005) present a rule hiding data mining algorithm by blocking sensitive data to test their evaluation methods for PPDM algorithms.

3.3.3 Probability Distribution

Band 1: Expectation Maximization: Agrawal & Srikant (2000) consider that the target of data mining is to discover useful patterns in aggregated data, therefore, it is possible to find patterns without obtaining every specific value from the dataset. With this in mind, hiding the *Targeted Information* in Band 1 has become an important research direction in PPDM. Example strategies, applying the Expectation Maximum (EM), are as follows:

> Agrawal & Srikant (2000) build a decision-tree classifier from training data in which the values of individual records have been perturbed, by adding random values from a probability distribution. By applying Bayes rules on the perturbation dataset and the distribution of noisy data, a distribution of the original dataset is constructed. The resulting data records look very different from the original records and the distribution of data values is also very different from the original distribution. While it is not possible to accurately estimate original values in individual data records, they propose a novel reconstruction procedure to accurately estimate the distribution of original data values. The distribution reconstruction process leads to some loss of information, but the authors argue that this is acceptable in many practical situations.

An algorithm (Agrawal & Aggarwal 2001) is presented for distribution reconstruction which is more effective than that published earlier (Agrawal & Srikant 2000) from the perspective of information loss. This algorithm converges to the maximum likelihood estimate of the original distribution based on the perturbed data, even when a large amount of data is available. They note that the EM algorithm is in fact identical to the Bayesian reconstruction proposed in (Agrawal & Srikant 2000), except for the approximation partitioning values into intervals.

Band 2: Amplification on Itemset Randomization: Evfimievski et al. (2002, 2003) use a randomization distribution strategy in the scenario of multiple data owners and one mining service server. The aim is to pull the randomized dataset from data owners to a centralized dataset, on which the mining service provider implements the mining task. Although this is a multi-party scenario, these papers provide suitable examples for centralized data mining while protecting privacy.

Each client protects privacy of their numerical and categorical data by perturbing it with a randomization algorithm and then submitting the randomized version to the server (Evfimievski 2002). In their following papers, this strategy was applied to randomize the itemset during association rule mining.

Meanwhile, an "amplification" methodology is proposed to limit the potential privacy breaches inherent in the randomization approach (Evfimievski et al. 2003).

Band 3: Simulating Correlation Distribution in Original Data: Kargupta et al. (2003) submit that randomization might not be able to properly preserve privacy. Further, Huang et al. (2005) state that the key factor about this issue is the correlation between attributes. As proof, they propose two data reconstruction methods based on data correlations: the Principal Component Analysis (PCA) technique and the Bayes Estimate (BE) technique. Their theoretical and experimental analysis proved that both these techniques can reconstruct more accurate data when the correlation of data increases.

To deal with this issue, they propose a modified random perturbation schema by adding random noises. The correlation of the added noise data is similar to that of the original data. Experimental results show that the higher the similarity between the two types of correlation is made, the less accurate reconstructed data are obtained.

Band 4: MASK Mining: Mining Associations with Secrecy Konstraints (MASK) (Rizvi & Haritsa 2002) is based on probabilistic distortion of user's data, employing random numbers generated by a predefined distribution function, which is composed of a privacy metric and an analytical formula. The data owner modifies data values for any individual transaction, yet the rules learned on the distorted data are still valid. Although this framework provides a high degree of privacy to the user and retains a high level of accuracy in the mining results, mining the distorted database can be, apart from being error-prone, significantly more expensive in terms of both time and space when compared to mining the original database.

Band 5: Itemset Randomization: The randomization method can be applied for privacy protection on categorical data (Evfimievski 2002). As the composition of an itemset, in the context of association rule mining, is mostly categorical data, Evfimievski et al. (2002) extended the categorical data protection to itemset protection by using their new randomization method.

3.3.4 Grouping Data

Band 2: Condensation: A methodology which condenses the data into multiple groups of predefined size is proposed by Aggarwal & Yu (2004). This condensation-based approach may be used to protect the values in individual records as a number of records is aggregated into a condensed group. Each group maintains three types of statistical information about its records: 1) the sum of corresponding values upon each attribute; 2) the sum of the product of corresponding attributes' values over each pair of attributes; 3) the total number of records in each group.

The condensation technique is able to preserve the inter-attribute correlations of the data and does not require modification of existing data mining algorithms.

Band 3: ℓ-diversity and Data Utility Preservation: Machanavajjhala et al. (2007) present two simple attacks in which a k-anonymized dataset has some subtle, but severe privacy problems. One is that an attacker can discover the values of sensitive attributes when there is little diversity in those sensitive attributes. The other is that k-anonymity does not guarantee privacy against attackers using background knowledge. These two findings lead to introducing *ℓ-diversity* to further protect sensitive attributes.

Poovammal & Ponnavaikko (2010) note that *ℓ-diversity* focuses on a universal approach that exerts the same amount of privacy preservation for all persons against a linking attack, which results in a high loss of information. Hence, privacy is not 100% guaranteed because of proximity and divergence attacks. They present a micro data sanitization technique which applies a graded grouping transformation on numerical sensitive attributes and a mapping table-based transformation on categorical sensitive attributes.

Band 4: $O(k)$-approximation: In the study of the k-anonymity problem (Aggarwal et al. 2005) it is shown that it is NP-hard even for the special case of ternary attribute values. This can be reduced to $O(k)$ through approximation in which two algorithms are used: 1) Create a forest G with cost at most optimal n-Anonymity solution (OPT). The number of vertices in each tree is at least k; 2). Compute a decomposition of this forest such that each component has between k and $3k-3$ vertices. The decomposition is done in a way that does not increase the sum of the costs of the edges.

Band 5: K-anonymous Patterns: The *K-anonymous Patterns* method (Atzori et al. 2005) shifts the concept of *k-anonymity* from data to patterns and highlights the privacy problem in the general setting of patterns which are Boolean formulas over a binary database. In the case of association rule mining, patterns are itemsets, which are the intermediate results during the mining process. Atzori et al. define k-anonymous patterns and present characterisations of potential inference channels holding among patterns that may threaten anonymity of source data.

3.3.5 Reconstructing Dataset

Band 6: Reconstruction of Frequent Itemset: Chen et al. (2004) introduced the idea of reconstructing a published dataset for association rules hiding, with the 3 steps:

1. Mining frequent itemsets;
2. Sanitizing patterns;
3. Reconstructing shared dataset

Step 1 can be implemented by conventional mining task, which is not the designing target of reconstruction strategy. Conventionally, sensitive pattern sanitizing methods are data-oriented. The patterns to be hidden are subject to whether the associated data are sensitive. This heuristic method leads to the problem that the hiding result can only be observed at the end of the mining process. Therefore, in Chen etc's paper, they focused on presenting a new sanitizing patterns method, which is mining result oriented. Unlike the conventional privacy protection method, from original data to pattern hiding, this method uses the inverse way, from the mining result to pattern hiding. Other work contributing to step 3, *reconstructing sharing dataset*, includes constructing a dataset horizontally (Calders 2004) and vertically (Chen & Orlowska 2005).

As this strategy is based on frequent itemsets to construct a new sharing dataset, the aim of which is to protect rules, there are few papers to distinguish the privacy preserving objects between rules and frequent itemsets. Given the success of constructing a dataset, sensitive rules and frequent itemsets are protected at the same time.

The application of reconstruction strategy on hiding classification rules is presented by Natwichai et al. (2005, 2006). Similar to the steps in reconstruction during association rule mining, there are also 3 steps for classification mining:

1. Classification Mining;
2. Decision Tree Construction;
3. Dataset Reconstruction

These papers also present a rule-based decision tree construction algorithm (RDTCA) for step 2 and a Decision Tree Building algorithm for step 3.

3.3.6 Geometric Transformation

Motivated by research in image processing, Oliveira & Zaïane (2003*b*) developed PPDM strategies utilising geometric data transformation.

Band 3: Rotation-based Perturbation: To improve privacy quality without sacrificing accuracy, Chen & Liu (2005) propose a rotation-based perturbation method to protect multi-column privacy in classifiers. Through exploiting the properties of Geometric Rotation over a matrix, it is shown that several existing classifiers are *rotation-invariant*, which can be used in privacy preserving classification by applying proper geometric rotation methods. *Multiple column privacy metrics* are employed to find the locally optimal rotation.

Band 4: Weighted SVD-based Method: The singular value decomposition (SVD) method has been widely adopted to preserve privacy in data mining. However, the weakness of traditional SVD-based method was discovered by Li & Wang (2011). One vulnerability is that, in classification, different samples are treated equally even they do not hold the same importance for data mining, prompting the development of a weighted SVD-based method. In this method, each sample has a weight, and different samples will be treated with different weights.

Band 5: Sanitization Matrix: For the purpose of hiding sensitive patterns during mining association rules, an approach of multiplying the original transaction dataset, D by a sanitization matrix S, is proposed by Lee et al. (2004). The most significant step in this approach is the construction of the sanitization dataset S. Initially, S is set as an identity matrix, S_{ij} is 1 if i=j, otherwise, S_{ij} is 0. Then the setting of the entry on the non-main diagonal, S_{mn}, in S follows the principles:

1. **Setting of -1**: Setting entry in S to -1 to decrease the correlation between items m and n. The support of a sensitive pattern m, n can be decreased by reducing the correlation between items m and n in D.
2. **Setting of 1**: Setting entry in S to 1 to maintain the correlation between items m and n, in order to minimize the effect on losing non-sensitive patterns.

For the different purposes of hiding items, they propose three algorithms to set the sanitization matrix, S.

1. Hidden-first (HF) No consideration of the error-hiding of the non-sensitive pattern, the algorithm focuses on eliminating all sensitive patterns in D by setting proper entries in S to 1.
2. Non-Hidden-first (NHF) With the prerequisite of avoiding the error-hiding of non-sensitive patterns, the algorithm considers the hiding of sensitive patterns.
3. Hiding sensitive patterns completely with minimum side effect on non-sensitive patterns (HPCME) As HF may accidentally hide some non-sensitive patterns and NHF may not successfully hide all the sensitive patterns, HPCME is proposed to combine the advantages of both HF and NHF but attempts to eliminate their disadvantages.

An improvement of this Sanitization Matrix approach has been made by Wang et al. (2005) to avoid the Forward-Inference Attack (Oliveira et al. 2004).

Band 6: Geometric Data Transformation Methods: A family of geometric data transformation methods (GDTMs) is introduced by Oliveira & Zaïane (2003*b*) to preserve the main features of clusters mined from the original database. These geometric transformation methods include translation, scaling and rotation. In order to ensure that the mining process will not violate privacy up to a certain degree of security, their proposed methods distort only confidential numerical attributes.

3.3.7 Data Sampling

Band 5: Indexing: A framework for enforcing privacy in mining frequent patterns is presented by Oliveira & Zaïane (2002). It combines three advances: inverted files, a transaction retrieval engine and a set of sanitizing algorithms. The retrieval engine retrieves transaction IDs from the inverted files and then feeds the sanitizing algorithms with a set of sensitive transactions to be sanitized. During construction of the inverted files, the original transactions are reconstructed by a series of indexing strategies.

Band 6: Limiting Exposure by Sample Size: Most PPDM algorithms are based upon the hypothesis that the mining results, e.g. rules, clusters, to be protected against are known. However, Clifton (1999, 2000) presents another scenario in which the sensitive mining results are uncertain currently for data custodians. For example, a developer for a new software system needs sample data for testing. However, the data custodian worries about the sensitive rules in the data to be discovered but without knowing which ones should be hidden. Under this scenario, conventional privacy preserving mining methods cannot be applied due to lack of knowledge of what should be protected. His research shows the relationship between the size of sample data and the error of the classifiers learned from the sample.

3.3.8 Rule Hiding & Others

Band 1: Multivariate Data Disguise: This is a mixed strategy, which combines the randomized response and uniform distribution and other pruning strategies to protect privacy (Du & Zhan 2003). The method contains two contributions: the multivariate data disguising technique and the modified ID3 decision tree building algorithm. While building the decision tree, all records in the original dataset are put into the disguised dataset G based on the Multivariate Randomized Response (MRR), which protects the values of attributes to a certain degree.

Band 3: Rule Hiding for Attributes Protection: Noting that malicious inference may lead to the disclosure of secret attributes, Chang & Moskowitz (2001) undertake to protect secret attribute data by analysing and hiding the rules. This paper predates most work dealing with PPDM and

the rules discussed in their paper are actually aggregated data generated from databases, nonetheless the method by using probabilistic impact is classic.

Band 4: Meta Privacy Protection: A conceptual framework from a different perspective of PPDM is proposed by Skinner et al. (2005), where the *Target Information* is the privacy in metadata, i.e. Meta Privacy. Due to the recency of this field, there has not been much research activity. Although this paper has recommended some technologies for distributed meta privacy mining and other privacy objects to protect meta privacy, it is difficult to categorise it precisely into one of our bands. However, as new perspectives always have potential research possibilities, we tentatively classify it as sensitive table/dataset protection for both centralized and multi-party datasets.

Band 5: Rule Hiding for Itemsets: Given specific rules to be hidden, many data altering techniques for hiding association, classification and clustering rules have been proposed. However, to specify hidden rules, the entire data mining process needs to be executed. For some applications, only certain sensitive items have to be hidden. In the work of Wang et al. (2004), it is assumed that only sensitive items are given and two algorithms are proposed to modify data in a database so that sensitive items cannot be inferred through association rules mining algorithms .

SENSITEM (Duraiswamy 2008), is an algorithm to identify sensitive items based on the discovered rules. Rules are sorted and the frequency of the consequent items are considered and compared with the threshold value. The selected consequent items are finally considered as sensitive items. In this, the confidence values of all rules with the threshold value are compared.

4 Categorising PPDM Strategies for Multi-party Datasets

Table 2 summarises the strategies under the scenario of multiple data owners, also called multi-party.

In (Bertino et al. 2005), all distributed database algorithms are classed under cryptographic-based PPDM algorithms. Since then there have been a number of new approaches developed for PPDM for multi-party datasets which use non-cryptographic algorithms. We categorise privacy preserving algorithms for multi-party datasets into two main groups: cryptographic and non-cryptographic. In the cryptographic group, we distinguish algorithms, the mining tasks of which are performed with other data owners, from those where the tasks need the participation of the parties outside the data owners. However, for non-cryptographic algorithms, we classify them under their mining approaches (e.g. Rule Mining). The reasons for this classification are 1) Different mining tasks, such as Rule Mining and Clustering, leverage different properties of a dataset. Thus, the information that is sensitive to rule mining may not be to clustering. Certainly, each of the examples in this category may utilize a different non-cryptographic approach to protecting sensitive data; 2) none of this research is mature enough to have proved Third or Non-Third party privacy protection.

4.1 Cryptographical SMC Protocols

Secure Multi-party Computation Protocols (SMC) were first researched for two parties by Lindell & Pinkas (2002) when they applied SMC technology to ID3 to mine classifications over vertically partitioned datasets. Since then, SMC protocols have been researched to further protect sensitive data in two party and three party mining activities.

4.1.1 Data Owner Computation

Band 2: Scalar Product Protocol: Scalar product is widely adopted by privacy preserving protocols in the scenario of mining multi-party datasets. These protocols are originally generated from Yao's Millionaire secure computation (Yao 1982), the targeted information of which are the records in each site (horizontal partitioning). Note that as the application of each scalar product computation requires that the datatypes in each site are identical, in some cases, the targeted information of each scalar product computation is column information (Band 3). Nevertheless, scalar product protocol is valid for a privacy protection strategy over tuples where each party owns a number of records in the whole dataset.

Goethals et al. (2005) define the private scalar product protocol as:

A protocol between Alice and Bob a *scalar product* (SP) protocol is when Bob obtains, on Alice's private input $x = (x_1, \ldots, x_N) \in \mathbb{Z}_m^N$ and on Bob's private input $y = (y_1, \ldots, y_N) \in \mathbb{Z}_m^N$, the scalar product $x.y = \sum_{i=1}^{N} x_i y_i$. A protocol is a *shared scalar product (SSP) protocol* when Alice receives a uniformly distributed random value $s_A \in \mathbb{Z}_m$ and Bob receives a dependent uniformly distributed random value $s_B \in \mathbb{Z}_m$, such that $s_A + s_B \equiv x.y(mod\ m)$. A scalar product protocol is *private* when after executing the protocol, Bob obtains no more knowledge than $x.y$ and Alice obtains no new knowledge at all. In particular, Alice gets to know nothing new about Bob's vector and Bob gets to know nothing about Alice's vector that is not implied by x and $x.y$. A *private shared scalar product protocol* is defined analogously.

The Add-vectors Permuted-outputs Computation (Amirbekyan & Estivill-Castro 2007), based on the Add Vector Protocol, assumes a horizontal partition of datasets. Each party sending a dataset adds the vector which is first permutated by a function generated by the receiver.

1. Alice and Bob apply the ADD VECTORS PROTOCOL for Alice to obtain $\pi_0(\vec{a} - \vec{b})$ where 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 $\sum_{i=1}^{n} b_i^2$
3. Now Bob can send $\sum_{i=1}^{n} b_i^2$ to Alice to compute the scalar product, that is

Table 2: Privacy Protection Strategies for Multiparty Dataset

		Band 1	Band 2	Band 3	Band 4	Band 5	Band 6
Cryptographical SMC Protocol	Data Owner Computation	⎵	Scalar Product Protocol	Conversion to Malicious Party	⎵	Recluster Protocol	⎵
	Outsourced Mining Computation	⎵	Resampling & Dissimilarity Matrix	Homomorphic Encryption and Reduced Kernel	Aggregated Queries	⎵	⎵
Non-Cryptographical Strategies for Mining Algorithms	Rule Mining	⎵	⎵	⎵	⎵	⎵	⎵
	Clustering	⎵	Game Theory Punishment	⎵	⎵	⎵	⎵
	Classification	⎵	Gini Indexing	⎵	⎵	⎵	⎵
	Others	⎵	Perturbation on Distributed Schema	⎵	Meta Privacy Protection ⋄	Random Projection	⎵

⋄:the strategy may also fit in other bands.
⎵: no examples were found in the literature

$$2\ \vec{a}^T\ .\ \vec{b} = 2\sum_{i=1}^{n} a_i b_i$$
$$= \sum_{i=1}^{n} a_i^2 + \sum_{i=1}^{n} b_i^2 - \sum_{\pi_0(i)=1}^{n} (a_{\pi_0(i)} - b_{\pi_0(i)})^2$$

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.

Band 3: Conversion to Malicious Party: Han & Ng (2008) investigated the issues of security violations for PPDM in the case of a malicious party providing false data. They discovered four possible privacy vulnerabilities of secure scalar product protocols and proposed a general model to fix these vulnerabilities. According to the examples to illustrate the applicability of their proposed model, they showed two efficient approaches to securely split $(x_1 + y_1)(x_2 + y_2)$ and $(x + y)log_2(x + y)$ respectively. Therefore, this method protects sensitive column, in Band 3, in terms of the two examples proposed because only the same type of attributes can be applied to the functions. For example, if x is numeric, then y has to be numeric.

Shah & Zhong (2007) present two methods to convert PPDM solutions from the semi-honest model into a malicious model. Although they do not address the distribution type of the dataset, we again look to the mathmetical nature of the computation components in the protocols to classify this approach in this band.

Yang & Huang (2007) adopt attribute-wise transformation to mine clusters under the privacy preservation requirement among multiple computation parties.They state that their solutions work on both semi-honest and malicious models by using a Randomized Diagonal (RD) matrix to protect the sensitive attributes defined by the data owner over a vertically partitioned dataset. The RD matrix leverages orthogonal transformation in the semi-honest condition to avoid the compromise between privacy and accuracy, and also protects data from malicious complicity by randomization without losing much accuracy.

Band 5: ReCluster Protocol: To efficiently and privately apply k-means clustering over a horizontally partitioned dataset, Jagannathan et al. (2010) present a distributed PPDM protocol, named *Private ReCluster Protocol*.

This protocol has been designed based on the scenario of Alice and Bob from Yao's Millionaire protocol (Yao 1982). However, in order to make the protocol communication efficient, they propose two other sub-protocols under their *ReCluster Protocol*, *Permute-share Protocol* and *Merge-clusters Protocol*. The first sub-protocol is designed for securely permuting and sharing data between two parties. The latter merges clusters generated from different parties. As no non-data-owner party is involved in this protocol, the mining task is completed by the data owners. The participants of the protocol learn only the final cluster centres on completion of the protocol, thus no intermediate candidate cluster centre is revealed.

4.1.2 Outsourced Mining Computation

The semi-trusted party in cryptographical SMC protocols may be the mining service provider, the party who holds the intermediate result during secure computation.

Band 2: Resampling and Dissimilarity Matrix: To apply the classification mining task over horizontally distributed dataset with privacy preserving needs from data custodians, a non-distance-preserving approach by using Kernel Distance Estimation (KDE) Resampling is proposed by Tan & Ng (2007). The private data at each site are randomized independently by this resampling method before being pooled to the mining service provider.

KDE Resampling provides consistent density estimates with randomized samples that are asymptotically independent of the original samples. This algorithm assumed that there are L (for $2 \leq L \leq N$) distributed data sites (private) and 1 centralized (untrusted) server, which is an untrusted data mining service provider.

In multi-party data mining, each data type requires different comparison functions and corresponding protocols. Therefore, Inan et al. (2007) propose a clustering PPDM over horizontally partitioned by constructing the dissimilarity matrix of objects from different sites, specially designed for categorical, numerical and alphanumerical attributes. Their protocol is motivated by the following scenario:

There are k Data Owners (DO)($k \geq 2$). Each of DO owns a horizontal partition of the data matrix D. These parties want to cluster their data by means of a third party, not a DO, but serves as a means of computation power and storage space. The Third party's

duty in the protocol is to govern the communication between DOs, construct the dissimilarity matrix and publish clustering results to DOs.

Every DO and the third party must have access to the comparison functions so that they can compute distance/dissimilarity between objects. This attribute list is also shared with the third party so that it can run appropriate comparison functions for different data types.

At the end of the protocol, the third party will have constructed the dissimilarity matrices for each attribute separately. These are weighted using a weight vector sent by the DOs. The third party then runs a hierarchical clustering algorithm on the final dissimilarity matrix and publishes the results. Every DO can impose a different weight vector and clustering algorithm of its own choice.

Band 3: Homomorphic Encryption and Reduced Kernel: On a vertically partitioned dataset, the columns are partitioned into different sites. Vaidya & Clifton (2003) present a method for k-means clustering when different sites contain different attributes for a common set of entities. Each site learns the cluster of each entity, but learns nothing about the attributes at other sites. To find the closest clusters in each site, this k-means privacy preserving algorithm makes use of Homomorphic Encryption knowledge to compute the permutation.

Mangasarian et al. (2008) provide a Support Vector Machine (SVM) classifier to protect different columns belonging to different data owners. Their classifier is based on a reduced kernel $K(A, B^{'})$ where $B^{'}$ is the transpose of a random matrix B. The column blocks of B corresponding to the different entities are privately generated by each entity and not made public.

Band 4: Aggregated Queries: Thompson et al. (2009) propose protocols to avoid Service Providers (SP) accessing the individual data entries from the intermediate results. Their solutions for SUM-related aggregate queries are based on a homomorphic commitment scheme and leverage its linearity property. The solutions for MIN and MAX queries are based on the proof of knowledge of a greater-than relation of two values. In their protocols, queries protect the datasets owned by different data owners. In this scheme, a semi-honest service provider carries out the mining tasks.

4.2 Non-Cryptographic Strategies

4.2.1 Clustering Mining

Band 2: Game Theory Punishment: Protocols used for secure computation in PPDM in a multi-party environment often assume no collusion between parties. Kargupta et al. (2007) argue that this assumption often falls apart in real life applications of PPDM. Furthermore, they emphasise that if nobody is penalised for cheating, rational participants tend to behave dishonestly. Therefore, they propose a game-theoretic framework by applying punishment principles to avoid collusion between multiple parties. The secure sum computation in PPDM is taken as the example of implementing a penalty mechanism. The Secure Sum with Penalty (SSP) protocol penalises the colluding party by increasing their cost of computation and communication.

4.2.2 Classification Mining

Band 2: Gini Indexing: Ma & Deng (2008) present protocols for both vertically and horizontally partitioned data using the Gini index with the ID3 algorithm for decision trees, instead of the more widely researched entropy approach (Lindell & Pinkas 2000, Du & Zhan 2003, Vaidya & Clifton 2005, Xiao et al. 2005).

4.2.3 Others

Band 2: Perturbation on Distributed Schema: Because cryptography-based secure multi-party computation on large scale distributed datasets in PPDM has poor performance, Li et al. (2009) use the more efficient data perturbation technique in a multi-party scheme. They propose a light-weight anonymous data perturbation method with three aspects: privacy and integrity, consistency, and robustness. They propose two distributed privacy preserving data perturbation protocols: adaptive privacy preserving summary protocol to address the *consistency constraint* and the anonymous exchange protocol to address *privacy and integrity*. A Distributed Anonymous Data Perturbation framework (DADP) addresses the *robustness* constraint.

Band 4: Meta Privacy Protection: Metadata can contain many personal details about an entity. It is subject to the same risks and malicious actions to which personal data are exposed. Skinner et al. (2005) propose a conceptual framework termed *Meta Privacy Protection* concerned with the protection and privacy of information system metadata and metastructure details. This paper recommends technologies for distributed *Meta Privacy Protection* mining and analyses some factors influencing *Meta Privacy*.

Band 5: Random Projection: The random projection-based technique (Liu et al. 2006) transforms the data while preserving its statistical characteristics (intermediate mining result). According to the experimental results, the proposed technique can be successfully applied to different kinds of data mining tasks, including Euclidean distance estimation, correlation matrix computation, clustering, outlier detection and linear classification.

5 Observations and Discussions

Initially privacy preserving data mining research had as its objective the protection of the results of the mining process e.g. association rules, clusters, classification. There are some studies on protecting the values in original datasets, however, protection methods for sensitive attribute instances in Band 1 and sensitive table/dataset in Band 4 have not attracted enough research.

Observation 1

The privacy preserving strategies which are designed to protect records or attributes assume that attribute instances are also automatically preserved by the strategies. Consequently, few algorithms are specially designed for protecting attribute instances, the targeted information in Band 1.

Attribute instances should be protected by specially designed algorithms for two reasons. Firstly, the PPDM algorithms require spatial and temporal efficiency. For example, there are n sensitive values distributed into s attributes in the dataset ($n \geq s > 0$). Each attribute has m instances. The strategy to protect the targeted information in Band 3 is adopted to hide the n sensitive values. Suppose modifying all values in one attribute takes t seconds, ergo at least $s \times t$ seconds are needed to protect the n sensitive values and at least $s \times m$ instances have to be modified. In fact, only n sensitive instances need to be hidden. Particularly when the number of instances in every attribute is large, it will cause $(s \times m) \gg n$, which is a waste of time and space. Secondly, non-specially designed instance-hiding strategies may not protect values successfully, such as when an attacker utilizes aggregated queries (e.g. SUM, MAX, MIN) to get sensitive values in the dataset with the probability-applied attribute-hiding strategy (Nabar et al. 2006).

Observation 2

Few algorithms are designed to specifically protect the sensitive table/dataset, the targeted information in Band 4.

The reasons could be 1) the complicated data distribution and hybrid data types stored by data owners restrict the direct application of most existing hiding strategies, e.g. conventional geometric transformation method and probability method. 2) The properties of this complexity have not been studied sufficiently, which results in the limited research contribution to preserve privacy in Band 4. The complexity may have many causes, including, data owners who hold data with different data types, or, the whole dataset is randomly shuffled and each data owner holds a fraction of the shuffled data. By doing so, the records and the attributes in the whole dataset are distributed.

For example, patient A's dentist keeps dental information. A's General Practitioner (GP) keeps general health information and the insurance company holds health insurance information. Because both the GP and the insurance company optionally ask for patients' medical history, patient A chooses to leave his/her medical history with the GP. The whole medical record for patient A has been split into different sections, which are held by different data owners. However, another patient B supplies his/her medical history information to the insurance company. If the number of patients is huge, no data owner, GP, dentist or insurance company, could have all the instances for one attribute.

Observation 3

Centralized-data protection strategies concentrate more on protecting the mining results than the original data. In contrast, original data protection and privacy leakage issues on original data has received more research interest than mining results protection in multi-party contexts.

Observation 4

Research advances are more focused on studying complex information-hiding strategies, especially mathematics-based strategies. On the other hand, the simple modification strategy has been neglected, which could become useful and practical in many cases. The advantage of this strategy is there is *no rule*, which is naturally and completely random. In some circumstances, a *no rule* strategy protects more privacy because it is hard for attackers to find a formula to break through the sensitive information. Intuitively, it is true and needs further research. Additionally, simple modification is very useful to protect targeted information in Band 1. A prospective research direction could be the application of *simplicity*: simplicity in terms of both *software engineering* and *usability*.

Observation 5

In multi-party mining, most solutions are based on Yao's Millionaire protocol (Yao 1982), using cryptographical SMC technology. The efficiency and computation costs have been identified as the major issues of this technology, with much research effort spent to reduce the impact of the two issues. However, the application of this technology to real-world cases is still difficult. In this survey, we discovered that the contributions to using non-cryptographical strategies are rare, especially for association rule mining. We propose this as an open research area, *Non-Crytographical Multiparty Mining Strategy*. An example of this direction is *Game Theory Punishment* strategy (Kargupta et al. 2007). To protect privacy, instead of only guarding/hiding the sensitive information, the punishment policy frightens the attackers, especially the semi-honest or the curious parties. The self-constraint of their behaviour limits the disclosure of sensitive data to a great degree.

6 Conclusion

Typically privacy preserving data mining approaches are categorised by the strategy they take or the particular type of data mining to which they apply. The ambiguous definition of the *protected objectives* of a privacy preserving strategy reveals issues from three aspects: the assessment criteria of PPDM algorithms, the data owners' demands and the balance between the computation cost and privacy leakage.

This paper presents a six-bands of privacy preserving objectives for a reconceptualization of privacy-preserving data mining algorithms. As an initial work, selected existing privacy preserving algorithms demonstrate the possible use of the Bands before it is presented by this paper. Significantly, many gaps are revealed in the Privacy Preserving Data Mining (PPDM) literature that indicate areas for future research. Meanwhile, a further survey of the existing PPDM algorithms should be done to deliver more insightful gaps for further research and the applications of the six-bands solution on the three issues raised in the Introduction will become our future work.

References

Aggarwal, C. C. & Yu, P. S. (2004), A condensation approach to privacy preserving data mining, *in* 'Extending Database Technology(EDBT)', pp. 183–199.

Aggarwal, G., Feder, T., Kenthapadi, K., Motwani, R., Panigrahy, R., Thomas, D. & Zhu, A. (2005), Anonymizing tables, *in* T. Eiter & L. Libkin, eds, 'Database Theory - ICDT', Vol. 3363 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 246–258.

Agrawal, D. & Aggarwal, C. C. (2001), On the design and quantification of privacy preserving data mining algorithms, *in* '20th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems', PODS '01, ACM, New York, USA, pp. 247–255.

Agrawal, R. & Srikant, R. (2000), 'Privacy-preserving data mining', *SIGMOD Rec.* **29**, 439–450.

Amirbekyan, A. & Estivill-Castro, V. (2007), A new efficient privacy-preserving scalar product protocol, *in* '6th Australasian Conf. on Data Mining and Analytics', Vol. 70 of *AusDM '07*, Australian Computer Society, Inc., Darlinghurst, Australia, pp. 209–214.

Atzori, M., Bonchi, F., Giannotti, F. & Pedreschi, D. (2005), K-anonymous patterns, *in* '9th European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD)', pp. 10–21.

Bertino, E., Fovino, I. & Provenza, L. (2005), 'A framework for evaluating privacy preserving data mining algorithms', *Data Mining and Knowledge Discovery* **11**, 121–154.

Calandrino, J. A., Kilzer, A., Narayanan, A., Felten, E. W. & Shmatikov, V. (2011), "you might also like:" privacy risks of collaborative filtering, *in* 'IEEE Symposium on Security and Privacy(SP)', pp. 231–246.

Calders, T. (2004), Computational complexity of itemset frequency satisfiability, *in* '23rd ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems', PODS '04, ACM, New York, NY, USA, pp. 143–154.

Chang, L. & Moskowitz, I. S. (2001), An integrated framework for database privacy protection, *in* 'IFIP TC11/ WG11.3 14th Annual Working Conf. on Database Security: Data and Application Security, Development and Directions', Kluwer, B.V., Deventer, The Netherlands, pp. 161–172.

Chen, K. & Liu, L. (2005), Privacy preserving data classification with rotation perturbation, *in* '5th IEEE Intl. Conf. on Data Mining', ICDM '05, IEEE Computer Society, Washington, DC, USA, pp. 589–592.

Chen, X. & Orlowska, M. E. (2005), A further study on inverse frequent set mining, *in* '1st Intl. Conf. on Advanced Data Mining and Applications', Springer, Berlin Heidelberg, Germany, pp. 753–760.

Chen, X., Orlowska, M. & Li, X. (2004), A new framework of privacy preserving data sharing, *in* 'IEEE ICDM Workshop on Privacy and Security Aspects of Data Mining', IEEE Computer Society, pp. 47–56.

Ciriani, V., Vimercati, S. D. C., Foresti, S. & Samarati, P. (2008), k-anonymous data mining: A survey, *in* C. C. Aggarwal & P. S. Yu, eds, 'Privacy-Preserving Data Mining', Vol. 34 of *Advances in Database Systems*, Springer US, pp. 105–136.

Clifton, C. (1999), Protecting against data mining through samples, *in* 'IFIP WG 11.3 13th Intl. Conf. on Database Security: Research Advances in Database and Information Systems Security', Kluwer, B.V., Deventer, The Netherlands, pp. 193–207.

Clifton, C. (2000), 'Using sample size to limit exposure to data mining', *Journal of Computer Security - Special Issue on Database Security* **8**, 281–307.

Dasseni, E., Verykios, V. S., Elmagarmid, A. K. & Bertino, E. (2001), Hiding association rules by using confidence and support, *in* '4th Intl. Workshop on Information Hiding', IHW '01, Springer-Verlag, London,UK, pp. 369–383.

Du, W. & Zhan, Z. (2003), Using randomized response techniques for privacy-preserving data mining, *in* '9th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining', KDD '03, ACM, New York, NY, USA, pp. 505–510.

Duraiswamy, K. (2008), 'Sensitive items in privacy preserving - association rule mining', *Journal of Information and Knowledge Management* **7**, 31–35.

Estivill-Castro, V. & Brankovic, L. (1999), Data swapping: Balancing privacy against precision in mining for logic rules, *in* M. Mohania & A. Tjoa, eds, 'Data Warehousing and Knowledge Discovery', Lecture Notes in Computer Science, Springer Berlin / Heidelberg, pp. 796–797.

Evfimievski, A. (2002), 'Randomization in privacy preserving data mining', *SIGKDD Explor. Newsl.* **4**, 43–48.

Evfimievski, A., Gehrke, J. & Srikant, R. (2003), Limiting privacy breaches in privacy preserving data mining, *in* '22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems', PODS '03, ACM, New York, NY, USA, pp. 211–222.

Evfimievski, A., Srikant, R., Agrawal, R. & Gehrke, J. (2002), Privacy preserving mining of association rules, *in* '8th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining', KDD '02, ACM, New York, NY, USA, pp. 217–228.

Gkoulalas-Divanis, A. & Verykios, V. S. (2006), An integer programming approach for frequent itemset hiding, *in* '15th ACM Intl. Conf. on Information and Knowledge Management', CIKM '06, ACM, New York, NY, USA, pp. 748–757.

Goethals, B., Laur, S., Lipmaa, H. & Mielikäinen, T. (2005), On private scalar product computation for privacy-preserving data mining, *in* C.-s. Park & S. Chee, eds, 'Information Security and Cryptology ICISC 2004', Vol. 3506 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 23–25.

Golle, P. (2006), Revisiting the uniqueness of simple demographics in the us population, *in* '5th ACM Workshop on Privacy in Electronic Society', WPES '06, ACM, New York, NY, USA, pp. 77–80.

Han, S. & Ng, W. K. (2008), Preemptive measures against malicious party in privacy-preserving data mining, *in* 'SDM'08', pp. 375–386.

Hintoglu, A. A., Inan, A., Saygin, Y. & Keskinoz, M. (2005), Suppressing data sets to prevent discovery of association rules, *in* '5th IEEE Intl. Conf. on Data Mining', ICDM 05, IEEE Computer Society, Washington, DC, USA, pp. 645–648.

Huang, Z., Du, W. & Chen, B. (2005), Deriving private information from randomized data, *in* 'ACM SIGMOD Intl. Conf. on Management of Data', SIGMOD '05, ACM, New York, NY, USA, pp. 37–48.

Inan, A., Saygyn, Y., Savas, E., Hintoglu, A. & Levi, A. (2007), 'Privacy preserving clustering on horizontally partitioned data', *Data Knowl. Eng.* **63**, 646–666.

Jagannathan, G., Pillaipakkamnatt, K., Wright, R. N. & Umano, D. (2010), 'Communication-efficient privacy-preserving clustering', *Trans. Data Privacy* **3**, 1–25.

Johnsten, T. & Raghavan, V. V. (2002), A methodology for hiding knowledge in databases, *in* 'IEEE Intl. Conf. on Privacy, Security and Data Mining', Vol. 14 of *CRPIT '14*, Australian Computer Society, Inc., Darlinghurst, Australia, pp. 9–17.

Kargupta, H., Das, K. & Liu, K. (2007), Multi-party, privacy-preserving distributed data mining using a game theoretic framework, *in* '11th European Conf. on Principles and Practice of Knowledge Discovery in Databases', PKDD 2007, Springer-Verlag, Berlin, Heidelberg, pp. 523–531.

Kargupta, H., Datta, S., Wang, Q. & Sivakumar, K. (2003), On the privacy preserving properties of random data perturbation techniques, *in* '3rd IEEE Intl. Conf. on Data Mining', ICDM '03, IEEE Computer Society, Washington, DC, USA, pp. 99–106.

Lee, G., Chang, C.-Y. & Chen, A. (2004), Hiding sensitive patterns in association rules mining, *in* '28th Annual Intl. Computer Software and Applications Conf.', Vol. 1, IEEE Computer Society, Washington, DC, USA, pp. 424 – 429.

Li, F., Ma, J. & Li, J.-h. (2009), 'Distributed anonymous data perturbation method for privacy-preserving data mining', *Journal of Zhejiang University - Science A* **10**, 952–963.

Li, G. & Wang, Y. (2011), 'A new method for privacy-preserving data mining based on weighted singular value decomposition', *Journal of Convergence Information Technology*.

Lindell, Y. & Pinkas, B. (2000), Privacy preserving data mining, *in* M. Bellare, ed., 'Advances in Cryptology', Vol. 1880 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 36–54.

Lindell, Y. & Pinkas, B. (2002), 'Privacy preserving data mining', *Journal of Cryptology* **15**, 177–206.

Liu, K., Kargupta, H. & Ryan, J. (2006), 'Random projection-based multiplicative data perturbation for privacy preserving distributed data mining', *IEEE Trans. on Knowl. and Data Eng.* **18**, 92–106.

Ma, Q. & Deng, P. (2008), Secure multi-party protocols for privacy preserving data mining, *in* Y. Li, D. Huynh, S. Das & D.-Z. Du, eds, 'Wireless Algorithms, Systems, and Applications', Vol. 5258 of *Lecture Notes in Computer Science*, Springer Berlin/Heidelberg, pp. 526–537.

Machanavajjhala, A., Kifer, D., Gehrke, J. & Venkitasubramaniam, M. (2007), 'L-diversity: Privacy beyond k-anonymity', *ACM Trans. Knowl. Discov. Data*.

Mangasarian, O. L., Wild, E. W. & Fung, G. M. (2008), 'Privacy-preserving classification of vertically partitioned data via random kernels', *ACM Trans. Knowl. Discov. Data* **2**, 12:1–12:16.

Menon, S., Sarkar, S. & Mukherjee, S. (2005), 'Maximizing accuracy of shared databases when concealing sensitive patterns', *Info. Sys. Research* **16**, 256–270.

Nabar, S. U., Marthi, B., Kenthapadi, K., Mishra, N. & Motwani, R. (2006), Towards robustness in query auditing, *in* '32nd Intl. Conf. on Very Large Data Bases', VLDB '06, VLDB Endowment, pp. 151–162.

Natwichai, J., Li, X. & Orlowska, M. (2005), Hiding classification rules for data sharing with privacy preservation, *in* '7th Intl. Conf. on Data Warehousing and Knowledge Discovery', Springer, pp. 468–477.

Natwichai, J., Li, X. & Orlowska, M. E. (2006), A reconstruction-based algorithm for classification rules hiding, *in* '17th Australasian Database Conf.', Vol. 49 of *ADC '06*, Australian Computer Society, Inc., Darlinghurst, Australia, pp. 49–58.

Oliveira, S. R. M. & Zaïane, O. R. (2002), Privacy preserving frequent itemset mining, *in* 'IEEE Intl. Conf. on Privacy, Security and Data Mining', Vol. 14 of *CRPIT '14*, Australian Computer Society, Inc., Darlinghurst, Australia, pp. 43–54.

Oliveira, S. R. M. & Zaïane, O. R. (2003*a*), Algorithms for balancing privacy and knowledge discovery in association rule mining, *in* '7th Intl. Database Engineering and Applications Symposium', Hong Kong, pp. 54–63.

Oliveira, S. R. M. & Zaïane, O. R. (2003*b*), Privacy preserving clustering by data transformation, *in* '18th Brazilian Symposium on Databases', pp. 304–318.

Oliveira, S. R. M. & Zaïane, O. R. (2003*c*), Protecting sensitive knowledge by data sanitization, *in* '3rd IEEE Intl. Conf. on Data Mining', ICDM '03, IEEE Computer Society, Washington, DC, USA, pp. 613–616.

Oliveira, S., Zaïane, O. & Saygin, Y. (2004), Secure association rule sharing, *in* H. Dai, R. Srikant & C. Zhang, eds, 'Advances in Knowledge Discovery and Data Mining', Vol. 3056 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 74–85.

Poovammal, E. & Ponnavaikko, M. (2010), 'Privacy and utility preserving task independent data mining', *Intl. Journal of Computer Applications* **1**(15), 104–111. Published By Foundation of Computer Science.

Rizvi, S. J. & Haritsa, J. R. (2002), Maintaining data privacy in association rule mining, *in* '28th Intl. Conf. on Very Large Data Bases', VLDB '02, VLDB Endowment, pp. 682–693.

Saygin, Y., Verykios, V. S. & Clifton, C. (2001), 'Using unknowns to prevent discovery of association rules', *SIGMOD Rec.* **30**, 45–54.

Shah, D. & Zhong, S. (2007), 'Two methods for privacy preserving data mining with malicious participants', *Inf. Sci.* **177**, 5468–5483.

Skinner, G., Han, S. & Chang, E. (2005), Defining and protecting meta privacy: A new conceptual framework within information privacy, *in* 'Intl. Conf. on Computational Intelligence and Security', pp. 15–19.

Sun, X. & Yu, P. S. (2005), A border-based approach for hiding sensitive frequent itemsets, *in* 'ICDM'05', pp. 426–433.

Sweeney, L. (2000), Uniqueness of simple demographics in the u.s. population, Technical Report LIDAPWP4, Carnegie Mellon University, Laboratory for International Data Privacy, Pittsburgh, PA.

Tan, V. Y. F. & Ng, S.-K. (2007), Privacy-preserving sharing of horizontally-distributed private data for constructing accurate classifiers, *in* '1st ACM SIGKDD Intl. Conf. on Privacy, Security and Trust in KDD', PinKDD'07, Springer-Verlag, Berlin, Heidelberg, pp. 116–137.

Thompson, B., Haber, S., Horne, W. G., Sander, T. & Yao, D. (2009), Privacy-preserving computation and verification of aggregate queries on outsourced databases, *in* '9th Intl. Symposium on Privacy Enhancing Technologies', PETS '09, Springer-Verlag, Berlin, Heidelberg, pp. 185–201.

Vaidya, J. & Clifton, C. (2003), Privacy-preserving k-means clustering over vertically partitioned data, *in* '9th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining', KDD '03, ACM, New York, NY, USA, pp. 206–215.

Vaidya, J. & Clifton, C. (2005), Privacy-preserving decision trees over vertically partitioned data, *in* S. Jajodia & D. Wijesekera, eds, 'Data and Applications Security XIX', Vol. 3654 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 924–924.

Verykios, V., Elmagarmid, A., Bertino, E., Saygin, Y. & Dasseni, E. (2004*a*), 'Association rule hiding', *IEEE Transactions on Knowledge and Data Engineering* **16**(4), 434 – 447.

Verykios, V. S., Bertino, E., Fovino, I. N., Provenza, L. P., Saygin, Y. & Theodoridis, Y. (2004*b*), 'State-of-the-art in privacy preserving data mining', *SIGMOD Rec.* **33**, 50–57.

Wang, E. T., Lee, G. & Lin, Y. T. (2005), A novel method for protecting sensitive knowledge in association rules mining, *in* '29th Annual Intl. Conf. on Computer Software and Applications', Vol. 2, pp. 511–516.

Wang, S. L. & Jafari, A. (2005), Using unknowns for hiding sensitive predictive association rules, *in* 'IEEE Intl. Conf. on Information Reuse and Integration', pp. 223–228.

Wang, S.-L., Lee, Y.-H., Billis, S. & Jafari, A. (2004), Hiding sensitive items in privacy preserving association rule mining, *in* 'IEEE Intl. Conf. on Systems, Man and Cybernetics', Vol. 4, pp. 3239–3244.

Xiao, M.-J., Huang, L.-S., Luo, Y.-L. & Shen, H. (2005), 'Privacy preserving id3 algorithm over horizontally partitioned data', *Intl. Conf. on Parallel and Distributed Computing Applications and Technologies* **0**, 239–243.

Yang, W. & Huang, S. (2007), Privacy preserving clustering for multi-party, *in* '12th Intl. Conf. on Database Systems for Advanced Applications', DASFAA'07, Springer-Verlag, Berlin, Heidelberg, pp. 213–224.

Yao, A. C. (1982), Protocols for secure computations, *in* '23rd Annual Symposium on Foundations of Computer Science', SFCS '82, IEEE Computer Society, Washington, DC, USA, pp. 160–164.

A Supervised Learning and Group Linking Method for Historical Census Household Linkage

Zhichun Fu[1] **Peter Christen**[1] **Mac Boot**[2]

[1] Research School of Computer Science
College of Engineering and Computer Science
The Australian National University
Canberra ACT 0200, Australia

[2] Australian Demographic and Social Research Institute
College of Arts and Social Sciences
The Australian National University
Canberra ACT 0200, Australia

Email: {sally.fu, peter.christen, mac.boot}@anu.edu.au

Abstract

Historical census data provide a snapshot of the era when our ancestors lived. Such data contain valuable information that allows the reconstruction of households and the tracking of family changes across time, allows the analysis of family diseases, and facilitates a variety of social science research. One particular topic of interest in historical census data analysis are households and linking them across time. This enables tracking of the majority of members in a household over a certain period of time, which facilitates the extraction of information that is hidden in the data, such as fertility, occupations, changes in family structures, immigration and movements, and so on. Such information normally cannot be easily acquired by only linking records that correspond to individuals. In this paper, we propose a novel method to link households in historical census data. Our method first computes the attribute-wise similarity of individual record pairs. A support vector machine classifier is then trained on limited data and used to classify these individual record pairs into matches and non-matches. In a second step, a group linking approach is employed to link households based on the matched individual record pairs. Experimental results on real census data from the United Kingdom from 1851 to 1901 show that the proposed method can greatly reduce the number of multiple household matches compared with a traditional linkage of individual record pairs only.

Keywords: Historical census data, household linkage, support vector machine, classification, group linking.

1 Introduction

Historical census data contain valuable information on individual persons and households at a given point in time. Such data allows us to reconstruct key aspects of households and families, such as birth, age, marital status, death, occupation, neighbourhood, and so on, that are of enormous value to genealogists, historians, and a wide range of other social and health scientists (Quass & Starkey 2003, Ruggles 2006, Glasson et al. 2008). As valuable as they are, these data provide only snapshots of the main characteristics of the stock of a population, capturing a vague image of how that stock and its characteristic features changed over time. To capture these changes requires that we link person by person and household by household from one census to the next over a series of censuses, a problem that hitherto has proved prohibitively expensive in time and human resources even for small groups of households (Anderson 1971). Once linked together, however, the census data are greatly enhanced in value. The linked results allow us to trace the changes in the characteristics of individual households, families and individuals over time. Linked information facilitates improved retrieval of information, and provides new opportunities for improving the quality of the data and enriches it with additional information. Along with these benefits the development of an automatic or semi-automatic household linking procedure will significantly relieve social scientists from the tedious task of manually linking individuals, families, and households and will therefore improve their productivity. This will allow them to concentrate their time and efforts on the actual analytic research and writing-up of results.

Household linking is different from record linking in several aspects. Traditional record linking compares record pairs of individuals where the similarities of key characteristics remain reasonably stable over time. Household linkage on the other hand seeks to compare pairs of households in which some or even several of the characteristics may change from one census to the next. This suggests that household linkage needs to use richer information than record linking. The emphasis on similarities between record pairs in traditional record linking arises from the fact that a high similarity suggests a good chance of matching two records. Historical census data, however, do not fit this paradigm particularly well. The data they contain are notoriously faulty and, because people's characteristics change across time, i.e., they move house, leave home, marry (and perhaps change name) and change occupations, families and households can change considerably from one census to the next. Adding to these problems is the frequency of common given names and common surnames. Moreover, because record linking is normally used as an interim step towards household linkage, the compu-

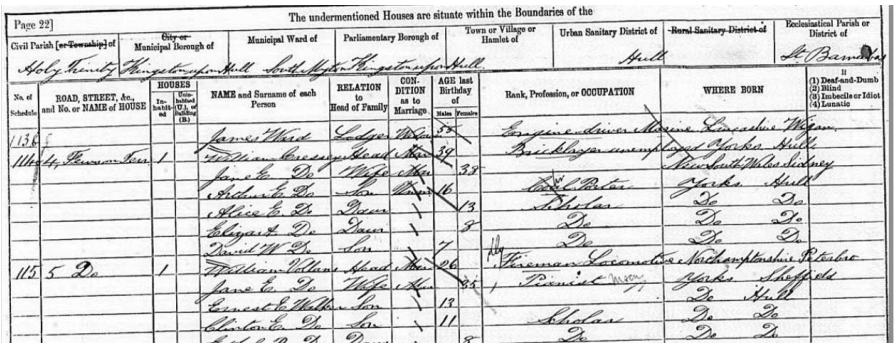


Figure 1: A sample of an original census form.

tation complexity of household linkage is higher than for individual record linkage. Together, these problems not only make it hard to find good matching record pairs, when links are made, many can have the same similarity scores, so that one record in one dataset may be linked to multiple records in another dataset.

Up to now, most research in historical census record linkage has been done by social scientists (Bloothooft 1995, 1998, Fure 2000, Quass & Starkey 2003, Ruggles 2006, Reid et al. 2006, Glasson et al. 2008). Only limited work has used the latest development of record linkage techniques to solve this problem. Vick & Huynh (2011) used the Febrl record linkage system (Churches et al. 2002, Christen & Belacic 2005) to standardise name strings in a population study of census data from the United States and Norway[1]. The authors used name dictionary and statistics of name frequencies to select the names to be cleaned and standardised. Then the Jaro-Winkler approximate string comparison algorithm (Winkler 2006) was used to match candidate names to their standard form. The effectiveness of the standardisation was validated by the fact that it can greatly reduce the number of false links. Goeken et al. (2011) have developed methods to modify the initial record linking results by consideration of the inaccuracy of historical census data collected in the late 19th century. After the initial linkage results were generated by classification of name and age similarity scores using a Support Vector Machine (SVM), name commonness and birthplace density measures were used to generate a set of new linkage results. Weights for each attribute were then generated based on a race, nativity and birthplace analysis on the two sets of linkage results, which lead to the final linked datasets. Larsen & Rubin (2001) looked at the record linking problem from a probabilistic point of view. A mixture model was first selected to divide record pairs into possible matches and non-matches using a maximum likelihood estimation. Then a manual check was performed on the data to update the estimation model. This process was iterated until few additional matches were found. It should be noted that all these work have focused on record linking, but not on household linkage.

In this paper, we introduce a method to link historical census households across time. The major contribution of our approach is to combine supervised learning and group linking methods for household linking. The proposed method first cleans and standardises the census data. Then attribute similarities between pairs of records are calculated. These similarity scores are used as inputs to an SVM classifier, which classifies record pairs into matches and non-matches. Finally, a group linking method is used to match households from different census datasets based on the outcome of the record linking step.

The rest of this paper is organised as follows. Section 2 introduces related work in the areas of data cleaning and record linkage. Section 3 describes the historical census datasets used in this study. A detailed description of the proposed method is given in Section 4, followed by experiments in Section 5. Finally, we draw our conclusions and point out future research directions in Section 6.

2 Related work

In recent years, computer science researchers, mainly in the fields of machine learning, data mining and database systems, have developed new record linkage techniques that can be used to meet the challenges posed by linking historical census data (Kalashnikov & Mehrotra 2006, Bhattacharya & Getoor 2007, On et al. 2007, Herschel & Naumann 2008, Christen 2008b). One recent set of developments are the so called "collective entity resolution" (or collective linkage) techniques (Bhattacharya & Getoor 2007). These techniques use information that explicitly connects records to collectively compute all links between records from two datasets in an overall optimal fashion. The techniques are based on unsupervised machine learning, or use graph-based approaches (Kalashnikov & Mehrotra 2006, Herschel & Naumann 2008). Experimental studies (mostly on bibliographic data) have shown that these techniques can improve linkage quality significantly compared to traditional approaches that consider only pairwise similarities between individual records.

Supervised learning has been investigated for record linking for many years. It uses a training set (labelled examples) to learn a classification model, and then applies the model to testing sets (unlabelled examples) in order to predict the classes of unlabelled examples. Among the supervised learning methods, decision trees and SVMs have been used in record linking (Elmagarmid et al. 2007). The SVM classification technique was developed by Vapnik (1995). It aims at computing a hyper-plane to classify data mapped into a high dimensional space via a kernel function. A key point here is to construct the kernel matrix for which an SVM can be used to perform the training and classification. Bilenko & Mooney (2003) proposed such a solution to compute the similarity of strings and used them as kernel matrix directly. Alternatively, Christen (2008a) constructed inputs to the SVM using a pre-selection step. In this work, a threshold method or nearest-based method was used

[1] Minnesota Population Center: http://www.ipums.org/

	1851	1861	1871	1881	1891	1901
Number of records	17,033	22,429	26,229	29,051	30,087	31,059
Number of households	3,295	4,570	5,575	6,025	6,379	6,848

Table 1: Number of records and households in the UK historical census datasets.

to select record pairs with high confidence of being a match or a non-match. Then these pairs become the positive and negative training samples for the SVM classifier. This method can be considered as a combination of supervised and un-supervised methods.

3 Application Background

The targets of this research are six census datasets collected in ten-year intervals between 1851 to 1901 for the district of Rawtenstall, a small cotton textile manufacturing town in North-East Lancashire. The data were collected on hand-filled census forms, which contains twelve attributes, such as the address of the household, full names, exact ages, sexes, their relationship to the household, occupations and places of birth of each individual residing in his or her accommodation[2]. The hand-filled census forms were transcribed manually onto enumerator's returns sheets. These sheets were subsequently scanned into digital form and, since the late 1990s, various organisations began transcribing the data from these images into tabular form and stored them in spreadsheets where they could be examined by members of the public. A sample of a scanned image is shown in Figure 1. In Table 1, we show the number of records and households in each dataset used on our experiments.

Errors are very common in the transcribed spreadsheets. This is because the original census forms were hand-filled. The English handwriting in the 19th century is quite different from nowadays. The education level of people was low, so even when instructions on how to fill-in the census had been given, many people made mistakes. Enumerators introduced errors when they transferred the data into their enumerator's returns. The quality of the digitisation varies a lot, which was highly related to the personality of the operators and even their gender.

Besides data quality problems, limited and non-standard information in historical census data is another obstacle. The UK 1851-1901 census data contain only twelve attributes (fields) for each record. Many of these attributes change significantly in a ten years interval, such as occupation and geographic mobility. Some attributes do not have values or lack standard values, for example, different names were used for the same occupation. Many nicknames had been used, for example, 'James' is the same as 'Jim', 'Charles' is the same as 'Chas'.

Because of the above problems, reconstruction of family and household data across time is difficult. Social scientists have attempted to clean and link the records manually, but the process is very expensive in terms of time and human resources required. The high cost of cleaning the data and of linking records from one census to another continues to be the principal restriction on their use for academic research.

4 Proposed Method

In this section we provide a detailed description of our proposed approach to household linking, with a focus on the linkage steps of the approach.

[2] www.uk1851census.com

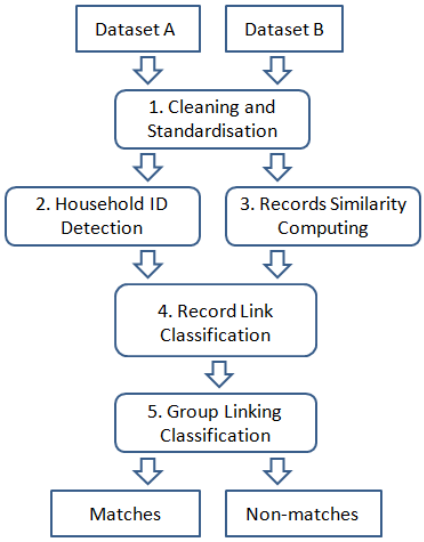


Figure 2: A flowchart of the proposed method.

4.1 Method Overview

The proposed linking method comprises five steps, as is illustrated in Figure 2. The inputs to the system are two datasets to be linked, and the output are record and household pairs that have been classified as matches or non-matches.

The first step in the approach is data cleaning and standardisation. Here, we follow the method proposed by Christen (2008b). The cleaning step aims at eliminating the errors and missing values in the data. It uses look-up tables to remove records without meaningful values, and to replace erroneous attribute values with correct values. An example is the cleaning of gender values, for example, value 'ff' is replaced with 'f'. The standardisation step formats the data into a unified form. It includes several operations, for example, converting values into lowercase letters, splitting first and middle name into two attributes, and unifying the age format into digits-only.

The second step is household detection. The purpose of this step is to assign a unique Household ID (HID) to each household. In the census datasets, we assume that the value for the 'relationship to head' attribute for each household begins with the head of the household. Therefore, we have developed a linear searching algorithm to scan through a census data file, seeking for values for the head of the household, which are 'head', 'head of family', 'widow', 'widower', 'husband', and 'married'. Each time a record has a head of household role, the HID number is incremented by one, and this HID number is assigned to all following records until another record with a head of household role is found (Fu et al. 2011).

The third step is to compute a similarity score for each pair of records under comparison. This step uses several measures to compute the similarities between individual attributes. The attribute similarities are

Attribute	Methods
Surname	Q-gram/Jaccard
First name	Q-gram/Jaccard
Sex	String exact match
Age	Absolute value differences
Occupation code	Percentage value differences
Address	Q-gram/Jaccard

Table 2: Comparison methods used for the six attributes under consideration (Christen 2008b)

concatenated into a vector which is then used in the following classification step. In the last two steps, a record link classification is performed using an SVM, and a group linking classification is used to further improve the linking results.

In the following sections, we will focus on the last three steps of the proposed method. We will address the problem of lacking a ground truth for supervised learning, and how this is solved. We will also show that due to the characteristics of historical census data, domain knowledge can be used to improve both the efficiency and the accuracy of the linking performance.

4.2 Calculating Similarities between Records

We have calculated the similarity for six selected attributes using Febrl (Christen 2008b). Appropriate similarity measures have been chosen for each attribute. A summary of the attributes compared and the corresponding similarity measures is given in Table 2. Details of these measures and their implementation have been described by Cohen et al. (2003) and Christen (2008b).

Here, we give a formal definition of the notion used for our method. Let H_1^i be the i^{th} household in the first census dataset $\mathbf{C}_1$, and $r_i^a \in H_1^i$ be the a^{th} record in this household, with $m_{1,i} = |H_1^i|$ the number of records in household H_1^i, and $1 \leq a \leq m_{1,i}$. Similarly, let H_2^j be the j^{th} household in the second census dataset $\mathbf{C}_2$, and $r_j^b \in H_2^j$ be the b^{th} record in this household, with $m_{2,j} = |H_2^j|$ the number of records in household H_2^j, and $1 \leq b \leq m_{2,j}$.

By concatenating the similarity score calculated for the six attributes shown in Table 2, we get a vector $\mathbf{x}_{r_i^a, r_j^b}$ for record r_a^i from one census dataset and r_j^b from another dataset. For convenience, we denote the similarity vector as $\mathbf{x}_{a,b}$. By summing over the similarity scores, we get a total similarity score $s_{a,b}$. In Table 3, we show the distribution of $s_{a,b}$ on all six historical census datasets under study.

Generally, $s_{a,b}$ reflects the similarity between two records. The larger the similarity value, the more similar two records are. Therefore, a simple way of finding matched pairs of records is to compare the similarity $s_{a,b}$ against a predefined threshold ρ, which is also adopted by the group linkage method by On et al. (2007). If $s_{a,b} > \rho$, the record pair is considered to be a match, otherwise it is considered a non-match. However, there are two problems with this simple method which prohibit effective record linking. Firstly, a number of factors may reduce the total similarity score between two records that belong to the same person. Such factors include, but are not limited to, errors in the data, changes of addresses or surnames, and so on. Therefore, it is difficult to find an optimal ρ for this binary classification scenario. Secondly, the summed similarity score $s_{a,b}$ does not explicitly characterise the contribution of each attribute. In order to take the advantage of separability of all attributes, we should use the full similarity vector, $\mathbf{x}_{a,b}$.

4.3 Classifying Linked Record Pair

To solve the problems with the above simple thresholding method for record linking, we investigated a supervised learning approach. More specifically, we used an SVM to classify the vectors $\mathbf{x}_{a,b}$ obtained from the record pair comparison step.

Training an SVM classifier requires training samples. Because the datasets we obtained do not contain the ground truth in the form of labels of which record pairs are matches or non-matches, we have manually identified 408 true matching record pairs by randomly sampling record links from the 1871 and 1881 datasets. We chose these two datasets because they are the middle ones among the six datasets in our collection. Thus, we assume the sampled pairs have a similar distribution as record pairs sampled from the other pairs of datasets. The labelling process was done as follows. Once a record pair is sampled, we manually decided whether or not the two records are matched. This approach of only labelling record pairs that are clearly matches or non-matches results in training data of high quality which will provide us with an accurate and robust SVM classifier.

Domain knowledge tells us that one record in a dataset must not match with more than one record in another dataset. Therefore, once a record pair is labelled as matched, all other links to the first record in the pair become non-matched. Such a sampling method has generated a large number of non-matched training samples because in the record pair comparison step an exhaustive number of record pairs has been acquired. As a consequence, we have generated a very imbalanced training set, with $314,437$ negative training samples, but only 408 positive samples.

Given the labelled binary dataset $(X, Y) = \{(\mathbf{x}_i, y_i) | i = 1, \ldots, N, y_i \in \{-1, 1\}\}$ (with class 1 being matches and class -1 being non-matches), where $\mathbf{x}_i$ are the indexed similarity vectors $\mathbf{x}_{a,b}$ and y_i are their labels, an SVM classifier recovers an optimal separating hyper-plane $\mathbf{w}^T\mathbf{x} + b = 0$ which maximises the margin of the classifier. This can be formulated as the following constrained optimisation problem (Vapnik 1995):

$$\min_{\mathbf{w},b,\xi} \quad \frac{||\mathbf{w}||^2}{2} + C\sum_i \xi_i \qquad (1)$$

$$\text{s.t.} \qquad y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \leq 1 - \xi_i \text{ and } \xi_i \leq 0$$

Here, a function ϕ is used to map the training vectors $\mathbf{x}_i$ into a higher dimensional space. $C > 0$ is the penalty parameter of the error term, and ξ is the margin slack variable. To handle the situation of imbalanced training data, we can assign a large penalty parameter for the positive class and a much smaller one for the negative class. In this study, we have set C^+ to $1000 \times C^-$.

After training, the SVM classifier is used to classify all record pairs generated by pair-wise linking all six datasets. In Table 4, we show the results of the total number of record pairs that are classified as matches, and the statistics of the number of records with single and multiple matches. As mentioned before, a record in a dataset should only be matched to at most one record in another dataset. Therefore, we have to remove those multiple matches.

	1851-1861	1861-1871	1871-1881	1881-1891	1891-1901
$s_{a,b} \in [0,1)$	431,610	705,570	891,011	981,225	1,048,323
$s_{a,b} \in [1,2)$	2,101,264	2,760,774	3,277,425	3,332,895	3,211,875
$s_{a,b} \in [2,3)$	1,926,086	2,437,898	2,860,517	2,865,787	2,665,369
$s_{a,b} \in [3,4)$	591,115	724,945	824,939	857,084	831,405
$s_{a,b} \in [4,5)$	55,053	64,462	65,908	62,317	60,316
$s_{a,b} \in [5,6)$	2,721	3,826	4,160	3,865	4,837
$s_{a,b} = 6$	187	278	239	76	296

Table 3: Distribution of Similarity scores $s_{a,b}$ on six historical census datasets.

	1851-1861	1861-1871	1871-1881	1881-1891	1891-1901
Total matched record pairs	56,301	71,752	80,802	80,504	79,442
Records involved in a single match	3,782	5,059	6,818	7,748	7,946
Records involved in multiple matches	8,784	10,910	11,965	10,406	13,034

Table 4: Record linking results on six historical census datasets based on SVM classification.

4.4 Group Linking

The group linking step aims at linking households based on the classified record links. Because the number of matched pairs generated by the SVM are larger than the number of records in both datasets, there are many multiple links. In the group linking step, if the households of the matched records in the multiple links will be compared against H_1^i one by one, then unnecessary household linking will be performed, which makes the step not efficient.

To solve this problem, three strategies can be adopted. Firstly, we can remove multiple record links by simply choosing the matched pairs with the highest $s_{i,j}$ values for each r_i^a. This will generate either a unique record link, or multiple but less record links when several links have the same $s_{i,j}$ score for each r_i^a. However, as we mentioned previously, due to erroneous data or changes in the data, exact matches are difficult to find, and $s_{i,j}$ may be low. Therefore, a true record match may not be at the top when ranked using $s_{i,j}$ only, and such a strategy will remove many true matches. The second method is to set a threshold ρ to help the decision. Record links with $s_{i,j} < \rho$ can be removed from consideration. Even if such a threshold is set, one record in a dataset still can be linked to several records in another dataset, because the corresponding similarity scores are too close or identical. Alternatively, as a third method, we can keep all record links in the group matching step. Because several linked records may belong to the same household, we calculate the best unique pairs of households that match across two census datasets.

Several group linking techniques have been proposed for bibliographic record linkage (Bhattacharya & Getoor 2007, Herschel & Naumann 2008, Kalashnikov & Mehrotra 2006). In this research, we modify the method by On et al. (2007) to link two households. For each pair of linked households, the household similarity score $\mathbb{S}_{i,j}$ between two households, H_1^i and H_2^j, can be calculated using the normalised weight of the matched individual record pairs in the two households:

$$\mathbb{S}_{i,j} = \frac{\sum_{(r_i^a, r_j^b) \in M} sim(r_i^a, r_j^b)}{m_{1,i} + m_{2,j} - |M|}. \quad (2)$$

where M is the set of record pairs matched between H_1^i and H_2^j. Here the similarity function $sim(r_i^a, r_j^b)$ can take two forms:

$$sim(r_i^a, r_j^b) = 1, \quad (3)$$

for taking the labels of matched record pairs predicted by the SVM, or

$$sim(r_i^a, r_j^b) = s_{i,j}, \quad (4)$$

for taking the sum of the attribute-wise similarity. In the former case, the group linking reduces to computing the Jaccard index (Tan et al. 2005). The second form corresponds to solving a weighted bipartite matching problem (Chartrand 1985).

Matched households can be classified by selecting the links with the highest $\mathbb{S}_{i,j}$ value. Here we assume that a household in one dataset can be matched to at most one household in another dataset. It should be mentioned here that this assumption does not always hold. The children in a household may get married or move out during the interval between two censuses. Therefore, a household can split into multiple households. However, as we mentioned at the beginning of the paper, the purpose of household linkage is to find the households which have a majority of their members matched. Thus, our purpose is to link the most 'stable' part of households.

We summarise our group linking approach in Algorithm 1. The input to the algorithm are all the matched record pairs Π between the two datasets $\mathbf{C}_1$ and $\mathbf{C}_2$, and a household $H_1^i \in \mathbf{C}_1$. The output is the household $H_2^{j*} \in \mathbf{C}_2$ which has the highest similarity to H_1^i. From Π, we can find all records in $\mathbf{C}_2$ that match to records in household H_1^i. Each of these matched records belongs to a household in $\mathbf{C}_2$, and some of them might belong to the same household. To improve the efficiency of household matching, we then merge duplicate households, so that only unique households will be used to calculate the similarities to H_1^i using Equation 2. Finally, the household(s) with the highest similarity $\tilde{\mathbb{S}}_{i,j}$ will be selected as the output H_2^{j*}.

Step 4 in Algorithm 1 is important because it improves the efficiency of the proposed method. This is because several records in a household may be matched to other records that belong to the same household. Therefore, finding unique households will reduce the number of household similarity calculations. An example of this situation is shown in Figure 4. The four records in household A are matched to five records in households B and C. Instead of calculating household similarities five times, by finding the unique matched households, we only need to conduct two similarity calculations. In this case, the number of household pairs to be linked is two.

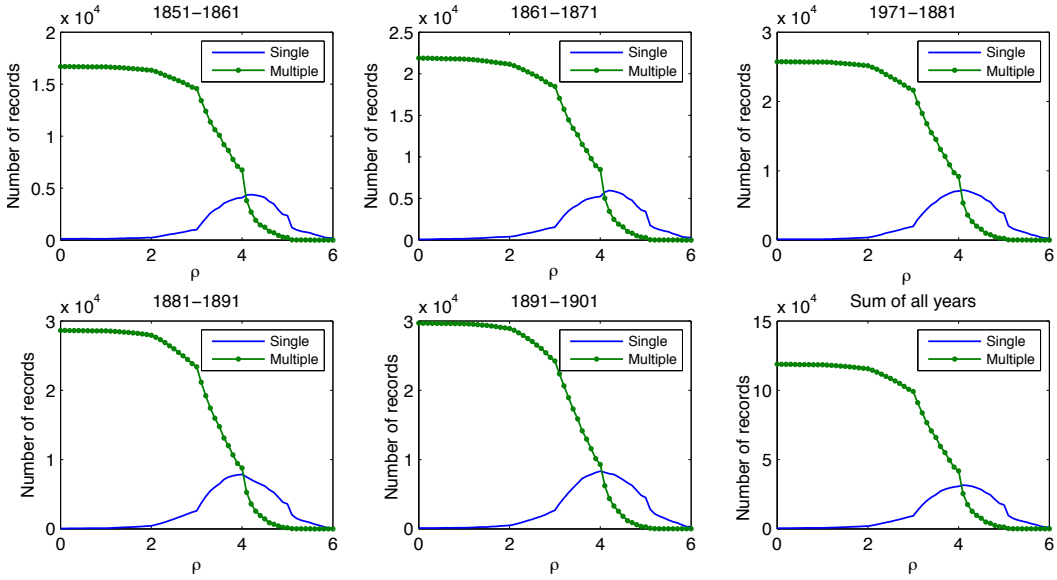


Figure 3: Record linking results using the thresholding method.

Algorithm 1: Group Linking

Input:
- Matched record pairs:Π
- All households in the second dataset: $\mathbf{C}_2$
- A household in the first dataset: H_1^i

Output:
- Best matching household: H_2^{j*}

1: **for** $r_i^a \in H_1^i$ **do**
2: Find all matched records $\{r_j^b\} \subset \mathbf{C}_2$ inΠ
3: Find households $\{H_2^j\} \subset \mathbf{C}_2$ for all r_j^b
4: Find unique households $\{\tilde{H}_2^j\} \subseteq \{H_2^j\}$
5: Calculate household similarities $\{\tilde{\mathbb{S}}_{i,j}\}$ for H_1^i and $\{\tilde{H}_2^j\}$ using Equation 2
6: Find H_2^{j*} with maximum $\tilde{\mathbb{S}}_{i,j}$

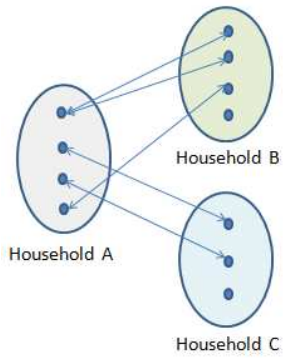


Figure 4: Example of the household (group) linking approach.

5 Experiments and Evaluation

We have conducted experiments on all six historical census datasets following the steps introduced in the previous sections. We used LIBSVM (Chang & Lin 2011) with an RBF kernel for training and testing of the record pair similarity vectors. To cope with the extremely unbalanced data in the training set, we have set the penalty parameter for the positive class to be $C^+ = 1000$ and for the negative class to be $C^- = 1$.

5.1 Experiments on Record Linking

First, we compare the performance of the SVM classifier against the thresholding method for the record linking step. Let's first consider the thresholding method. The similarity score $s_{a,b}$ for each pair of records r_i^a and r_j^b can be calculated by adding all attribute scores together. Appropriate setting of the thresholding parameter ρ is important when separating record pairs r_i^a and r_j^b into the matched and non-matched classes. We solve this problem by analysing the linking results with respect to the value of ρ. Figure 3 shows the number of records in one dataset with exactly one matched record and the number with multiple matched records in the other dataset, when different values for ρ have been set. The distribution of single matched records and multiple matched records are different for different ρ. Increasing ρ can reduce the number of records with multiple matches.

From Figure 3, two further observations can be obtained. Firstly, the curves in each plot follow a similar trend. This is consistent with the distribution of similarity scores shown in Table 3. This observation is important, because it suggests that a model trained on record similarities from any pair of datasets, or tuned on these datasets, can be applied directly to classify record pairs in other pairs of datasets as well. Secondly, the curves for only one match and for multiple matches intercept at certain points. We claim that these points can be set as the default ρ value for the group linking step. Therefore, we set $\rho = 4$ for the linking of all pairs of census datasets.

Using an SVM to perform classification of record pairs is more straightforward. As mentioned in the previous sections, we manually labelled some matched pairs in the 1871 and 1881 datasets, in total 314,437 training samples. We trained an SVM using this training set. After that, we used the trained model to classify record pairs into matched and non-matched classes, which generated the results in Table 4.

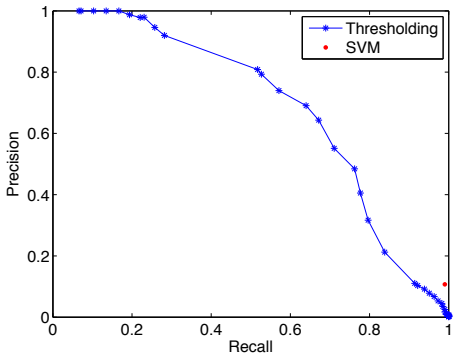


Figure 5: Training set precision and recall for SVM and thresholding for different ρ values.

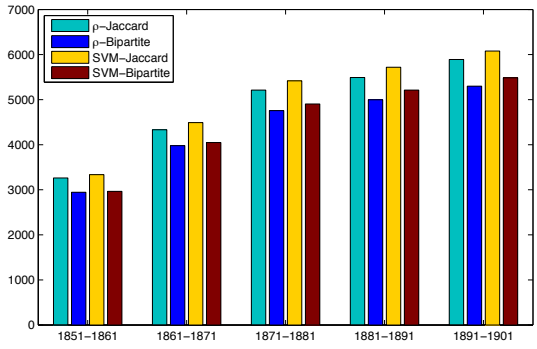


Figure 6: The number of households matched with different methods for the group linking step.

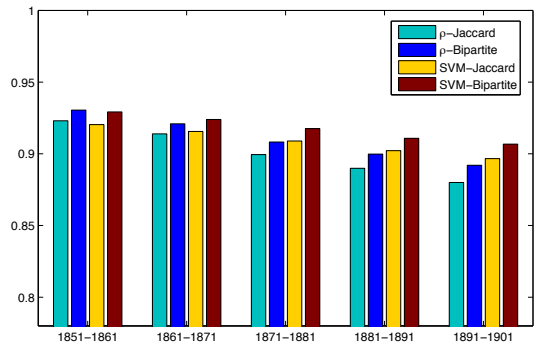


Figure 7: Group linking results shown as the percentage of reduction in the number of matched households.

We used the training set to compare the performance of the thresholding method and the SVM classification. We found that many true links had been missed when ρ was set too high in the thresholding method. For example, when ρ was set to 5.5, only 80 out of 408 pairs of matched records were found and there were no multiple matches. On the other hand, when ρ was too low, many multiple matches were generated. When ρ is set to 4, as suggested previously, 3,384 pairs of matched records were found, including 373 true matches. The SVM has generated 3,371 pairs of matched record with 404 true matches.

For further comparison, in Figure 5, we show the precision-recall curve when ρ changes. The precision and recall of the SVM classification is plotted using a red dot on the lower-right side of the graph. This plot suggests that at the same recall level, the SVM classification generates better precision than the thresholding method. The high recall score of the SVM guarantees that most true matches are retrieved. Though a high number of false matching record will be generated, this number can be greatly reduced in the following group linking step.

5.2 Group Linking

With the record linking results ready, we can perform the group linking step. Here we would like to compare four combinations of record linking and group linking methods. The methods for record linking include thresholding with ρ, and SVM classification. The methods for group linking include using either Jaccard or Bipartite metrics for the group similarity calculation. We label these four methods as ρ-Jaccard, ρ-Bipartite, SVM-Jaccard, and SVM-Bipartite. Here, we have set $\rho = 4$ for all experiments.

We start by showing in Table 5 the number of matched record pairs after the thresholding and SVM classification steps. For each of these pairs, the households that contain the record pair should also be compared. As described in Algorithm 1 and Figure 4, the number of household links can be reduced by finding the best unique household to be linked. In Table 5, we also show the number of households to be linked after such optimisation. It can be seen that the number of households generated by the SVM classification is higher than those generated by the thresholding method. This is because the number of matched record pairs for the former is higher than those from the latter. As mentioned earlier, the SVM classification generates a high number of matching records. This guarantees that less households are missed in the matching process. As a consequence, among the households detected in this step, there are many multiple links that have the same group similarity score, so that a household in the first dataset may be matched to multiple households in the second dataset.

After the group linking step, the number of matched households is greatly reduced. The total number of matched households for each matching period is shown in Figure 6, while the percentage of reduction is shown in Figure 7. From Figure 6, we can observe that when using bipartite matching in the group linking, the number of matched households is lower than the Jaccard index counterpart. This suggests that the bipartite matching is more powerful in removing multiple matches. We can also observe that the SVM-based methods generate more matched households than the thresholding-based methods, except for the period of 1851-1861. This is due to the fact that the record matching step has generated more matched record pairs when SVM classification is applied than for the thresholding method.

Figure 7 shows that higher reduction rates have been achieved on the SVM-based methods compared to the thresholding methods proposed by On et al. (2007). This is especially the case for the census datasets after 1871. In fact, all four methods under comparison have achieved high reduction rates of multiple links, with more than 87% multiple matched households removed in all the periods.

To further analyse the composition of matched households, in Table 6 we report statistics on households with single and multiple matches for the four methods under comparison. As can be seen, the num-

	1851-1861	1861-1871	1871-1881	1881-1891	1891-1901
Matched record pairs by thresholding	57,961	68,566	70,307	66,312	65,449
Household pairs to be linked after thresholding	42,360	50,312	51,815	49,868	49,070
Matched record pairs by SVM	56,301	71,752	80,802	80,504	79,442
Household pairs to be linked after SVM	41,900	53,214	59,473	58,435	58,816

Table 5: Number of matched records and household comparisons to be performed after the record linking step.

ber of households with a single match is much larger than the number with multiple matches. This suggests that our group linking method is very effective in removing the multiple matches generated in the record matching step. Among all four methods, the SVM-Bipartite method has achieved the highest number of single matches, as well as the lowest number of multiple matches. This has made this method suitable for application to historical census household linkage.

Finally, we show in Table 7 the number of households in the 1851 dataset that have been linked in periods of different lengths. The linking used the group linking results for each 10 year period reported above. For a household in the 1851 dataset, we first identified its match(es) in the 1861 dataset, then the match(es) in the 1871 dataset for each matched household in the 1861 dataset. The process continues iteratively until no match(es) can be found or until we have gone through all the datasets. All four methods have detected more than 2,200 households that have been linked over a period for 50 years. Only less than 200 households have disappeared every 10 years. Such results may occur for two reasons.

Firstly, the group linking is based on the record linking step. As long as record matches can be found for a member in a household for a 10 year period, the household linking continues for the next 10 year period. This means even if members in a household have perished or moved away, the linking process can be continued if at least one household member can be found in the following census datasets. The fact that a large number of household links has been found for the whole 50 year period tells that some children in a household tended to stay in the same area as their parents even when they've grown up and formed a new family. Therefore, such a process has generated the possibility of tracing family trees. We will manually evaluate these results with domain experts.

Secondly, such results may also be due to false matches in the record linking step. Although it is hard to judge the correctness of such matches due to lack of ground truth information, this study provides social scientists with a means to trace household changes across time. As far as we know, this is the first work of this kind in the field of historical census record linkage.

6 Conclusion

In this paper, we have introduced a novel approach to historical census household linkage. This approach first computes the similarity between record pairs. Then these similarities are used as input to an SVM classifier, which classifies record pairs into a matched and a non-matched class. The classification outcome forms the input to the household linking step. We have used a group linking technique to generate household linking similarities. The Jaccard and Bipartite measures are used in the group linking models, and their performance is compared. The results show that when combining support vector machine classification for record linking with group linking using bipartite matching, the household linkage generates better results than the alternative methods under comparison. This paper shows that the combination of supervised learning and group linkage methods for historical census household linkage is very effective. It provides social scientist with novel tools to analyse historical census data.

In the future, we will explore interactive and iterative learning methods to improve the supervised learning model. This includes learning from the instances where a household has split into multiple households between two censuses, and exploring other supervised learning approach as solution. We also plan to use a forward and backward linking method to further improve the household linking process over 20 to 50 years periods, and have the results evaluated by domain experts.

References

Anderson, M. (1971), *Family structure in nineteenth century Lancashire*, Cambridge: Cambridge University Press.

Bhattacharya, I. & Getoor, L. (2007), 'Collective entity resolution in relational data', *ACM Transactions on Knowledge Discovery from Data*, **1**(1).

Bilenko, M. & Mooney, R.J. (2003), Adaptive duplicate detection using learnable string similarity measures, *in* '9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', pp. 39–48.

Bloothooft, G. (1995), 'Multi-source family reconstruction', *History and Computing*, **7**, 90–103.

Bloothooft, G. (1998), 'Assessment of systems for nominal retrieval and historical record linkage', *Computers and the Humanities*, **32**(1), pp. 39–56.

Chang, C.-C. & Lin, C.-J., (2011), 'LIBSVM: A library for Support Vector Machines', *ACM Transactions on Intelligent Systems and Technology*, **2**(3), pp. 27.

Chartrand, G., (1995), *Introductory Graph Theory*, New York: Dover.

Christen, P. & Belacic, D. (2005), Automated probabilistic address standardisation and verification, *in* 'Australasian Data Mining Conference', pp. 53–68.

Christen, P. (2008a), Automatic training example selection for scalable unsupervised record linkage, *in* '12th Pacific-Asia Conference on Knowledge Discovery and Data Mining', Osaka, pp. 511–518.

Christen, P. (2008b), Febrl: An open source data cleaning, deduplication and record linkage system with a graphical user interface, *in* '14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', pp. 1065–1068.

	1851-1861		1861-1871		1871-1881		1881-1891		1891-1901	
	S	M	S	M	S	M	S	M	S	M
ρ-Jaccard	2,642	272	3,624	309	4,326	384	4,578	395	4,845	442
ρ-Bipartite	2,889	25	3,896	37	4,671	39	4,951	22	5,275	12
SVM-Jaccard	2,668	293	3685	357	4,497	398	4,805	405	5,025	456
SVM-Bipartite	2,956	5	4,035	7	4,886	9	5,208	2	5,478	3

Table 6: Number of households identified with single (S) and multiple (M) matches for all linked datasets.

	10 Years	20 Years	30 Years	40 Years	50 Years
ρ-Jaccard	134	133	136	119	2,392
ρ-Bipartite	140	158	165	156	2,295
SVM-Jaccard	121	132	134	132	2,442
SVM-Bipartite	120	147	157	146	2,391

Table 7: Households linked in time periods with different lengths.

Churches, T., Christen, P., Lim, K. & Zhu, J.X. (2002), Preparation of name and address data for record linkage using hidden Markov models, 'BMC Medical Informatics and Decision Making', Vol. 2, no. 9.

Cohen, W.W., Ravikumar, P. & Fienberg, S.E. (2003), A comparison of string distance metrics for name-matching tasks, *in* 'IJCAI-03 Workshop on Information Integration', pp. 73–78.

Elmagarmid, A.K., Ipeirotis, P.G. & Verykios, V. S. (2007), 'Duplicate Record Detection: A Survey', *IEEE Transactions on Knowledge and Data Engineering*, **19**, 1, 1–16.

Fu, Z., Christen, P. & Boot, M. (2011), Automatic cleaning and linking of historical census data using household information, *in* 'Workshop on Domain Driven Data Mining, held at IEEE ICDM'11', Vancouver.

Fure, E. (2000), 'Interactive record linkage: The cumulative construction of life courses', *Demographic Research*, **3**, 11.

Glasson, E., De Klerk, N., Bass, A., Rosman, D., Palmer, L. & Holman, C. (2008), 'Cohort profile: The western Australian family connections genealogical project', *International Journal of Epidemiology*, **37**, 30–35.

Goeken, R., Huynh, L., Lynch, T.A. & Vick, R. (2011), 'New methods of census record linking', *Historical Methods: A Journal of Quantitative and Interdisciplinary History*, **44**(1), 7–14.

Herschel, M. & Naumann, F. (2008), Scaling up duplicate detection in graph data, *in* '17th ACM Conference on Information and Knowledge Management', pp. 1325–1326.

Kalashnikov, D.V. & Mehrotra, S. (2006), 'Domain-independent data cleaning via analysis of entity-relationship graph', *ACM Transactions on Database Systems*, **31**(2), 716–767.

Larsen, M.D. & Rubin, D.B. (2001), 'Iterative automated record linkage using mixture models', *American Statistical Association*, **79**, 32–41.

On, B.W., Koudas, N., Lee, D. & Srivastava, D. (2007), Group linkage, *in* 'IEEE 23rd International Conference on Data Engineering', pp. 496–505.

Quass, D. & Starkey, P. (2003), Record linkage for genealogical databases, *in* '2003 ACM SIGKDD Workshop on Data Cleaning, Record Linkage, and Object Consolidation', pp. 40–42.

Rabiner, L. R. (1989), 'A tutorial on hidden Markov models and selected applications in speech recognition', *Proceedings of the IEEE*, **77**(2), 257–286.

Reid, A. , Davies, R. & Garrett, E. (2006), 'Nineteenth century Scottish demography from linked censuses and civil registers: a "sets of related individuals" approach', *History and Computing*, **14**(1+2), 61–86.

Ruggles, S. (2006), 'Linking historical censuses: A new approach', *History and Computing*, **14**(1+2), 213–224.

Tan, P., Steinbach M. & Kumar V. (2005), *Introduction to Data Mining*, Pearson Addison-Wesley.

Vapnik, V. (1995), *The Nature of Statistical Learning Theory*, Springer-Verlag.

Vick, R. & Huynh, L. (2011), 'The effects of standardizing names for record linkage: Evidence from the United States and Norway', *Historical Methods: A Journal of Quantitative and Interdisciplinary History*, **44**(1), 15–24.

Winkler, W. E. (2006), Overview of research linkage and current research directions, US Bureau of the Census, Statistical Research Report Series RRS2006/02.

Simulation Data Mining for Supporting Bridge Design

Steven Burrows[1] **Benno Stein**[1] **Jörg Frochte**[2]
David Wiesner[1] **Katja Müller**[1]

[1] Web Technology and Information Systems
Bauhaus-Universität Weimar
99421 Weimar, Germany
Email: <first>.<last>@uni-weimar.de

[2] Electrical Engineering and Computer Science
Bochum University of Applied Science
42579 Bochum, Germany
Email: joerg.frochte@hs-bochum.de

Abstract

We introduce simulation data mining as an approach to extract knowledge and decision rules from simulation results. The acquired knowledge can be utilized to provide preliminary answers and immediate feedback if a precise analysis is not at hand, or if waiting for the actual simulation results will considerably impair the interaction between a human designer and the computer. This paper reports on a bridge design project in civil engineering where the motivation to apply simulation data mining is twofold: (1) when dealing with real-world bridge models the simulation efficiency is inadequate to gain true interactivity during the design process, and (2) the designers are confronted with a parameter space (the design space) of enormous size, from which they can analyze only a small fraction. To address both issues, we propose that a database of models (the design variants) should be pre-computed so that the behavior of similar models can be used to guide decision making. In particular, simulation results based on displacement, strain, and stress analyses are clustered to identify models with similar behavior, which may not be obvious in the design space. By means of machine learning, the clustering results obtained in the simulation space can be transferred back into the design space in the form of a highly non-linear similarity measure that compares two design alternatives based on relevant physical connections. If the assessments of the measure are reliable, it will perfectly address the mentioned issues above. With this approach we break new ground, and our paper details the technology and its application for a real-world design setting.

Keywords: Engineering Design Support, Simulation Data Mining, Cluster Analysis, Similarity Measures.

1 Introduction

Simulation data mining combines the systems simulation and data mining fields to develop knowledge and intelligence from simulated data. Stein (2001) describes a diverse rationale for simulation data mining, which includes data generation when real-world sensor data is unavailable, the identification of heuristic shortcuts for complex analyses, the semi-automatic evaluation and ranking of design alternatives, and the identification of heuristic design rules. Simulation data mining has been successfully applied in different areas such as diagnosis of fluidic systems (Stein, 2003), automotive crash simulation (Painter et al., 2006), and aircraft engine maintenance (Mei and Thole, 2008). This paper deals with interactive bridge design and demonstrates the potential of simulation data mining for both improving interactivity and simplifying analyses (Burrows, 2011). Our research is part of a large, interdisciplinary project that investigates new modeling, simulation, and visualization methods for optimum bridge design.[1]

1.1 Simulation Data Mining for Design

An ideal interactive design workflow will let a designer specify design considerations in a familiar way, usually in a CAD (computer-aided design) program, and instantaneously provide the designer with feedback. An established feedback form is to render numerical simulation results obtained via a FEM (finite element method) analysis by coloring the CAD model. Feedback might also be given in the form of design alternatives, or as meta information such as reliability assessments concerning the simulation results. From this feedback, the designer evaluates the results and draws appropriate conclusions. We point out that a designer is planning and reasoning using a mental model in a kind of *design space*, while receiving feedback by interpreting selected results in a *simulation space*. See Figure 1 for an illustration.

The driving forces behind an ideal workflow are threefold: analysis performance, result comprehensibility, and ease of model manipulation. With simulation data mining, certain performance issues can be addressed: if the dialog between the human designer and the machine is impeded due to the model complexity or insufficient computing power, data mining technology can be applied to shift computation effort from the interactive (runtime) phase into a preprocessing phase. With preprocessing, design variants are instantiated and simulated offline, and the generated simulation data is exploited to learn heuristic connections between the design space and the simulation space. Of course, such heuristics are not intended to replace a deep analysis, but to guarantee a fluent dialog, since they can be evaluated at a fraction of the simulation runtime; the tentative results are superseded if the actual simulation results are at disposal. That is, the outlined advantage of simulation data mining disappears if performance bottlenecks are bypassed due to the use of less advanced models or the

This is research is supported by the German Thuringian State Ministry Grant B514-090521.

[1] "Strategies for the Robust Design of Structures", Thuringian Program on Excellence in Germany.

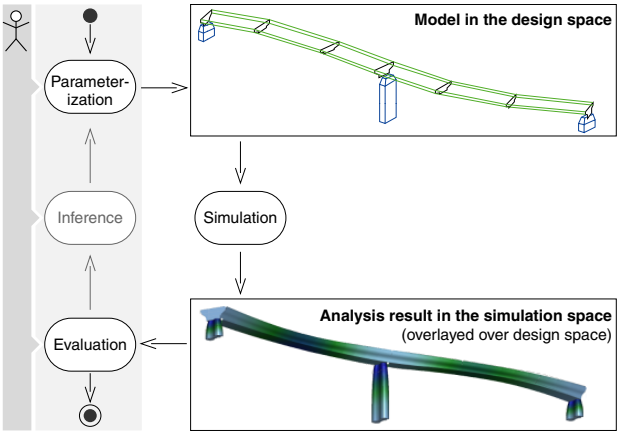


Figure 1: Interactive design workflow. Based on experience, a designer selects a model by parameterization, simulates it, evaluates the design decisions in the simulation space, and continues with a purposeful parameter modification in case of unsatisfactory results.

introduction of additional computing power, but also if the acquired heuristics are highly unreliable.

A second very interesting application of simulation data mining is not concerned with simulation speed, but with the increased volume of generated data in general. Today, in times of high computing power and seemingly endless storage capabilities, people expect that every aspect of a planned system (such as a bridge) that *could* be simulated also *should* be simulated. In other words, a design space may not be investigated for just a handful of points but for a wide range of parameter combinations. Simulation data mining then is applied to filter the set of simulation results and to identify the interesting design variants to be presented to the human designer. Even more, an exhaustive range of simulations can be used to organize the design space in the form of a topological map, exhibiting regions of good solutions, below average solutions, and unacceptable solutions. In the end, simulation data mining can drive a design optimization strategy.

1.2 Task, Approach, and Contributions

In bridge design, the task of a designer is to carefully manage a series of competing demands. The work by Spector and Gifford (1986) summarizes six kinds of such demands very well, concerning functionality, serviceability, ultimate strength, aesthetics, long-term maintainability, and cost-effectiveness, from which the last three are not relevant in our study. Conversely, the strength and serviceability of a bridge are interesting, as thresholds for key simulation outputs such as displacements, strains, and stresses can indicate likely cracking and failure, which relates to the weights and materials chosen. Moreover, the functionality of the bridge is also interesting, as it directly relates to the geometry chosen for the models to be simulated. Since the geometry design space is effectively infinite, some subset has to be selected, and we consider alternate geometries for the flat surface of a bridge model spanning a valley with three pillars. For the time being the dimensionality of the entire design space, that is, the number of parameters from a conceptual design perspective can be regarded as 10; Section 3.1 discusses the relevant dimensions of our design space in detail.

We now introduce an approach to support the outlined bridge design task using simulation data mining.[2] A first but very simple strategy to acquire design knowledge is to learn a mapping from the design space M onto selected dimensions of the simulation space Y. If such a mapping existed it could be used to answer design-relevant questions quickly, thereby circumventing an FEM simulation. But, except for simple design tasks, even sophisticated machine learning methods will fail to capture design constraints directly from raw simulation data. Instead, we start with the observation that two models $\mathbf{m}_1, \mathbf{m}_2$ in M are similar iff their simulation results (that is, their entire sets of displacement, strain, and stress values) $\mathbf{y}_1, \mathbf{y}_2$ in Y are similar:

$$\|\mathbf{y}_1 \ominus \mathbf{y}_2\| < \varepsilon \quad \Leftrightarrow \quad \varphi_{\text{Design}}(\mathbf{m}_1, \mathbf{m}_2) \approx 1, \qquad (1)$$

where $\ominus$ denotes a difference operator, and $\varphi_{\text{Design}} : M \times M \to [0;1]$ denotes a similarity function in the design space. A value of φ_{Design} close to one indicates a high similarity between two models and a value close to zero indicates a low similarity. This understanding of design similarity appears naturally and is rooted in the theory of case-based design problem solving (Maher and Pu, 1997; Antonsson and Cagan, 2001). Having φ_{Design} at our disposal we can address various issues within an interactive design workflow, such as the following:

- Given a model $\mathbf{m} \in M$, the most similar designs (in terms of behavior) can be looked-up by querying $\varphi_{\text{Design}}(\mathbf{m}, \cdot)$ in a k-nearest-neighbor fashion.
- Likewise, from the simulation results of its "design neighbors" a behavior valuation for $\mathbf{m}$ can be stated without an FEM simulation.
- From a set of models $M' \subseteq M$ that forms an equivalence class under φ_{Design}, one can learn design rules for cost optimization. Moreover, from the cardinality of M', information about the robustness of its elements may be derived.

The main contribution of our work relates to the construction of φ_{Design}. We propose to consider M as a subset of $\mathbf{R}^d$, where $d < 10$ is the dimensionality of the design space. The treatment of Y, however, is more involved: the dimensionality of the simulation space is determined by the output of an FEM analysis; that is, it depends on the resolution of the mesh discretization and introduces the according number of degrees of freedom. We hence apply an aggregation function $\alpha : Y \to Z$, which maps the original quantities of a simulation result vector $\mathbf{y} \in Y$ onto aggregate quantities $\mathbf{z} \in Z$, $\mathbf{z} \mapsto \alpha(\mathbf{y})$. While the dimensionality of Y is in 10^4 order of magnitude, that of Z is in 10^2. The function α considers physical connections and is optimized to capture as much as possible of a model's behavior characteristics. Based on these preliminaries we suggest the following steps to derive φ_{Design}, as visually summarized in Figure 2:

1. Sampling of m models $\{\mathbf{m}_i \in M \mid i = 1, \ldots, m\}$.
2. Simulation of the sampled models, yielding the set $\{\mathbf{y}_i \in Y \mid \mathbf{y}_i = \mu(\mathbf{m}_i)\}$, where μ denotes an FEM simulation algorithm.
3. Aggregation of the simulation results, yielding the set $\{\mathbf{z}_i \in Z \mid \mathbf{z}_i = \alpha(\mathbf{y}_i)\}$, where α denotes an aggregation function.

[2] Mathematical notation: sets, such as design or simulation spaces, are denoted by capital Latin characters, vectors are column vectors and denoted by small bold-faced Latin characters, and functions are denoted by small Greek letters.

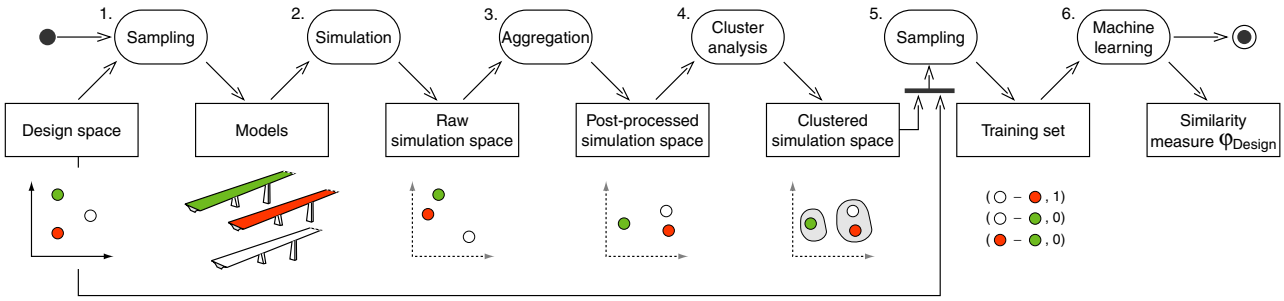


Figure 2: Six-step process to derive the similarity measure φ_{Design} for the design space.

4. Determining groups of similar models by clustering the $\mathbf{z}_i \in Z$ in the aggregated simulation space.

5. Sampling of n data points $D := \{\langle \mathbf{m}_k \ominus \mathbf{m}_l,\, c_j \rangle \mid k, l \in (1, \ldots, m),\ k < l,\ j = 1, \ldots, n\}$, where $\ominus$ denotes a difference operator and c_j is defined as follows:

$$c_j = \begin{cases} 1, \text{if } \alpha(\mu(\mathbf{m}_k)), \alpha(\mu(\mathbf{m}_l)) \text{ in same cluster,} \\ 0, \text{otherwise.} \end{cases}$$

μ and α denote the simulation algorithm and the aggregation function respectively.

6. Computing of φ_{Design} as a class probability estimator from the data points in D by means of machine learning. In particular, φ_{Design} is determined by a weight vector $\mathbf{w}^*$ that minimizes a combined error measure:

$$\mathbf{w}^* = \underset{\mathbf{w} \in \mathbf{R}^d}{\operatorname{argmin}} \sum_{(\mathbf{x}, c) \in D} L(c,\, \mathbf{w}^T \mathbf{x}) + \frac{\lambda}{2} \|\mathbf{w}\|^2,$$

where L is a loss function and λ is a non-negative regularization parameter that penalizes model complexity.

Other contributions of our research relate to the algorithmic and methodological details of the above six-step process: (1) the application of density-based clustering technology to determine equivalence classes in the simulation space, and (2) a means to evaluate φ_{Design} with machine learning by evaluating the accuracy of the mapping of the equivalence classes in both the design and the simulation spaces.

The remainder of this paper is organized as follows. Section 2 provides a literature review of simulation data mining, Section 3 details the above six-step process to derive φ_{Design}, Section 4 reports on selected experiments and analyses that illustrate the effectiveness of our approach, and Section 5 offers concluding remarks.

2 Related Work

Data mining is deployed when large numbers of data records are created and knowledge can be distilled from this data. So it seems to be quite natural to apply data mining techniques to the field of simulation. The field of possible application is wide and includes assistant systems for choosing an optimal or at least suitable simulation approach out of the set of possible algorithms (Ewald et al., 2008) as well as competing variables in semiconductor manufacturing in semiconductor factories (Brady and Yellig, 2005) or studies concerning aircraft engine maintenance (Painter et al., 2006). On closer inspection, it can be seen that the studied simulations are concentrating more on discrete event simulation fields or ones that are closely related to computer science research fields such as agent based modeling and simulation (Baqueiro et al., 2009). However, the interdisciplinary field of continuous or hybrid simulation is given less attention. An active field, where data mining is applied to finite element simulations, are crash test scenarios (Kuhlmann et al., 2005; Mei and Thole, 2008; Zhao et al., 2010), which form quite different points of view.

It is important to recall that simulation just gains knowledge about the model, and not about the real system. This fact requires additional verification and validation steps as described by Baqueiro et al. (2009). When dealing with continuous models that arise typically in engineering and natural science, like the finite element method in this paper, this step can be of less importance when dealing with pure mathematical continuous models. Also, the effects of the continuous model itself are the goal of data mining. In the work by Mei and Thole (2008), data mining techniques are used to find the reason for uncertainties in numerical simulation results. A typical reason is a change concerning regularity in the finite element model, caused for example by changes of the angle of elements around 90 degrees (Mei and Thole, 2008). So this is not a property of the real world system, but a property of the numerical model in combination with an unsatisfactory implementation.

It is quite common to pre-process the finite element method data, as noted by Kuhlmann et al. (2005): "To the authors best knowledge no approach for data mining on raw finite element data exists." The approach of Kuhlmann et al. (2005) to restore the physical properties from the finite element method data is related to our data pre-processing, but with different improvements and adoptions to the application area. In contrast to the research mentioned above, we develop an assistant system for engineers that provides real-time feedback during the design process. The process will finally be validated by a full numerical simulation, so our approach deals with both: on one hand, the real world system is the one the engineer has in mind, but on the other hand, the numerical model simulation will be the last step in this part of the design process: the user assumes that the advice given based on the data mining knowledge is close to the numerical model he or she will finally solve.

3 Development of φ_{Design}

The development of φ_{Design} follows the six steps summarized in Figure 2, which comprise sampling (from the design space), simulation, aggregation, cluster analysis, sampling (for training), and machine learning. The following subsections present all of these steps in detail.

Figure 3: A typical bridge from the design space, comprised of a curved solid surface and three pillars.

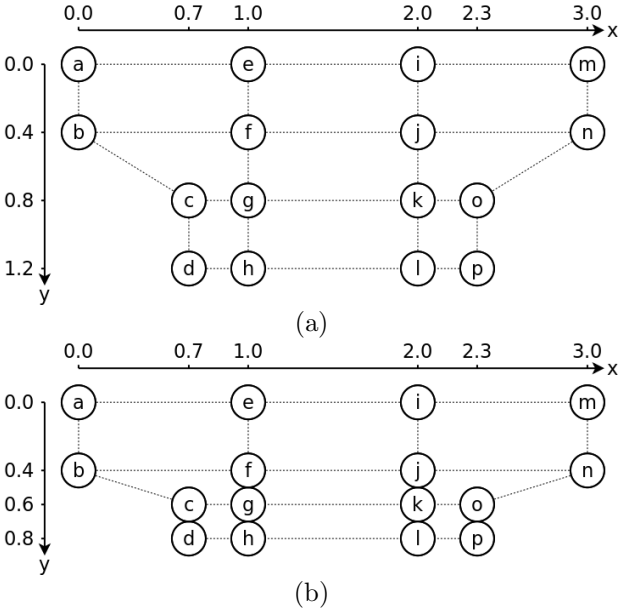


Figure 4: Two bridge surface cross-sections approximated by a trapezoid-like shape each using 16 points.

3.1 Sampling from the Design Space

All of our generated bridge models are represented using the IFC (Industry Foundation Classes) standard for data in the construction and building industries (Liebich, 2009) along with the IFC-BRIDGE extension (Lebegue et al., 2007). In this regard, researchers in our project group have added a novel extension using NURBS (non-uniform rational basis spline) solids (Hughes et al., 2005). Figure 3 shows a design variant from the design space. It is comprised of a curved solid surface and three pillars. The surface is comprised of a spline with seven 16-point cross sections evenly interspersed along the length of the spline. The two unique cross-sections are shown in Figure 4. The thicker cross-section shown in Figure 4a is positioned above each of three pillars. The remaining four cross sections shown in Figure 4b are positioned evenly between the two pairs of pillars. Observe how the NURBS modeling creates the smooth curves in Figure 3 from the original partial specifications that lack curves in Figure 4. Though the pillars are equally interesting, we have decided to not focus on these in this paper. Finally, we note that the curvature of the bridge surface is not completely symmetrical between the two pairs of pillars, so we expect different simulation results for each segment.

The design space covers the key parameters that are considered during the conceptual design phase: material properties and the geometry. Though the sensible variations in these parameters have been discussed with expert colleagues, each design dimension is sampled independently from the others to avoid both an implicit encoding of design preferences and search biases.

The material properties we explore are Young's modulus, which measures material stiffness (Mitchell and Green, 1999), and density. The values we use for these parameters are in the ranges [2e+10, 3e+10] (gigapascal) and [23, 25] (kilogram per cubic meter) respectively. The derived geometry features relate to the two bridge cross-sections as explained in Figure 4. From the figure it is clear that various cross-section heights and widths can be altered to create new geometries, such as $height_{\text{row2}-\text{row1}} = b_y - a_y$. We devise four rules for widths and three rules for heights and contract or expand the resulting values within the range [-0.3, 0.3] units to create new geometries.[3] We note that the four widths are always expanded or contracted by the same amount at any given time to preserve the rough shape of the cross section. This is less important for the three heights, but we again apply the adjustments evenly so that we can achieve the same number of height and width adjustments. Altogether, we sample 14 641 ($= 11^4$) models from the design space comprising:

1. eleven values for Young's modulus {2.0e+10, 2.1e+10, 2.2e+10, 2.3e+10, 2.4e+10, 2.5e+10, 2.6e+10, 2.7e+10, 2.8e+10, 2.9e+10, 3.0e+10}.
2. eleven values for density {23.0, 23.2, 23.4, 23.6, 23.8, 24.0, 24.2, 24.4, 24.6, 24.8, 25.0}.
3. eleven bridge surface cross-section height adjustments {-0.30, -0.24, -0.18, -0.12, -0.06, 0.00, 0.06, 0.12, 0.18, 0.24, 0.30} applied uniformly.
4. eleven bridge surface cross-section width adjustments {-0.30, -0.24, -0.18, -0.12, -0.06, 0.00, 0.06, 0.12, 0.18, 0.24, 0.30} applied uniformly.

We believe this design space is interesting for the defined model as it represents a good mixture of model properties, input parameters, and increments. However, it must nevertheless be noted that the model is not industrial-scale, so proving that our methods work is of more interest than the defined design space itself.

3.2 Simulation

A lot of phenomena in engineering are modeled using partial differential equations. One of the most popular approaches to simulate these models is the finite element method (FEM). For stationary models such as ours, the FEM only requires a discretization in space. Traditional FEM simulation uses triangles or rectangles in 2D and tetrahedrons or cuboids in 3D. Nowadays, the usage of NURBS is becoming increasingly popular. Independent of the used technique, the geometry is described by elements that are given by nodes. The number of nodes and the modeled physical phenomenon influence the number of degrees of freedom. Beyond this, the number of nodes together with the used base functions affects the quality of the simulation results. It is an oversimplification that more nodes always leads to more accurate results, because there are various conditions to fulfill, as described by Brenner and Scott (2002) and Quarteroni (2009). If these necessary conditions are fulfilled, then generally one can say that a higher number of nodes increases the quality of the results. This happens mainly because of two effects: The first is that the approximation quality of the geometry might be increased. The second is that even if the geometry has

[3] Geometries use a relative and user-interpreted unit of measurement in the simulation environment, which could be meters.

already been perfectly captured, the numerical approximation is increased. In our work, most aspects such as the number of nodes and the mathematical model have not been changed for the data mining. We in part concentrated on changing the geometry by moving the nodes. Moving the nodes on the boundary means changing the physical model, which obviously leads to different simulation results. Nevertheless, one must keep in mind that different geometries not only imply different physical behavior, but a different numerical behavior. This means that some attributes of the geometry itself might increase or decrease the numerical error, which might influence data mining processes. If the boundary changes regarding regularity (Brenner and Scott, 2002; Quarteroni, 2009), this will influence the simulation results. The aspects of the angles are strongly limited by the constraints we have given to geometry changes. If the physical parameters such as materials inside the volume are changed from one run of the simulation to the next, the FEM will have different properties. The FEM has the best properties if the parameters change very smoothly over the volume. A very rough or unfavorable discretization of the geometry might also influence the simulation results beyond the physical effects that ought to be simulated.

3.3 Aggregation

The simulation output returned from the FEM simulation uses the VTK (Visualization Toolkit) framework (Schroeder et al., 1996), which allows derived output (such as bridge displacements, strains, and stresses) to be viewed with three-dimensional models in visualization tools such as ParaView.[4] The VTK files include different sections with data concerning nodes, the resulting elements, and the simulation results at these nodes. Because the Visualization Toolkit framework is not designed to work with NURBS, the visualization and the data are mapped on a traditional mesh using hexahedrons as elements. The output in the VTK legacy format is not only used for visualization, but also for the input for our clustering and data mining processes.

The simulation output features available in our framework are displacements, strains, and stresses. A displacement is the amount of movement at any given point expressed as a vector in the three-dimensional space. Strains and stresses are instead expressed as matrices, or second-order tensors.[5] We reduce the second-order tensors to vectors using a vector triple product in order to have identical expressions for displacements, strains, and stresses. Individual measurements for displacements, strains, and stresses are expressed in the simulated output, with one measurement per point in the output mesh. Since interpolations of the finite element method simulations can provide results for thousands of points, it is therefore necessary to aggregate or combine these in some fashion, such as expressing maximums, averages, variances, standard deviations, and so on of vector measurements.

We focus on maximum measurements in the output, which is based on the idea that the most severe localized results are of high interest. The output granularity level we use interpolates 12 064 measurements from each finite element method simulation of our model, which we aggregate to 45 ($3 \times 3 \times 5$) dimensions:

- three types of output variables comprising displacements, strains, and stresses
- all three X, Y, and Z dimensions
- five divisions of the model to capture localized results comprising three at the pillars and two in between

All values are normalized in the range [0,1], so that no individual measurement completely dominates any other in a similarity measurement such as cosine. Other possible measurements (averages, variances, standard deviations) are left as future work. We note that care is needed when dealing with sets of values that differ in many orders of magnitude.

3.4 Cluster Analysis

In order to identify groups of models with similar simulated behavior, we apply the k-means (Hartigan and Wong, 1979) and Hierarchical Agglomerative Clustering (HAC) (Gowda and Krishna, 1978) clustering algorithms as competing alternatives. Both of these algorithms are *hard* clustering algorithms, meaning that each instance is assigned to only one cluster each, unlike in *soft* clustering, where each instance may be assigned to multiple clusters. The number of clusters used by each algorithm needs to be carefully considered. For k-means, the clustering must be done on a pre-defined number of clusters, which requires evaluation of many alternatives. For HAC, the clustering begins with each instance in its own cluster, and the clusters are merged one at a time in a bottom-up fashion until some optimum configuration is found, or a given threshold is reached.

In both cases, the quality of each clustering needs to be evaluated and compared with the alternatives. In this work, we use the Expected Density (Stein et al., 2003) measure to measure the quality. This measure gives a value beginning from 1.0 where a higher value denotes higher quality. A value of 1 is assigned to any clustering where the instances are either all in one cluster or are each in their own cluster, but the optimum is expected to be somewhere in between for meaningful scenarios. In practice, the expected density measure may peak at some configuration providing a meaningful recommendation for a cluster configuration to be adopted.

After some set of optimal clusterings are chosen, each pair of samples is assigned either a 1 or 0 denoting whether they appear in the same cluster or not. This data forms part of the machine learning training data described in Section 3.6.

3.5 Sampling for Training

There are many considerations for selecting a meaningful training set from the full set of pairs generated above. Perhaps most obvious is the fact that the number of within-cluster and between-cluster pairs is rarely even, so this needs attention or our machine learning algorithms may be adversely affected by the class imbalance problem (Ertekin et al., 2007). Furthermore, many clusters are likely to be of unequal size meaning that any simple random sampling strategy would overly represent the largest clusters at the expense of the smaller ones, therefore the sampling should be balanced between the clusters. A possible refinement for future work could apply bias towards clusters with the highest density. Finally, the cluster sizes are not relevant for sampling the negatives, so we simply randomly sample an equivalent number of negatives in this case. A possible refinement for

[4] `http://www.paraview.org`

[5] Note that a vector is a first-order tensor.

future work for the negative samples is to introduce bias towards the clusters that are furtherest apart.

The training samples for the machine learning step are the tuples in the form $\langle \mathbf{m}_k \ominus \mathbf{m}_l, c_j \rangle$ as described in Section 1.2. The $\mathbf{m}_k \ominus \mathbf{m}_l$ component is the vectorial difference of the four features in the design space for a pair of designs with each dimension normalized in the range [0,1], so that no individual dimension dominates the others. This leaves the c_j component, which takes on the value 1 or 0 for within-cluster or between-cluster pairs respectively, as determined by the clustering algorithm.

3.6 Machine Learning

For the machine learning step, we consider the class probability estimates of two classifiers that were shown to produce meaningful class probability estimate distributions in previous unrelated work (Burrows et al., n.d.): naive bayes and maximum entropy. Suitable implementations were taken from the Weka machine learning toolkit.[6] We apply these classifiers to generate the class probability estimates by applying ten-fold cross validation. When evaluating our approach, the machine learning accuracy scores can provide a strong endorsement of our methodology and similarity measure φ_{Design}. Also, using the ranked lists from class probability estimates, there is potential to evaluate the rankings using a rank correlation coefficient to those of the simulation space computed by the cosine distance between the aggregated results in future work.

4 Experiments and Analysis

This section first provides intermediate experiment results and analysis obtained from the clustering results and from sampling the simulation data. Then, we analyze the overall performance of φ_{Design} using our machine learning classification scores. Finally, we review other methods that may be applied to further validate our work in the future.

4.1 Clustering Results

Our expected density results of our clustering indicate that the optimum number of clusters for k-means and HAC is 12 and 37 respectively. Both distributions of results form bell-curves, making the decision simple. Ideally, these numbers should be closer, so we err on the side of caution and proceeded with both results for both cluster algorithms, making four combinations. The trends are given in Figure 5, which show that the peaks are not steep, so having some variation between clustering algorithms is tolerable.

4.2 Sampling Strategy Analysis

As described in Section 3.5, we want an equal number of positive and negative pairs, and for the positive pairs to be evenly sampled from all clusters. The minimum cluster sizes that are created is 220 instances when 12 clusters are built, and 110 instances when 37 clusters are built. Given that the number of pairs in any cluster is $\frac{n(n-1)}{2}$, this gives us 66 and 666 positive pairs per cluster, and 14 520 and 73 260 positive pairs in total respectively to satisfy our rule. These numbers then double when an equivalent number of randomly selected negative samples are chosen. Larger samples can be drawn if the smaller clusters are underrepresented.

[6] We used the classifiers weka.classifiers.bayes.NaiveBayes and weka.classifiers.functions.Logistic from Weka 3.6.5.

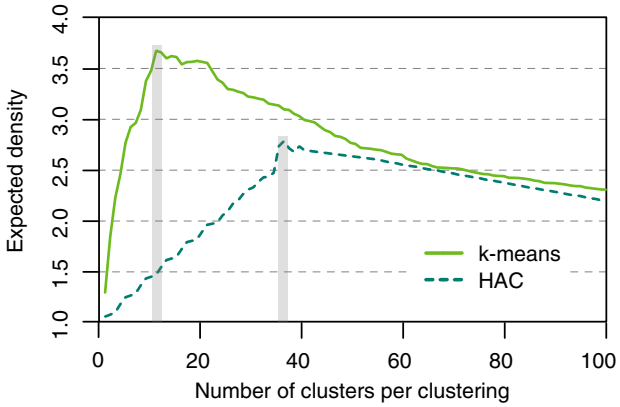


Figure 5: Expected density results for k-means and HAC clusterings from 1 to 100 clusters. These results show variation between the optimum, however there are several good choices for each clustering, so some variance is acceptable.

Algorithm	k-means		k-means		HAC		HAC	
Cluster	605	1	110	8	220	6	220	16
distribution	660	1	220	11	330	3	330	6
(size and	726	1	330	3	440	2	440	12
frequency)	990	3	363	1	11 451	1	550	1
	1 320	2	440	2			1 331	1
	1 760	4	484	2			1 980	1
			550	2				
			880	8				
Clusters	12		37		12		37	

Table 1: Cluster sizes and frequencies for k-means and HAC clustering algorithms for 12 and 37 clusters.

Table 1 shows the actual distribution of cluster sizes formed. There is a clear outlier of 11 451 instances for the largest cluster for HAC when 12 clusters are formed. This is one of the two non-optimal clusterings as far as expected density is concerned, so this anomaly further justifies the use of the expected density measure. Overall, for our two best clusterings concerning expected density, we have 10 064 780 positive and 97 107 340 negative instances for k-means with 12 clusters, and 4 865 410 positive and 102 306 710 negative instances for HAC with 37 clusters. The sampling strategy therefore only takes a small fractions of all instances available, with the proportion of available positives around 5-10% compared with the pool of negatives.

4.3 Classification Results

A consequence of generating class probability estimates for the purposes of ranking results is that we can additionally test the effectiveness of our approach by evaluating the mapping from the design space onto the within-cluster and between-cluster class labels as a binary classification experiment. We explore these accuracy results with two alternatives for each choice of clustering method, cluster size, and classification algorithm as given in Table 2. The combinations explored are the k-means and HAC classifiers using the optimum cluster size for each as determined by the expected density measure, plus the reverse setting. These four combinations of clusterings are combined with the Naive Bayes and Maximum Entropy classification algorithm choices for generating the class probability estimates. These results show accuracy scores around 92–93% for several settings, including the configurations using maximum entropy and 12 clusters in

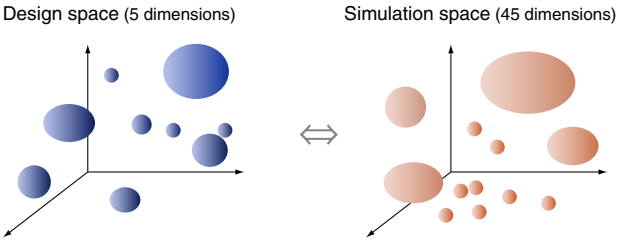


Figure 6: The purpose of φ_{Design} is to demonstrate a mapping between the design and simulation spaces.

particular. In comparison to the naive random-chance baseline of 50% accuracy, we can conclude that we have good initial results to support φ_{Design}.

4.4 Rank Correlation Co-efficients

In future work, our next step to justify φ_{Design} will be with the use of a rank correlation co-efficient such as Spearman's *rho*, Pearson's *r*, or Kendall's *tau* to compare the correlation of the ranks of φ_{Design} with the ranks of the cosine similarity taken from the simulation space. That is, we want to demonstrate some kind of correlation between the design space and simulation space as shown in Figure 6. This additional evidence will be of much value since our results demonstrating the mapping of the design space to the cluster class labels from Section 4.3 are coarse. An open problem we are working on is the generation of class probability estimates with less duplicates, as these are far from ideal for calculating correlation co-efficients relying on ranks. We will consider methods to interpolate the aggregated values from the simulation space (discussed in Section 3.3) since there are often very small differences in the values. Alternatively, different choices for clustering algorithms or sampling strategy methods (Sections 3.4 and 3.5 respectively) may also alleviate this problem.

An alternative to the rank correlation co-efficients is to consider the design space to simulation space correlations in classes such high, medium, neutral, and not similar, to treat duplicates in a more uniform way if many remain after exploring the above ideas. Therefore a rank correlation co-efficient will have to deal with duplicates in this case. Another option is to apply clustering instead of ranking for the evaluation. In this regard two cluster analyses have to be performed, one in the design space and the other in the simulation space, whereas the underlying similarity measures are φ_{Design} and the cosine similarity respectively. The two resulting clusterings are then compared by a weighted F-measure analysis, which quantifies the quality of the coverage based on weighted precision and recall values (this idea is also suggested by Figure 6). Though this comparison involves a lot of hyperparameter tuning, it should be considered as one of the most comprehensive approaches to evaluate the entire six-step process shown in Figure 2. We hence will focus on coverage-based evaluation in the near future.

5 Summary and Conclusions

To recap, real-life solid models are too complex to be processed within a reasonable amount of time and interactive design can only work if updated results triggered by model modifications from the user can be made available almost instantaneously. Since equation solving for complex FEM simulations requires a huge amount of computing power, alternative methods are consequently required to provide simulation results for complex models. With our work we introduce simulation data mining as such a method. The starting point of simulation data mining is to pre-compute a large number of models. Upon model modification by the designer, a finite element simulation as well as a φ_{Design}-search in the model database is launched, and the results of the first-completed operation are sent to the user. Depending on the size of the database, the exact model requested may not be available, but the closest available model could be returned as an approximation instead. For sufficient model complexity, the database search is order of magnitudes faster than the real simulation; the intermediate results are replaced as soon as the real simulation finishes without disturbing the designer's workflow.

We have proposed the development of φ_{Design} based on the idea that two models in the design space are similar if their simulated behavior is similar. This is necessary as the mapping is otherwise highly complex due to the numerical analysis in FEM simulation. Our similarity measure φ_{Design} is constructed based on a six-step approach comprising choosing a design space, simulating the models, aggregating the simulated results, clustering the simulated models to identify similar behavior, sampling some subset of the clustering pairs using an appropriate selection strategy, and then developing class probability estimates to show that we can map unseen examples from the design space to the similarity space.

To date, the first iteration of φ_{Design} is essentially complete. We have demonstrated that we can map the design space to our binary class labels from our

	Accuracy for k-means				Accuracy for HAC			
	12 clusters		37 clusters		12 clusters		37 clusters	
Training size	Naive bayes	Entropy	Naive bayes	Entropy	Naive bayes	Entropy	Naive bayes	Entropy
100	94.0	94.0	82.0	81.0	86.0	87.0	94.0	96.0
200	92.5	93.0	82.5	83.5	88.5	92.5	90.0	91.0
500	85.4	90.6	83.0	85.4	89.2	92.2	89.8	90.8
1 000	91.4	94.3	84.5	86.4	91.5	93.0	90.8	91.2
2 000	88.6	92.4	84.9	85.9	88.4	92.0	89.8	91.0
5 000	89.4	92.7	84.3	84.6	88.5	92.1	89.3	90.5
10 000	89.6	92.4	84.6	85.2	89.7	92.7	88.5	89.4
20 000	89.8	92.3	84.0	84.8	89.9	93.0	89.6	90.3
50 000	89.9	92.7	84.6	85.2	89.5	92.4	89.1	90.1
100 000	89.7	92.4	84.6	85.4	89.6	92.6	89.1	89.7
200 000	89.8	92.5	84.6	85.2	89.5	92.5	89.1	89.8
500 000	89.8	92.5	84.6	85.3	89.4	92.4	89.1	89.7

Table 2: Classification accuracy for k-means (columns 2-5) and HAC (columns 6-9) clustering algorithms, clustering sizes of 54 (columns 2-3 and 6-7) and 110 (columns 4-5 and 8-9), and naive bayes and maximum entropy classifiers (alternating columns). Many classification accuracy scores around 92–93% are achieved.

clusterings with high accuracy. Future work is needed to extend the verification to the class probability estimate rankings, whether we use alternative methods to develop the class probability estimates, or group or cluster the rankings. Other intermediate results are of interest where we have been able to provide some initial estimates about the number of equivalence classes in our design space and their sizes from our clustering results.

There is much scope for further development of the ideas presented in this paper. Indeed, the methodology comprises six distinct steps, where we have evaluated a small number of alternate options at each step, or simply just made a single decision. We plan to explore the available alternatives in much depth in the future.

Acknowledgements

We thank Fabian Gerold, Tim Gollub, Katja Müller, Michael Schwedler, and David Wiesner for their technical support on this project beyond the research contributions of this paper.

References

Antonsson, E. and Cagan, J. (2001), *Formal Engineering Design Synthesis*, Cambridge University Press.

Baqueiro, O., Wang, Y. J., Mcburney, P. and Coenen, F. (2009), Integrating Data Mining and Agent Based Modeling and Simulation, *in* P. Perner, ed., 'Proceedings of the Ninth Industrial Conference on Advances in Data Mining', Springer-Verlag, Leipzig, Germany, pp. 220–231.

Brady, T. F. and Yellig, E. (2005), Simulation Data Mining: A New Form of Computer Simulation Output, *in* M. E. Kuhl, N. M. Steiger, F. B. Armstrong and J. A. Joines, eds, 'Proceedings of the Thirty-Seventh Winter Simulation Conference', Orlando, Florida, pp. 285–289.

Brenner, S. C. and Scott, L. R. (2002), *The Mathematical Theory of Finite Element Methods*, second edn, Springer, Berlin, Germany.

Burrows, S. (2011), Simulation Data Mining in Artificially Generated Data, *in* H. Cunningham, O. Etzioni, N. Fuhr and B. Stein, eds, 'Proceedings of the Schloss Dagstuhl Seminar on Challenges in Document Mining', Leibniz-Zentrum für Informatik, Wadern, Germany. Invited talk.

Burrows, S., Potthast, M. and Stein, B. (n.d.), 'Paraphrase Acquisition via Crowdsourcing and Machine Learning'. In submission.

Ertekin, S., Huang, J. and Giles, C. L. (2007), Active Learning for Class Imbalance Problem, *in* W. Kraaij, A. P. de Vries, C. L. A. Clarke, N. Fuhr and N. Kando, eds, 'Proceedings of the Thirtieth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval', ACM, Amsterdam, The Netherlands, pp. 823–824.

Ewald, R., Himmelspach, J. and Uhrmacher, A. M. (2008), An Algorithm Selection Approach for Simulation Systems, *in* F. Quaglia and J. Liu, eds, 'Proceedings of the Twenty-Second Workshop on Principles of Advanced and Distributed Simulation', IEEE Computer Society, Rome, Italy, pp. 91–98.

Gowda, K. C. and Krishna, G. (1978), 'Agglomerative Clustering using the concept of Mutual Nearest Neighbourhood', *Pattern Recognition* **10**(2), 105–112.

Hartigan, J. A. and Wong, M. A. (1979), 'Algorithm AS 136: A K-Means Clustering Algorithm', *Journal of the Royal Statistical Society* **28**(1), 100–108.

Hughes, T. J. R., Cottrell, J. A. and Bazilevs, Y. (2005), 'Isogeometric Analysis: CAD, Finite Elements, NURBS, Exact Geometry and Mesh Refinement', *Computer Methods in Applied Mechanics and Engineering* **194**(39–41), 4135–4195.

Kuhlmann, A., Vetter, R.-M., Lübbing, C. and Thole, C.-A. (2005), Data Mining on Crash Simulation Data, *in* P. Perner and A. Imiya, eds, 'Proceedings of the Sixth International Conference of Machine Learning and Data Mining', Springer Berlin / Heidelberg, Leipzig, Germany, pp. 635–635.

Lebegue, E., Gual, J., Arthaud, G. and Liebich, T. (2007), IFC-BRIDGE V2 Data Model, Technical Report Edition R8, buildingSMART International Modeling Support Group.

Liebich, T. (2009), IFC 2x Edition 3 Model Implementation Guide, Technical Report Version 2.0, buildingSMART International Modeling Support Group.

Maher, M. and Pu, P., eds (1997), *Issues and Applications of Case-Based Reasoning in Design*, Lawrence Erlbaum Associates, Mahwah, New Jersey.

Mei, L. and Thole, C.-A. (2008), 'Data Analysis for Parallel Car-Crash Simulation Results and Model Optimization', *Simulation Modelling Practice and Theory* **16**(3), 329–337.

Mitchell, B. and Green, A. (1999), 'The Young Modulus', `http://www.matter.org.uk/schools/content/YoungModulus`. Accessed 21 August, 2011.

Painter, M. K., Erraguntla, M., Hogg, G. L. and Beachkofski, B. (2006), Using Simulation, Data Mining, and Knowledge Discovery Techniques for Optomized Aircraft Engine Fleet Management, *in* D. Nicol, R. Fujimoto, B. Lawson, J. Liu, F. Perrone and F. Wieland, eds, 'Proceedings of the Thirty-Eighth Winter Simulation Conference', IEEE Press, Monterey, California, pp. 1253–1260.

Quarteroni, A. (2009), *Numerical Models for Differential Problems*, first edn, Springer, Milan, Italy.

Schroeder, W. J., Martin, K. M. and Lorensen, W. E. (1996), The Design and Implementation of an Object-Oriented Toolkit for 3D Graphics and Visualization, *in* R. Crawfis, C. Hansen, N. Max and S. Uselton, eds, 'Proceedings of the Seventh Conference on Visualization', IEEE Computer Society Press, San Francisco, California, pp. 93–99.

Spector, A. and Gifford, D. (1986), 'A Computer Science Perspective on Bridge Design', **29**, 267–283.

Stein, B. (2001), Model Construction in Analysis and Synthesis Tasks, Habilitation, University of Paderborn, Germany, Department of Mathematics and Computer Science.

Stein, B. (2003), Model Compilation and Diagnosability of Technical Systems, *in* M. Hanza, ed., 'Proceedings of the Third International Conference on Artificial Intelligence and Applications', ACTA Press, Anaheim, Calgary, Zurich, pp. 191–197.

Stein, B., Meyer zu Eißen, S. and Wißbrock, F. (2003), On Cluster Validity and the Information Need of Users, *in* M. Hanza, ed., 'Proceedings of the Third International Conference on Artificial Intelligence and Applications', ACTA Press, Anaheim, Calgary, Zurich, pp. 216–221.

Zhao, Z., Jin, X., Cao, Y. and Wang, J. (2010), 'Data Mining Application on Crash Simulation Data of Occupant Restraint System', *Expert Systems with Applications* **37**, 5788–5794.

Zero-day Malware Detection based on Supervised Learning Algorithms of API call Signatures

Mamoun Alazab[1], Sitalakshmi Venkatraman[1], Paul Watters[1], and Moutaz Alazab[2]

[1]Internet Commerce Security Laboratory
School of Science, Information Technology & Engineering
University of Ballarat

{m.alazab, s.venkatraman, p.watters } @ ballarat.edu.au

[2]School of Information Technology
Deakin University, Australia

{malazab} @ deakin.edu.au

Abstract

Zero-day or unknown malware are created using code obfuscation techniques that can modify the parent code to produce offspring copies which have the same functionality but with different signatures. Current techniques reported in literature lack the capability of detecting zero-day malware with the required accuracy and efficiency. In this paper, we have proposed and evaluated a novel method of employing several data mining techniques to detect and classify zero-day malware with high levels of accuracy and efficiency based on the frequency of Windows API calls. This paper describes the methodology employed for the collection of large data sets to train the classifiers, and analyses the performance results of the various data mining algorithms adopted for the study using a fully automated tool developed in this research to conduct the various experimental investigations and evaluation. Through the performance results of these algorithms from our experimental analysis, we are able to evaluate and discuss the advantages of one data mining algorithm over the other for accurately detecting zero-day malware successfully.

The data mining framework employed in this research learns through analysing the behavior of existing malicious and benign codes in large datasets. We have employed robust classifiers, namely Naïve Bayes (NB) Algorithm, k−Nearest Neighbor (kNN) Algorithm, Sequential Minimal Optimization (SMO) Algorithm with 4 differents kernels (SMO - Normalized PolyKernel, SMO – PolyKernel, SMO – Puk, and SMO- Radial Basis Function (RBF)), Backpropagation Neural Networks Algorithm, and J48 decision tree and have evaluated their performance.

Overall, the automated data mining system implemented for this study has achieved high true positive (TP) rate of more than 98.5%, and low false positive (FP) rate of less than 0.025, which has not been achieved in literature so far. This is much higher than the required commercial acceptance level indicating that our novel technique is a major leap forward in detecting zero-day malware. This paper also offers future directions for researchers in exploring different aspects of obfuscations that are affecting the IT world today.

Keywords: Malware, Intrusion Detection, Obfuscation, API.

1 Introduction

Windows API enables the programs to exploit the power of Windows and hence malware authors make use of the API calls to perform malicious actions (Tamada et al. 2006; Sharif et al. 2008; Choi et al. 2009; Alazab, Venkatraman & Watters 2010). This approach has enabled malware authors to adopt various obfuscation techniques, thereby posing a major challenge to the existing Anti-Virus (AV) detection engines. Literature provides several approaches being investigated to thwart such malware, and among these approaches, data mining methods have been more successful in detecting these recent malware that adopt API calls for infecting the executables (Symantec Enterprise Security, 1997; Sung et al. 2004; Kolter & Maloof 2006). However, due to the requirement of high accuracy in the case of malware detection with very low false positives, such methods have not been effective in the practical world (Venkatraman, 2010; RSA, 2011; Symantec Enterprise Security, 2011). In this research, we have employed a novel and effective method of extracting API call features and in training the classifiers using several data mining techniques for an efficient detection of obfuscated malware that lead to zero-day (unknown) attacks of today.

1.1 Malware Obfuscation

The term obfuscation means modifying the program code in a way to preserve its functionally with the aim to

reduce vulnerability to any kind of static analysis and to deter reverse engineering by making the code difficult to understand and less readable (Linn & Debray 2003). Obfuscation techniques such as packing, polymorphism and metamorphism are used by malware authors as well as legitimate software developers (Alazab, Venkatraman & Watters 2009). They both use code obfuscation techniques for different reasons. Code obfuscation is very effectively used by malware authors to evade antivirus scanners since it modifies the program code to produce offspring copies which have the same functionality but with different byte sequence or 'virus signature' that is not recognized by antivirus scanners (Rabek et al. 2003).

1.2 Windows API Calls

Windows API calling sequence reflects the behaviour of executables. The Windows API function calls fall under various functional levels such as system services, user interfaces, network resources, windows shell and libraries. Since the API calls reflect the functional levels of a program, analysis of the API calls would lead to an understanding of the behaviour of the file. Malicious codes are able to disguise their behaviour by using API functions provided under Win32 environment to implement their tasks. Therefore, in binary static analysis, the focus is on identifying all documented Windows API call features to understand the malware behaviour.

In the Windows operating system, user applications rely on the interface provided within a set of libraries, such as KERNEL32.DLL, NTDLL.DLL and USER32.DLL in order to access system resources including files, processes, network information and the registry. This interface is known as the Win32 API. Applications may also call functions in NTDLL.DLL known as the Native API. The Native API functions perform system calls in order to have the kernel provide the requested service. In our previous works (Alazab 2010; Alazab et al. 2010; Alazab, Venkatraman & Watters 2010) we have demonstrated how to extract and analyse these API call features including hooking of the system services that are responsible to manage files. The extracted calls are confined to those that affect the files. Various features related to the calls that create or modify files or even get information from the file to change some value and information about the DLLs that are loaded by the malware before the actual execution are considered for the analysis.

1.3 Need for the Study

Recently, API calls have been explored for modeling program behaviour. There are studies (Choi et al. 2009) (Tamada et al. 2006) (Park et al. 2008b) (Park et al. 2008a) that have used analysis of API calls for generation of birthmark on portable execution. Use of statistical analysis of file binary content including statistical N-gram modeling techniques (Stolfo, Wang & Li 2005) (Wang et al. 2009) have been tested in identifying malcode in document files and does not have sufficient resolution to represent all class of file types. From other study on related work (Venkatraman 2009) (Bruschi, Martignoni & Monga 2006) (MetaPHOR 2010) (Perriot & Ferrie 2004) (Ferrie & Szor 2001) (Ferrie & Szor 2001) (Linn & Debray 2003) (Chang & Atallah 2002), it has been found that the statistical modeling of hidden malcode that predominantly use Windows API calling sequence for evading detection is yet to be explored. This is a motivation for this research towards a positive contribution in understanding malware behavior through statistical analyses of API calls.

The analysis of computer system performed offline is called static analysis, which has been employed in this research to study the patterns of the API calls within binary executables by reverse engineering the code. Static analysis provides a better understanding of the anomalous behavior patterns of the code since we adopt a methodology to perform a deep analysis into the code program and their statistical properties. The existing techniques and methods exhibit false positives as they do not perform sufficient statistical analysis to determine if the anomaly was 'actually' malicious (Jacob et al., 2008; Symantec Enterprise Security, 2011). Therefore, in this research, static anomaly-based detection analysis is adopted to perform introspection of the program code with the goal of determining various dynamic properties of API function calls that are extracted from these codes in an isolated environment.

The results of the following recent studies have been the prime motivation for this research: 1) malware authors are able to easily fool the detection engine by applying obfuscation techniques on known malwares (Sharif et al. 2008), 2) identifying benign files as malware (false positive) is becoming very high (Symantec Enterprise Security 2011), 3) failing to detect obfuscated malware is high (false negative) (Symantec Enterprise Security 2011) (Symantec Enterprise Security 2010) (Symantec Enterprise Security 2009), 4) the current detection rate is decreasing, and 5) current malware detectors are unable to detect zero day attacks (Symantec Enterprise Security 2011) (RSA 2011). These results imply that code obfuscation has become a challenge for digital forensic examiners with the limitations of signature based detection (Tang, Zhou & Zuo 2010) (Santos et al. 2009).

2 Data Mining

In the recent years, data mining has become the focus of many malware researchers for detecting unknown Malware or to classify malware from benign files. Data mining is usually referred to as knowledge discovery in databases. Frawley et al. (Frawley, Piatetsky-shapiro & Matheus 1992) define it as "The nontrivial extraction of implicit, previously unknown, and potentially useful information from data". It is also defined as "The science of extracting useful information from large data sets or databases" (Hand, Mannila & Smyth 2001). In this research, data mining involves the application of a full suite of statistical and machine learning algorithms on a set of features derived from malicious and clean programs.

Features form the input data to the detection systems, and features can be used as patterns for classification in malware detection systems. Reverse engineering of executables results in extracted features useful for two types of detections: i) Host-Based Intrusion Detection System (HIDS) – to check, analyse and monitor the computer system internally such as extract byte

sequences, ASCII, instruction sequences, and API call sequences, and ii) Network-Based Intrusion Detection system (NIDS) – to detect malicious activity by monitoring network traffic such as denial of service (DOS) attacks, port scans.

A data mining approach to malware detection usually involves employing statistical methods for classification. Each classification algorithm constructs a model, using machine learning, to represent the benign and malicious classes. In this approach, a labeled training set is required to build the class models during a process of supervised learning. Many statistical classification algorithms exist including Naive Bayes (NB) Algorithm, k−Nearest Neighbor (kNN) Algorithm, Sequential Minimal Optimization (SMO) Algorithm with 4 different kernels (SMO - Normalized PolyKernel, SMO – PolyKernel, SMO – Puk, and SMO- Radial Basis Function (RBF)), Backpropagation Neural Networks Algorithm, Logistic Regression, and J48 decision tree. The key to statistical classification is to represent the malicious and benign samples in an appropriate manner to enable the classification algorithms to work effectively. Feature extraction is an important component of effective classification, and an associated feature vector that can accurately represent the invariant characteristics in the training sets and query samples is highly desirable. Classification is the process of classifying data into two or more predetermined groups based on features.

2.1 Related Study

Data mining techniques for malware detection usually starts with the first step of generating a feature set. In 2005, studies reported in (Malan & Smith 2005) that a temporal consistency element was added to the system call frequency to calculate the frequency of API system call sequences. Similarity measures were calculated using edit distance and Measuring Similarity with Intersection. The first measure was on ordered sets of native API system calls, while the second one was on unordered sets. Both similarity measures based on API gave the probabilities of two peers. The drawback is that they had considered only native API call features.

Static Analysis for Vicious Executables (SAVE) (Sung et al. 2004) is another work based on API calls made in an attempt to detect polymorphic and metamorphic malwares. They defined signature as an API sequence of calls and started the reverse engineering process from decompressed 16 binaries, which are then passed through a PE file parser. Next, they extracted and mapped the sequence of Windows API calls, and lastly passed them through the similarity measure module, where similarity measures such as, Euclidian distance, Sequence alignment, Cosine measure, extended Jaccard measure, and the Pearson correlation measure were used. Binary executables under inspection is classified by identifying a high similarity to a known instance of malware in the training set. Although these similarity measures enable SAVE to detect polymorphic and metamorphic malwares efficiently against 8 malware scanners, their weakness is not being able to detect unknown malware.

Another signature-free system to detect polymorphic malware and unknown malware based on the analysis of Windows API execution sequences extracted from binary executable is called Intelligent Malware Detection System (IMDS) (Ye et al. 2007). IMDS was developed using Objective-Oriented Association (OOA) mining based classification with large data set gathered for the experiment (29580 binary executables, of which 12214 were benign binary executables and 17366 were malicious binary ones). For detection, a Classification Based on Association rules (CBA) technique such as Naïve Bayes, SVM and Decision Tree were used. The result was compared against anti-virus software's such as Norton, Kaspersky, McAfee, and Dr.Web. In 2010 the authors of IMDS had incorporated the CIDCPF method into their existing IMDS system with larger dataset, and called it CIMDS system (Yanfang et al. 2010). CIDCPF adapted the post processing techniques as follows: first Chi-square testing was applied and Insignificant rule pruning followed by using Database coverage based on the Chi-square measure rule ranking mechanism and Pessimistic error estimation, and finally prediction was performed by selecting the best First rule. Their results were good, but involved unbalanced test data while the training data was quite balanced. Also, the detection rate was for training set about 89.6% and the accuracy was approximately 71.4 and in the testing set about 88.2% and the accuracy was approximately 67.6 which still the work need to be improve to achieve higher detection rate and higher overall accuracy.

In 2006, researchers (Kolter & Maloof 2006) described the use of machine learning and data mining to detect and classify malicious executables. They tested several classifiers including, IBk, naive Bayes, support vector machines (SVMs), decision trees, boosted naive Bayes, boosted SVMs, and boosted decision trees. Kolter found that support vector machine performed exceptionally well and fast as compared to the other classifiers. Hence, for the obfuscated malware detection system, this research adopts SVM as a classifier for the detection of hidden malware that invariably uses API call sequence.

API based features are not only good in classification of malware, but is also good in detecting injected malicious executable. DOME (Rabek et al. 2003) is a host-based technique that uses static analysis based on monitoring and validating Win32 API calls for detecting malicious code in binary executables. In a study on the performance of kernel methods in the context of robustness and generalization capabilities of malware classification (Shankarapani et al. 2010), results revealed that analysis based on the Win API function call provides good accuracy to classify malware.

3 Methodology

This section describes the overall methodology adopted as shown in Figure 1, which consists of three groups of processes; In the first group, the following 3 steps have been employed: Step 1: Unpack the malware and disassemble the binary executable to retrieve the assembly program, Step 2: Extract API calls and important machine-code features from the assembly program, and Step 3: Map the API calls with MSDN library and analyse the malicious behaviour to get the API sequence from the binaries. In the second group,

after getting the API sequences from the binaries, the signature database is updated based on their API calls. This sequence is compared to a sequence or signature (from the signature database) and is passed through the similarity measure module to generate the similarity report. In the third group, Mutual Information (MI) based Maximum Relevance (MR) filter ranking heuristics on the set of Win API function calls is used for feature selection of relevant features, which provides more information about the class variables than irrelevant features. After getting the best features on the set of Win API calls, supervised learning method has been applied that uses a dataset to train, validate and test, an array of classifiers. Eight robust classifiers have been selected for this purpose, namely, Naive Bayes (NB) Algorithm, k−Nearest Neighbor (kNN) Algorithm, The Sequential Minimal Optimization (SMO) Algorithm with 4 differents kernels (SMO - Normalized PolyKernel, SMO – PolyKernel, SMO – Puk, and SMO- Radial Basis Function (RBF)), Backpropagation Neural Networks Algorithm, and J48 decision tree. However, the classification methods require training data and data to validate the models that have been formulated. Therefore, K-fold cross-validation has been used for evaluating the results of a statistical analysis generating an independent dataset. Extensive testing and analysis performed on malware and benign datasets with different learning techniques have shown that 10 folds provide the best estimate of error. Having k=10 folds means 90% of full data is used for training (and 10% for testing) in each fold test. Evaluation (feature selection + classification) was done inside 10-fold cross-validation loop on all Malware and benign datasets.

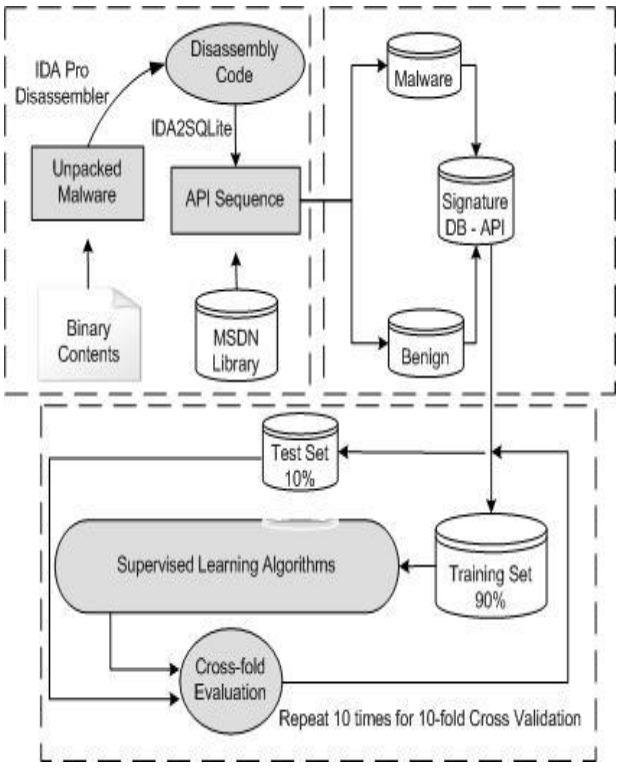


Figure 1. System overview of zero-day Malware detection methodology

Our proposed methodology mainly focuses on the API call features and the similarity based detection for identifying unknown malware and classifying them with existing malware families. Figure 1 gives a system overview of the similarity based detection methodology adopted. Signature database has been used to statistically calculate and compute the similarity measures. Eight distance measures have been adopted to analyse and differentiate between malware variants and benign executables from various families. The Signature database has been generated from the existing malware datasets to produce fingerprint or benchmark for each record based on the API calls that the executable programs had used as shown in table 1.

The signature database has been used later in next section to measure the distances between the programs and subsequently the results are analysed to classify these programs based on their API call features.

TABLE. 1 Database Signatures

ID	Called API	Class
Prg.1	$\sum_{i=0}^{5} API_1, \sum_{i=0}^{1} API_3, \sum_{i=0}^{10} API_{16}$	Malware
Prg.2	$\sum_{i=0}^{7} API_3, \sum_{i=0}^{2} API_{11}$	Malware
Prg.3	$\sum_{i=0}^{3} API_{22}, \sum_{i=0}^{4} API_1, \sum_{i=0}^{1} API_{20}, \sum_{i=0}^{2} API_6, \sum_{i=0}^{5} API_4, \sum_{i=0}^{2} API_3, \sum_{i=0}^{5} API_8$	Malware
Prg.5	$\sum_{i=0}^{2} API_{11}, \sum_{i=0}^{2} API_4, \sum_{i=0}^{7} API_6, \sum_{i=0}^{16} API_{10}$	Benign
Prg.6	$\sum_{i=0}^{5} API_4$	Benign
..	…	…

4 Database

We have gathered 66,703 executable files in total consisting of 51,223 recent Malware datasets and the remaining being benign datasets as shown in table 2. Such large malware datasets with obfuscated and unknown malware used in this research study have been collected from honeynet project, VX heavens (VX Heavens 2011) and other sources.

The 15,480 benign datasets include: Application software such as Databases, Educational software,

Mathematical software, Image editing, Spreadsheet, Word processing, Decision making software, Internet Browser, Email and many others system software, Programming language software and many other applications. Both (Malware, Benign) have been uniquely named according to their MD5 hash value.

TABLE. 2 Dataset

Type	Qty	Max. Size (KB)	Min. Size (KB)	Avg. Size (KB)
Benign	15,480	109,850	0.8	32,039
Virus	17,509	546	1.9	142
Worm	10,403	13,688	1.6	860
Rootkit	270	570	2.8	380
Backdoor	6,689	1,299	2.4	685
Constructor	1,039	77,662	0.9	1,193
Exploit	1,207	22,746	0.5	375
Flooder	905	16,709	1	1,397
Trojan	13,201	17,810	0.7	1,819

5 Evaluation and Validation

The classification algorithms require training data to train the formulated models, and testing data to test those models. Validation of the models is achieved by making a partition on the database of malware and benign for carrying out the experiments. The cross-validation is a technique used for evaluating the results of a statistical analysis by generating an independent dataset for Malware and benign. The most common types of cross-validation are repeated random sub-sampling validation and K-fold cross-validation (Hand, Mannila, & Smyth, 2001). For this research study of Malware and Benign classification, K-fold cross-validation has been selected for validation as it is commonly adopted for many classifiers (Witten and Frank, 2010; Bhattacharyya, etal 2011).

In k-fold cross-validation the data is first partitioned into k sized segments or folds. Then, k iterations of training and validation are performed such that within each iteration a different fold of the data is held-out for validation while the remaining k-1 folds are used for learning. The advantage of K-Fold cross-validation is that all the examples in the dataset are eventually used for both training and testing. Also, all observations are used for both training and validation, and each observation is used for validation exactly once.

We have evaluated various algorithms based on the following standard performance measures:

- True Positive (TP): Number of correctly identified malicious code,
- False Positive (FP): Number of wrongly identified benign code, when a detector identifies benign file as a malware.
- True Negative (TN): Number of correctly identified benign code.
- False Negative (FN): Number of wrongly identified malicious code, when a detector fails to detect the malware because the virus is new and no signature is yet available.
- True detection Rate (TP rate): Percentage of correctly identified malicious code.

$$TP\ Rate = \frac{TP}{TP + FN}$$

- False alarm Rate (FP rate): Percentage of wrongly identified benign code, given by:

$$FP\ Rate = \frac{FP}{FP + TN}$$

- F-Measure: It is a measure of a test's accuracy by combining recall and precision scores into a single measure of performance, usually it is between 0.0 and 1.0 closer to 1 is good and closer to 0.0 is poor.

$$F = 2 * \frac{Precision * Recall}{Precision + Recall}$$

- Overall Accuracy: Percentage of correctly identified code, given by:

$$Overall\ Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- ROC curve: In a Receiver Operating Characteristic curve the true positive rate is plotted in function of the FP rate for different points. Each point on the ROC curve represents a sensitivity pair corresponding to a particular decision threshold. A test with perfect discrimination has a ROC curve that passes through the upper left corner (100% sensitivity). Therefore the closer the ROC curve to the upper left corner, the higher is the overall accuracy of the test. Usually, ROC area higher (closer) to 1 is considered good, and closer to 0.0 is considered poor.

6 Supervised Learning Algorithms

6.1 The Naive Bayes (NB) Algorithm

The Naive Bayes algorithm (Kuncheva 2006) is one classification method based on conditional probabilities that uses a statistical approach to the problem of pattern recognition. Literature reports that it is the most successful known algorithms for learning to classify text documents, and further it is fast and highly scalable for model building and scoring reference. The idea behind a Naive Bayes algorithm is the Bayes' Theorem and the maximum posteriori hypothesis. Bayes Theorem finds the probability of an event occurring given the probability of another event that has occurred already. For instance, for a feature vector x with n attributes values $\mathbf{x} = \mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_n$, and a class variable C_j, $\mathbf{C} = \mathbf{c}_1, \mathbf{c}_2, \ldots \mathbf{c}_j$.

Bayesian classifiers can predict class membership C_j with probabilities $P(\mathbf{x}|C_j)$ for the feature vector x whose

distribution depends on the class C_j. The class C_j for which the probability is given by $P(C_j|x)$, is called the maximum posteriori probability that feature vector x belongs, and can be computed from $P(x|C_j)$ by Bayes' rule:

$$P(C_j|x) = \frac{P(x|C_j)\,P(C_j)}{P(x)} \quad (1)$$

It applies "naïve" conditional independence assumptions which states that all n features $x = x_1, x_2, \ldots x_n$ of the feature vector x are all conditionally independent of one another, given $P(C_j)$, and Naïve Bayes assumption is calculated as follows:

$$P(x|C_j) = P(x_1, x_2, \ldots x_n|C_j) = \prod_{i=1}^{n} P(x_i|C_j) \quad (2)$$

$$P(C_j|x) = \frac{P(C_j)\prod_{i=1}^{n} P(x_i|C_j)}{P(x)} \quad (3)$$

The most probable hypothesis given the training data 'Maximum a posteriori' hypothesis results in the following:

$$C_{max} = arg\,max_{C_m}\, P(C_j) \prod_{i=1}^{n} P(x_i|C_j) \quad (4)$$

Among data mining methods, Naive Bayes algorithm is easy to implement and is an efficient and effective inductive learning algorithm for machine learning. Figure 2 provides the overall accuracy rate for malware detection achieved through our experiments using Naive Bayes with k cross validations, k= {2,3,4,5,6,7,8,9,10}.

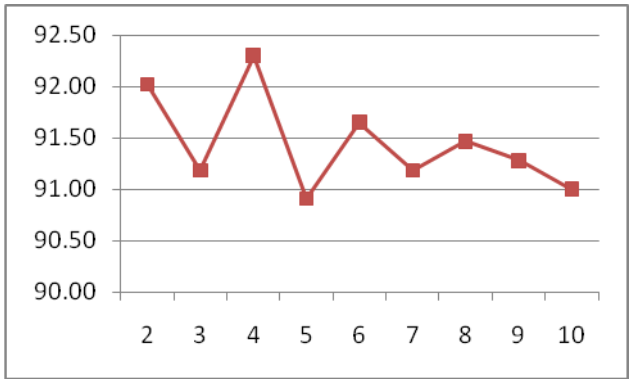


Figure 2 Performance of Naive Bayes (NB) with k cross validations (k=2 to 10)

6.2 The Sequential Minimal Optimization (SMO) Algorithm

Sequential Minimal Optimization (SMO) is a set of simple algorithms that can quickly solve the SVM QP problem, expand QP without any extra matrix storage and without using numerical QP optimization. The advantage of SMO is its ability to solve the Lagrange multipliers analytically. SMO is a supervised learning algorithm used for classification and regression, and it is a fast implementation of Support Vector Machines (SVM). The basic advantage is that it attempts to maximise the margin, for example the distance between the classifier and the nearest training datum. SMO constructs a hyperplane or set of hyperplanes in an n-dimensional space, which can be used for classification. Basically, a separation can be considered good when the hyperplane has the largest distance to the nearest training data points of any class, since in general the larger the margin the lower the generalization error of the classifier. SMO has been selected to classify malicious and benign executables because it is competitive with other SVM training methods such as Projected Conjugate Gradient "chunking", and in addition it is easier to implement in WEKA (Witten & Frank 2010).

As shown in figure 3, we have employed 4 different kernels; Radial Basis Function Kernel (RBF), Polynomial kernel, Normalized Polynomial kernel, and the Pearson VII function-based universal kernel (Puk), and the overall accuracy rate for malware detection achieved through Normalized Polynomial kernel is the highest for all the k cross validations, k={2,3,4,5,6,7,8,9,10}.

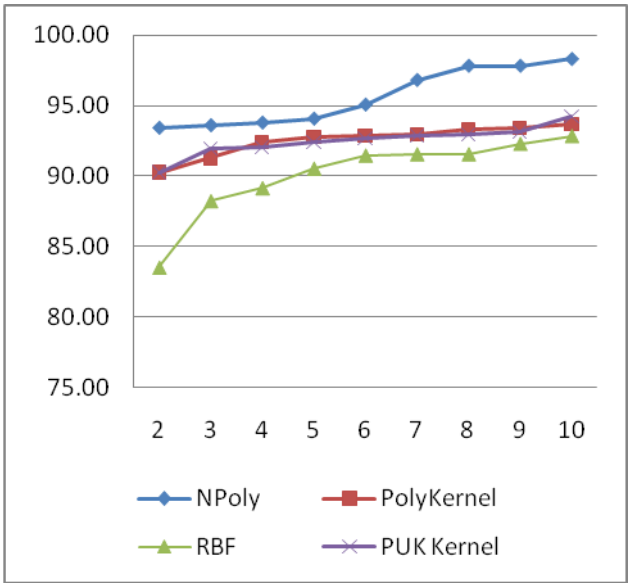


Figure 3 Performance of four Sequential Minimal Optimization kernels with k cross validations (k=2 to 10)

6.3 Artificial Neural Networks (ANN) Algorithm

Artificial Neural Networks (ANN) are biologically inspired form of distributed computing usually comprising of a set of nodes (including input, hidden and output) and weighted connections between them (Chen, Hsu, & Shen, 2005). Guo and Li (Guo & Li, 2008) define ANNs as a topology/architecture formed by organizing nodes into layers and linking the layers of neurons. The nodes are interconnected by weighted connections, and the weights are adjusted when data is presented to the network during a training process (Dayhoff & DeLeo, 2001)

A number of variations of neural networks are in use today in different applications, including in fraud detection. The use of ANNs in fraud detection spans almost all major forms of fraud including telecommunications fraud, financial fraud and computer

intrusion fraud among others (Kou, Lu, Sirwongwattana, & Huang, 2004). In fraud detection and anomaly detection, ANNs are fundamentally used as classification tools (Chandola, Banerjee, & Kumar, 2009). Usually, anomaly detection approach using neural networks involves two steps: training and testing. First, the network is trained on some part of the data to learn the different classes. Then, the remaining portion of the data is used to run the network to test accuracy and other performance indicators.

ANNs provide a non-linear mapping from the input space to the output space so that it can learn from the given cases and generalize the internal patterns of a given dataset (Guo & Li, 2008). Thus, ANNs adapt the connection weights between neurons and approximate a mapping function that models the training data provided for this purpose. Neural networks have the ability to learn distinct classes without knowledge of the data distribution (Chandola et al., 2009). However, most classifications rely on accurately labelled data which is often not readily available, especially for online banking and credit card fraud detection (Chandola et al., 2009). In Credit Card fraud detection, the FALCON system, which the developers claim to be in use by 65% of the credit systems worldwide, employs Neural Networks (FICO, 2010). Furthermore, VISA, Eurocard and Bank Of America (among others) use Neural technology in their Credit Card systems (Aleskerov, Freisleben, & Rao, 1997). The SAS fraud management system employs an ensemble of neural networks called Self Organizing Neural Network Arboretum (SONNA). Lastly, ACI's Proactive Risk Manager (PRM) also features a neural network in its architecture (IBM, 2008).

The downside to neural networks' distribution free generalisation is that they are prone to local minima and over-fitting (Bhattacharyya, Jha, Tharakunnel, & Westland, 2011). When the ANN is learning, a stopping condition may be declared as the anticipated net training error after a particular training session. This value is often a global minimum relative to the network's training errors. Sometimes, the ANN stops learning and gets stuck at a local minimum instead of the desired global minimum. This situation is most commonly referred to as the local minimum problem. Another problem with ANNs is hidden neuron saturation, where the hidden layer inputs are too high or too low such that the hidden layer output is almost close to the bounds of the activation function at that layer (Wang, etal, 2004). The other drawback with ANNs is their lack of adaptation to new data trends. At any point, ANNs will model only the data they have been trained on. This means that when a statistically different data pattern is introduced, the ANN will need to be re-trained or it may not correctly classify the new pattern. Consequently, this dictates that ANNs be retrained on a regular basis to keep up with emerging data trends. In online banking, ANNs are retrained after a defined period or after a certain number of examples have been collected.

Recently, classification method using a NN was used for Malware detection. Generally, the classification procedure using the NN consisted of three steps, data preprocessing, data training, and testing. In our experiments, the data preprocessing was performed as the feature selection stage. In the data training, the selected features from the data preprocessing step were fed into the NN, and the classifier was generated through the NN for classifying the data as either Malware or Benign. For the testing step, the classifier was used to verify the efficiency of NN. In the experiment, an error BP (Back Propagation) algorithm was used. The best-known example of a neural network training algorithm, namely back propagation was employed. Back propagation algorithm within neural network was used because of the large amount of input/output data and the overwhelming amount of complexity due to the fuzzy outputs. Figure 4 provides the overall accuracy rate for malware detection achieved through our experiments using Artificial Neural Networks with k cross validations, k= {2,3,4,5,6,7,8,9,10}.

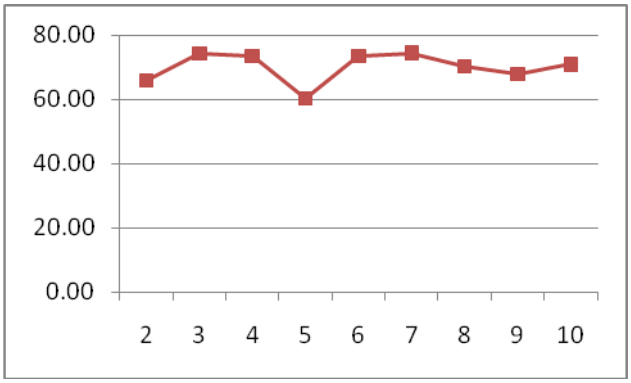


Figure 4 Performance of Artificial Neural Networks (ANN) with k cross validations (k=2 to 10)

6.4 J48 Algorithm

J48 classifier is a C4.5 decision tree used for classification purposes. In order to classify a new item, the classifier first needs to create a decision tree based on the attribute values of the available training data. So, whenever it encounters a set of items (training set) it identifies the attribute that discriminates the various instances most clearly. This feature that is able to tell the most about the data instances for classifying them the best is said to have the highest information gain.

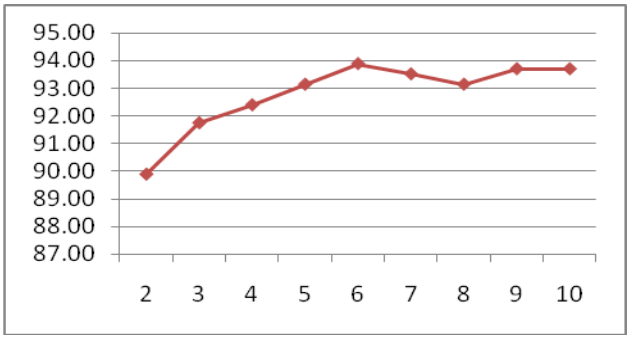


Figure 5 Performance of J48 with k cross validations (k=2 to 10)

Among the possible values of this feature, if there is any value for which there is no ambiguity, that is, when the data instances falling within its category have the same value for the target variable, then that branch is terminated and the target value arrived is assigned to it. Figure 5 provides the overall accuracy rate for malware

detection achieved through our experiments using J48 with k cross validations, k= {2,3,4,5,6,7,8,9,10}.

6.5 K-Nearest Neighbors (kNN) Algorithm

kNN is a simple supervised machine learning algorithm used for classifying objects based on closest training instants in the feature space. It has been used in many applications in data mining, statistical pattern recognition and many others. The object is classified based on a majority vote of its k nearest neighbors at closest distant from the object.

In our experiments, the K-nearest neighbors are compute as follows with K:

- Store all training samples x_j^j in memory.
- Determine the parameter K = number of nearest neighbors beforehand. (A good k can be selected using cross-validation for example).
- Measure the distance between the query-instance (x) and all the training samples x_j^j. (any distance algorithm can be used to) such as:

$$dis(x, x_i^j) \sqrt{\sum_{i=1}^{d} x(i) - x_i^j\,(i)^2} \qquad (7)$$

- Find the K-minimum distance between the query-instance (x) and each K

  $j_{min}^1, j_{min}^2, \cdots, j_{min}^k$.

- Get all categories of training data for the sorted value under K.
- Find the weighted distance of the query-instance (x) from each of the k nearest points as follows:

$$w = 1 - \frac{dis(x, x_i^j)}{\sum_{i=0}^{k} dis(x, x_i^j)} \qquad (8)$$

Figure 6 provides the overall accuracy rate for malware detection achieved through our experiments using K-Nearest Neighbors with k cross validations, k= {2,3,4,5,6,7,8,9,10}.

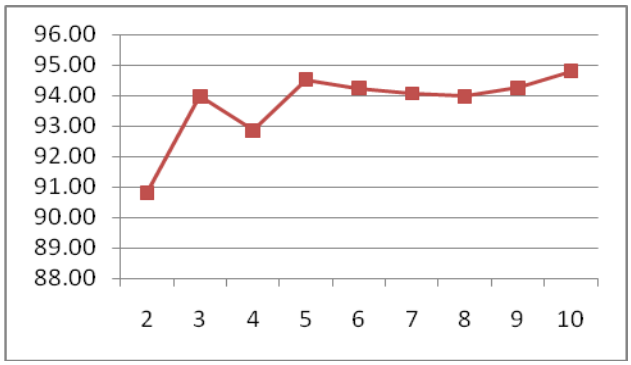


Figure 6 Performance of K-Nearest Neighbors (kNN) with k cross validations (k=2 to 10)

7 Results of the Study

The implementation involves employment of several software such as; WEKA version 3.6.4 software for performing the classification, and MatLab for feature selection. Table 3 shows the effectiveness of different data mining approaches. We had applied k cross validation, with k={2,3,4,5,6,7,8,9,10} for each of the data mining algorithms, and we observed that with k = 10 most of the algorithms provided the best accuracy. By comparing the evaluation measures achieved by each of the data mining techniques, we observe that SVM - Normalized PolyKernel has performed the best, and NN-Backpropagation exhibited the worst results. This could be attributed to the fact that and NN-Backpropagation follows a heuristic path and usually converges only to locally optimal solutions and can suffer from muliple local minima, while SVM - Normalized PolyKernel always finds a unique global minimum. Through our experimental analysis we found that SVM-Normalized Polynomial Kernel provided an average of 98.5% true positive rate. With 99% true detection rate of malware as malware, the average weight for the false alarm rate achieved was about 0.025 in this case. Overall, SVM-Normalized Polynomial Kernel had outperformed all other classification methods in all measures, namely, TP Rate, FP Rate, Precision, Recall, F-Measure and ROC Area.

8 Conclusion

Countermeasures such as antivirus detectors are unable to detect new malware and are in search of employing effective techniques, since the latest new malware adopt obfuscations to evade detection. With an exponential growth in unknown malware arising from innumerable automated obfuscations, there is a need to establish malware detection methods that are robust and efficient. In this paper, we have proposed and developed a machine learning framework using eight different classifiers to detect unknown malware and to achieve high accuracy rate. In this work, iterative patterns based on Windows API calls have been used and statistical measures have been adopted to further improve the classification results. Our experiments conducted on large malware datasets have shown very promising results achieving more than 98.5% accuracy rate.

Overall, the salient achievements of the research reported in this paper are:

- The proposed machine learning framework has resulted in high accuracies in malware detection. This is attributed to the unique feature selection of API sequences and the development of a fully-automated system used for evaluating data mining algorithms on large datasets of unknown malware.
- The proposed system is efficient as it uses filter approaches to be able to successfully detect malware with a smaller feature set. The term frequency of reduced API feature set using SVM (normalised poly kernel) has performed the best among the eight classifiers evaluated in this study.
- The system is signature-free and does not require knowledge or detailed study about the API sequence of execution to classify a malware.

TABLE 3 Results

		TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
J48		0.919	0.057	0.947	0.919	0.933	0.931	Malware
		0.943	0.081	0.913	0.943	0.928	0.931	Benign
	Weighted Avg.	**0.93**	**0.068**	**0.931**	**0.93**	**0.93**	**0.931**	
KNN		0.938	0.041	0.962	0.938	0.95	0.966	Malware
		0.959	0.062	0.933	0.959	0.946	0.966	Benign
	Weighted Avg.	**0.948**	**0.051**	**0.948**	**0.948**	**0.948**	**0.966**	
NB		0.913	0.094	0.915	0.913	0.914	0.94	Malware
		0.906	0.087	0.904	0.906	0.905	0.936	Benign
	Weighted Avg.	**0.91**	**0.09**	**0.91**	**0.91**	**0.91**	**0.938**	
NN - BackPropagation *(Worst results)*		0.983	0.301	0.82	0.983	0.894	0.744	Malware
		0.699	0.017	0.966	0.699	0.811	0.839	Benign
	Weighted Avg.	**0.864**	**0.183**	**0.881**	**0.864**	**0.859**	**0.783**	
SVM - Normalized PolyKernel *(Best results)*		0.99	0.018	0.982	0.99	0.986	0.981	Malware
		0.981	0.031	0.969	0.981	0.983	0.982	Benign
	Weighted Avg.	**0.986**	**0.025**	**0.976**	**0.986**	**0.984**	**0.982**	
SVM - PolyKernel		0.966	0.102	0.913	0.966	0.939	0.932	Malware
		0.898	0.034	0.96	0.898	0.928	0.932	Benign
	Weighted Avg.	**0.934**	**0.069**	**0.936**	**0.934**	**0.934**	**0.932**	
SVM - Puk		0.901	0.033	0.968	0.901	0.933	0.934	Malware
		0.967	0.099	0.898	0.967	0.931	0.934	Benign
	Weighted Avg.	**0.94**	**0.064**	**0.94**	**0.932**	**0.932**	**0.939**	
SVM- Radial Basis Function (RBF)		0.94	0.084	0.925	0.94	0.933	0.928	Malware
		0.916	0.06	0.932	0.916	0.924	0.928	Benign
	Weighted Avg.	**0.929**	**0.073**	**0.929**	**0.929**	**0.929**	**0.928**	

9 References

Alazab, M 2010, 'Static Analysis of Obfuscated Malware', in *Annual Research Conference*, Ballarat, p. 17.

Alazab, M, Layton, R, Venkatraman, S & Watters, P 2010, 'Malware Detection Based on Structural and Behavioural Features of API calls', in *the 1st International Cyber Resilience Conference*, Perth Western Australia, pp. 1-10.

Alazab, M, Venkatraman, S & Watters, P 2009, 'Effective digital forensic analysis of the NTFS disk image', *Ubiquitous Computing and Communication Journal*, vol. 4, no. 1, pp. 551- 8.

Alazab, M, Venkatraman, S & Watters, P 2010, 'Towards Understanding Malware Behaviour by the Extraction of API Calls', in *Cybercrime and Trustworthy Computing, Workshop*, Ballarat, pp. 52-9.

Bhattacharyya, S.; Jha, S.; Tharakunnel, K.; Westland, J.C. 2011 'Data mining for credit card fraud: A comparative study', *Decision Support Systems*, vol. 50, Issue 3, pp. 602-613

Bruschi, D, Martignoni, L & Monga, M 2006, 'Detecting Self-mutating Malware Using Control-Flow Graph Matching', in R Büschkes & P Laskov (eds), *Detection of Intrusions and Malware & Vulnerability Assessment*, Springer Berlin / Heidelberg, vol. 4064, pp. 129-43.

Chang, H & Atallah, M 2002, 'Protecting Software Code by Guards', in T Sander (ed.), *Security and Privacy in Digital Rights Management*, Springer Berlin / Heidelberg, vol. 2320, pp. 125-41.

Choi, S, Park, H, Lim, H-i & Han, T 2009, 'A static API birthmark for Windows binary executables', *Journal of Systems and Software*, vol. 82, no. 5, pp. 862-73.

Ferrie, P & Szor, P 2001, 'Zmist opportunities', *Virus Bulletin*, pp. 6–7.

Frawley, W, Piatetsky-shapiro, G & Matheus, C 1992, 'Knowledge discovery in databases: An overview', *Al Magazine*, vol. 13, no. 3, pp. 213–28.

Hand, DJ, Mannila, H & Smyth, P 2001, *Principles of data mining*, The MIT press.

Kolter, JZ & Maloof, MA 2006, 'Learning to Detect and Classify Malicious Executables in the Wild', *J. Mach. Learn. Res.*, vol. 7, pp. 2721-44.

Kuncheva, LI 2006, 'On the optimality of NaIve Bayes with dependent binary features', *Pattern Recognition Letters*, vol. 27, no. 7, pp. 830-7.

Linn, C & Debray, S 2003, 'Obfuscation of executable code to improve resistance to static disassembly', in *10th ACM conference on Computer and communications security* Washington, DC, USA, pp. 290-9.

Malan, DJ & Smith, MD 2005, 'Host-based detection of worms through peer-to-peer cooperation', paper presented to Proceedings of the 2005 ACM workshop on Rapid malcode, Fairfax, VA, USA.

MetaPHOR 2010, *W32.Simile*, Symantec Enterprise Security,, <http://www.symantec.com/security_response/writeup.jsp?docid=2002-030617-5423-99>.

Park, H, Choi, S, Lim, H-i & Han, T 2008a, 'Detecting code theft via a static instruction trace birthmark for Java methods', in *International Conference on Industrial Informatics*, Daejeon pp. 551-6.

Park, H, Choi, S, Lim, H-i & Han, T 2008b, 'Detecting Java Theft Based on Static API Trace Birthmark', in K Matsuura & E Fujisaki (eds), *Advances in Information and Computer Security*, Springer Berlin / Heidelberg, vol. 5312, pp. 121-35.

Perriot, F & Ferrie, P 2004, 'Principles and practise of x-raying', in *Virus Bulletin Conference* pp. 1- 17.

Rabek, JC, Khazan, RI, Lewandowski, SM & Cunningham, RK 2003, 'Detection of injected, dynamically generated, and obfuscated malicious code', paper presented to Proceedings of the 2003 ACM workshop on Rapid malcode, Washington, DC, USA.

RSA 2011, 'The Current State of Cybercrime and What to Expect in 2011', *RSA 2011 cybercrime trends report*.

Santos, I, Penya, YK, Devesa, J & Bringas, P 2009, 'N-grams-based file signatures for malware detection', in pp. 317–20.

Shankarapani, M, Kancherla, K, Ramammoorthy, S, Movva, R & Mukkamala, S 2010, 'Kernel machines for malware classification and similarity analysis', in *The 2010 International Joint Conference on Neural Networks*, Barcelona pp. 1-6.

Sharif, M, Yegneswaran, V, Saidi, H, Porras, P & Lee, W 2008, 'Eureka: A Framework for Enabling Static Malware Analysis', in S Jajodia & J Lopez (eds), *Computer Security - ESORICS 2008*, Springer Berlin / Heidelberg, vol. 5283, pp. 481-500.

Stolfo, S, Wang, K & Li, W-j 2005, 'Fileprints: Identifying File Types by n-gram Analysis', in *IEEE Workshop on Information Assurance United States Military Academy*, West Point, NY.

Sung, AH, Xu, J, Chavez, P & Mukkamala, S 2004, 'Static analyzer of vicious executables (SAVE)', in *20th Annual Computer Security Applications Conference,* , Tucson, AZ, USA, pp. 326-34.

Symantec Enterprise Security 1997, 'Understanding Heuristics: Symantec's Bloodhound Technolog', *Virus Bulletin*, vol. XXXIV.

Symantec Enterprise Security 2009, 'Symantec Global Internet Security Threat Report Trends for 2008', *Symantec Enterprise Security*, vol. XIV.

Symantec Enterprise Security 2010, 'Symantec Internet Security Threat Report: Trends for 2009', *Symantec Enterprise Security*, vol. XV.

Symantec Enterprise Security 2011, 'Symantec Internet Security Threat Report: Trends for 2010', *Symantec Enterprise Security*, vol. 16.

Tamada, H, Okamoto, K, Nakamura, M, Monden, A & Matsumoto, K-i 2006, 'Dynamic software birthmarks based on API calls', *IEICE Transactions on Information and Systems*, vol. 89, no. 8, pp. 1751-63.

Tang, K, Zhou, M-T & Zuo, Z-H 2010, 'An Enhanced Automated Signature Generation Algorithm for Polymorphic Malware Detection', *Journal of Electronic Science and Technology*, vol. 8, no. 2, pp. 114-21.

Venkatraman, S 2009, 'Autonomic Context-Dependent Architecture for Malware Detection', in *e-Tech 2009, International Conference on e-Technology*, Singapore, pp. 2927-47.

Venkatraman, S. (2010). Self-Learning Framework for Intrusion Detection, *Proceedings of The 2010 International Congress on Computer Applications and Computational Science (CACS 2010)*, 4-6 December, Singapore, ISBN 978-981-08-6846-8, pp. 517-520.

VX Heavens 2011, *VX Heavens Site*, retrieved 2/3 2011, <http://vx.netlux.org/>.

Wang, C, Pang, J, Zhao, R, Fu, W & Liu, X 2009, 'Malware Detection Based on Suspicious Behavior Identification', in *First International Workshop on Education Technology and Computer Science*, Wuhan, Hubei, China, vol. 2, pp. 198-202.

Witten, H & Frank, E 2010, *Data mining: Practical machine learning tools and techniques*, 3.6.4 edn, San Francisco, CA, <http://www.cs.waikato.ac.nz/ml/weka/>.

Yanfang, Y, Tao, L, Qingshan, J & Youyu, W 2010, 'CIMDS: Adapting Postprocessing Techniques of Associative Classification for Malware Detection', *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 40, no. 3, pp. 298-307.

Ye, Y, Wang, D, Li, T & Ye, D 2007, 'IMDS: intelligent malware detection system', paper presented to Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, San Jose, California, USA.

Irrigation Water Demand Forecasting – A Data Pre-Processing and Data Mining Approach based on Spatio-Temporal Data.

Mahmood A. Khan [1], Md. Zahidul Islam [2, 3], Mohsin Hafeez [1]

[1] International Centre of Water for Food Security, Charles Sturt University, Wagga Wagga 2678, NSW, Australia
[2] School of Computing and Mathematics, Charles Sturt University, Wagga Wagga 2678, NSW, Australia
[3] Centre for Research in Complex Systems (CRiCS), Charles Sturt University, Bathurst 2795, NSW, Australia

makhan@csu.edu.au, zislam@csu.edu.au, mhafeez@csu.edu.au

Abstract

World population is increasing at a fast rate resulting in huge pressure on limited water resources. Just about 3% of the earth's total water is freshwater that can be used for various applications including irrigation. Therefore, an efficient irrigation water management is crucial for the survival of human being. In our study area farmers need to order water based on their requirements. Once a request for water is made it typically takes about 7 days to get it at the farm gate from the upstream. Therefore, farmers need to estimate water requirement for the next 7 days in advance in order to get it at the farm gate on time. Currently there is no reliable tool available to the farmers of our study area for estimating future water requirement accurately. Hence, a water demand forecasting technique is crucial for the efficient use of available water.

In this study we first prepare a data set containing information on suitable attributes obtained from three different sources namely meteorological data, remote sensing images and water delivery statements. In order to make the prepared data set useful for demand forecasting and pattern extraction we pre-process the data set using a novel approach based on a combination of irrigation and data mining knowledge. We then apply a decision tree technique to forecast future water requirement. We also develop a web based decision support system for the managers, farmers and researchers in order to access various data including the prediction of possible water requirement in future. We evaluate our pre-processing technique by comparing it with another approach. We also compare our decision tree based prediction technique with a traditional prediction approach. Our experimental results indicate the usefulness of our pre-processing and prediction techniques.

Keywords: Demand forecasting, Data Mining, Decision Tree, Decision Support System, Water management, and Data pre-processing.

1 Introduction

Water availability plays an important role in agricultural. The world population is growing at a fast rate resulting in rising demand for household and irrigation water. Therefore, in the past decades irrigation water supply systems are under huge pressure in fulfilling the irrigation water requirements. Over 70% of the water in Australia is currently being used by agriculture (Khan et al. 2009). Since all the existing water resources are fully exploited and it is not possible to extract more water, the best alternative is to increase the water productivity.

For efficient irrigation water management, application of various hydrological models and data mining approaches has become crucial. Most of the water delivered for irrigation is not always efficiently used for crop production. On an average only 45% of the water is used by crop, 15% is lost during conveyance, 15% is lost in supply channels within the farms and the remaining 25% is lost due to inefficient water management practices (FAO 1994, Smith 2000). Therefore, it is evident that most of the water losses occur at farm level because of inefficient water management practices. In order to increase the water management efficiency a water demand forecast model can be useful.

There is a propagation delay for water to reach a farm from the original source in the upstream. Often the delay can be as long as 7 days, as it is the case for many farms in our study area. Therefore, to get water on time a farmer often needs to order water 7 days in advance. Since currently there is no reliable scientific tool for the farmers at our study area for estimating exact water requirement, a farmer relies on his/her experience for guessing the possible water requirement for the next 7 days. Hence, a farmer generally either overestimates or underestimates the water requirement. If the requirement is overestimated there will be on farm water loss, whereas if it is underestimated there can be adverse effect on the crop productivity. Therefore, having a reliable water demand forecast model can be useful for a farmer to estimate water requirement more accurately. The demand forecasting tool can also be useful for the irrigation managers for estimating water requirement for the whole irrigation area.

There are two major approaches for estimating water demand: i) conceptual and ii) system theoretical (Pulido-Calvo et al. 2009, Zhou 2002, Alvisi 2007). A conceptual model predicts the irrigation water requirement based on several factors including soil moisture, seepage, and evapotranspiration. Subsequently irrigation managers use these factors to estimate irrigation water demand for the whole season. However, water requirements estimated at the beginning of the irrigation season may not be the same as the actual water usage due to many reasons such as difference in expected and actual weather conditions and change in farming practices (Pulido-Calvo 2003).

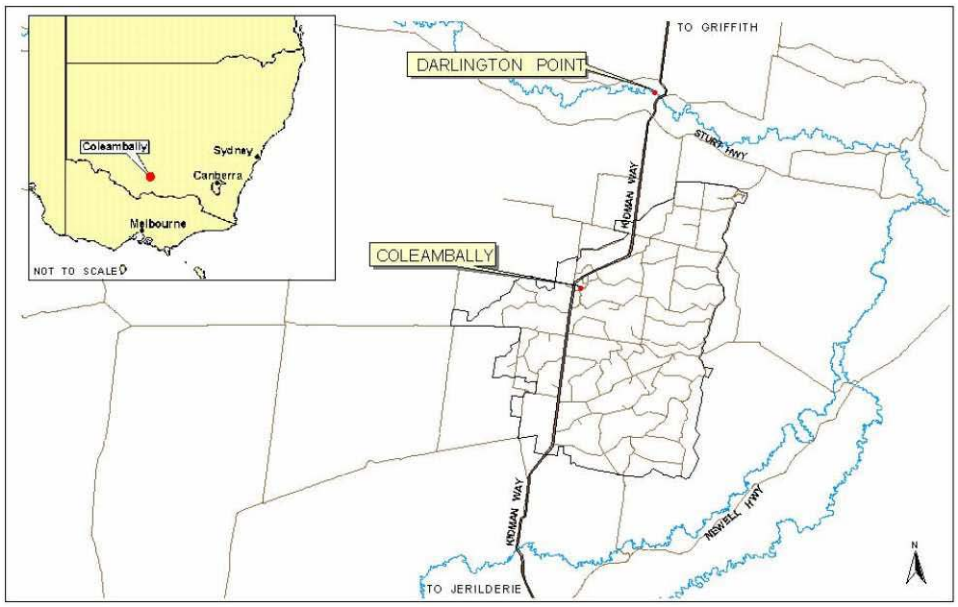


Figure 1: Location of Coleambally Irrigation Area (Source: CICL Annual Compliance Report, 2010)

The second approach for estimating water demand is known as system theoretical approach. In this approach a model is first trained on available data and then used for estimating future water demand. The system theoretical approach is more efficient and accurate than the conceptual approach (Pulido-Calvo 2008). Moreover, it can base on easily available data only.

To build an effective model using data mining techniques adequate historical data for the parameters such as crop water usage, crop type and weather conditions are required. A decision tree model can be useful for water demand forecasting. A decision tree (as shown in Figure 5) contains nodes and leaves, where each node in the tree tests an attribute and each leaf represents the value for the records belonging to the leaf (Han & Kamber 2001).

It recognises the relationship between the classifying (class) and the classifier (non-class) attributes. A class attribute is an attribute within the data set, which contains the values that are possible outcomes of the record. A decision tree analyses a set of records whose class values are known (Quinlan 1996). In other words, a decision tree explores patterns also known as logic rules from any data set (Islam 2010).

In this paper, we perform few interesting data pre-processing approaches using a combination of knowledge on irrigation engineering and data mining in order to improve the quality of our training data set. During the data pre-processing step we compare a few different approaches and finally adapt the most logical one. Our training data set contains attributes on various weather parameters (such as maximum and minimum temperature, wind speed, humidity, rainfall, and solar radiation), soil type, crop type (obtained from remote sensing satellite images) and water usage.

We then build a decision tree on the pre-processed training data set in order to extract existing pattern and predict future water demand for the next 7 days. The performance of the decision tree model is compared with a traditional way of estimating water demand using actual evapotranspiration (ET_c).

Our experiments indicate that as a result of the data pre-processing the quality of the training data set increases significantly. Moreover, the accuracy of water demand prediction made by the decision tree approach from the pre-processed data is higher than the accuracy of the traditional approach.

The demand forecasting technique is incorporated into our DSS called Coleambally IRIS (Integrated River Information System). Coleambally IRIS is a web based information system which stores information about time series, geospatial, climate and remote sensing data. It helps the users in decision making for sustainable irrigation water management. The paper is organised as follows: In section 2 we introduce our study area. Data collection, data pre-processing and the decision tree based demand forecasting technique are explained in section 3. Section 4 introduces our web based Decision Support System for managers, farmers and researchers. Experimental results are discussed in section 5. Finally Section 6 provides concluding remarks.

2 Study Area

The Coleambally Irrigation Area (CIA) is situated in the Riverina District of New South Wales which falls under Murrumbidgee River catchment as shown in Figure 1. CIA was developed in 1970 when the Water Conservation and Irrigation Commission acquired a large number of pastoral holdings to make use of water diverted from the Snowy Mountains Hydro-Electric Scheme. CIA contains approximately 79,000 hector of irrigated agriculture.

Coleambally Irrigation Cooperative Limited (CICL) was formed in the year 2000 after privatisation of irrigation corporations. Water in CIA is used to irrigate crops on 473 irrigation farms with an average size of 250ha. The main summer crops grown in CIA from November – April include rice, soybeans, maize (corn), grapes, prunes, sunflowers and Lucerne, whereas the winter crops grown from May – October include wheat, oats, barley, canola and Lucerne. Pasture for grazing is generally grown round the year.

Different soil types found in CIA are highly suitable for irrigated cropping, for example heavy clay is suitable for production of rice. Soil types such as, Self Mulching Clay (SMC) with a small portion of Sand are found at the northern areas of CIA, while Red Brown Earth (RBE) and Traditional Red Brown Earth (TRBE) are generally found in south as shown in Figure 3.

The climate of the area ranges from warm temperature with hot summers and mild winters. The climate averages obtained over the last 10 years from the Bureau of Meteorology (BoM) weather station located at Coleambally showed an average maximum temperature in January of 33.2°C and an average minimum temperature in July of 3.6°C. The long term average rainfall is 396 mm per year.

The surface water distribution for CIA from Murrumbidgee River is through the Main Canal and supply channels of length 477kms. Due to drought in the last decade, there has been a significant reduction in water allocations to the farmers highlighting the need to manage water demand and supply more sustainably in CIA.

According to the Annual Compliance Report of CICL (Coleambally Irrigation Company Limited, 2010), the average water allocation in 1995-2001 was 91% and was significantly declined to only 15% in 2006-2009. Due to declining water allocation and changing weather patterns, CIA requires new management measures for water use efficiency ranging from the farm (sub system) to the whole irrigation area (system) level. These measures can be enhanced by developing a model for irrigation water demand forecast which helps CIA farmers and managers use available water efficiently by irrigating right amount of water at the right time.

2.1 Existing water management practices in CIA

CICL is owned and operated by local farmers. It runs on a cooperative status. Its main purpose is to manage surface water distributions to the farms in CIA (Jackson 2009). CICL supplies water to all the farms through supply channels using a demand driven irrigation system. The water supply is managed every year based on the following rules.

1. Every farm has a predefined water entitlement which remains the same over the years. Based on the water availability in a particular year and the entitlement of a farm the allocation of water for the farm is determined.
2. Water is delivered by CICL to farmers upon their request.
3. Water ordering is made by the farmers based on their knowledge and the experience.

The CICL officers use their domain knowledge, experience and water order information placed by farmers to order water from the state water commission which supplies irrigation water to CIA. It is a common practice among the CIA farmers to place the orders with their initial plans on the crop type and the irrigation area at the beginning of a season. Often, the farmers revise their earlier decision (change their mind) after placing the orders, depending on water availability, market prices and many other socio economic factors. Currently in CIA there is no particular method or model to estimate the future water requirement, except the method using Evapotranspiration (ET). This method is generally used by the irrigation managers to estimate water demand. The water ordered by the farmers does not provide CICL with adequate information about the actual cropping area, which may result in inaccurate water ordering by the managers from the state water. It is also not mandatory for the farmers to provide information regarding the area under different crops. Due to the inaccurate estimate of water demand, shortage of water on the day of delivery can occur. An irrigation water demand forecast model can help overcome these difficulties by using remote sensing and meteorological data to ensure the optimum quantity and timely delivery of water for crops.

2.2 Irrigated Crop Area and Land Use in CIA

In 2009/10, rice and corn were the major summer crops covering 11.2% of the total irrigated area, while wheat, barley, canola and pasture being the main winter crops covering 65.8% of the total irrigated area. These crops have remained dominant since the establishment of CIA (CICL ACR, 2010). Table 1 illustrates the information about the crop area and the proportion of water used by them in 2009/10. Figure 2 shows the land use and land cover map obtained by remote sensing for the same year.

Crop	Area(ha)	Percentage of water delivered by CICL (%)
Rice	3668.5	46
Corn/Maize	1516	4
Wheat	10635	10
Barley	10499	7
Canola	2523	2
Pasture	6903	12
Others *	10995	19
Total	**46431**	**100**

Table 1: Crop area and percentage of crop water use for the year 2009/10 (Source: CICL Annual Compliance Report 2010)

* Lucerne, Triticale, soybean, sunflower, clover, prunes, grapes, vegetables etc.

3 Data Collection and Data Pre-processing

To build the training dataset, we collect data from three different sources. The first source is the water delivery statements that are obtained from CICL and provides us with the information about total water usage for a crop growing season by each farm. The second source is the meteorological data that are obtained from the installed weather stations in the study area. The third source is spatial data that are of two types a) Land Use Land Cover images, which provide us with the information about the crops grown and the cropping area as shown in Figure 2, b) soil type images that gives us the information about the different soil types associated with the farms in the study area as shown in Figure 3.

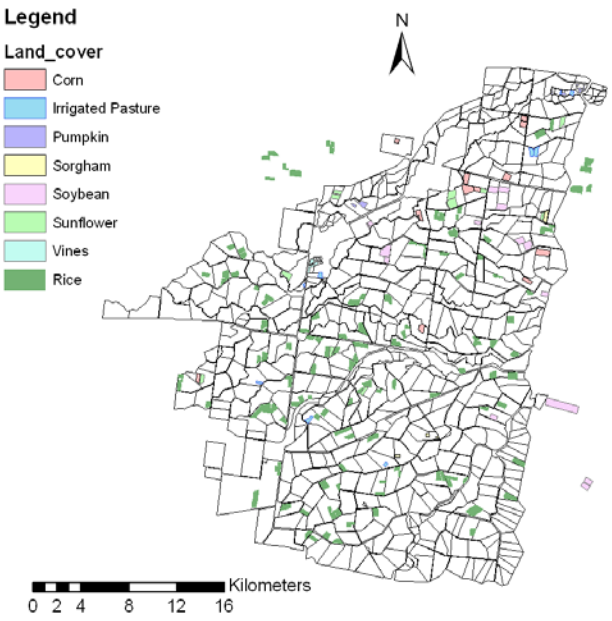


Figure 2: Land Use and Land Cover map of CIA for summer 2009/10

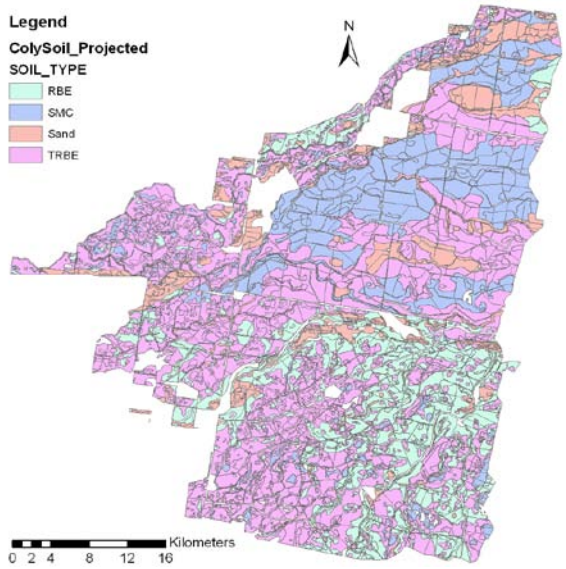


Figure 3: Different soil types of CIA

To run the decision tree algorithm, there is a strong need for data pre-processing to prepare good quality data. Data pre-processing takes approximately 80% of the total data mining effort (Zhang et al, 2003). It is also known that good results can be achieved by using data mining techniques/algorithms only if we have a good quality data set (Miksovsky et al, 2002). Often the real time data is very inconsistent as it contains many attributes which are not useful for our purpose and have some missing values. The purpose of data pre-processing is to remove noise from the data, extract and combine the required/relevant attributes from different data sources, make the data reliable and transform the data into our required format (Xu, 2003). By pre-processing the raw data, it is possible to prepare a good quality data set, which enables efficient and quality knowledge discovery (Zhang et al, 2003). Using the raw data obtained from three different sources we prepare our training data set as follows.

3.1 Attribute Selection

Based on our domain knowledge in irrigation, we select the attributes that have major influence on crop water usage. Moreover, we only select those influential attributes the data for which are easily available throughout the crop growing season. The selected attributes are various weather parameters (such as Maximum and Minimum Temperature, Wind speed, Humidity, Rainfall, and Solar Radiation), soil type, crop type and crop water usage.

3.2 Construction of data set

The data set was prepared from the historical data obtained from three different sources as discussed before. In this study we consider the data set as a two dimensional table where columns are attributes (categorical and numerical) and rows are records. Each record holds the daily average values of the corresponding attributes. Categorical attributes include soil type and crop type, whereas all other non-class attributes are numerical.

The water delivery statement only provides us with the information on the date and quantity of water supplied to a farm. Note that a farm does not take water supply every day. Instead it takes a specific volume of water on a day and uses the water for a period of time. Generally the farms have their own water storage facilities. The farmers can then order more water when they require. Therefore, from the water delivery statement it is not possible to estimate the exact amount of water usage for a particular day. However, our training data set contains records having daily average values of the non-class attributes. Each record represents information on a farm and a farm can have many records in the training data set. Hence, in order to obtain an accurate relationship between the non-class attributes and the class attribute (i.e. water usage) we need to store daily water usage for each record of the training data set.

We consider three possible techniques/approaches to estimate the daily water usage of a farm. We call the techniques as Equal Water Distribution, Averaging Out of the Parameters, and Reference Evapotranspiration Based Estimate. We explain the techniques as follows.

In equal water distribution technique we divide the volume of water delivered to a farm by the number of days between two consecutive deliveries. Therefore, we get an average water usage per day. However, if we divide the water usage evenly among the days then water usage remains same for each day regardless of weather conditions. Since crop water demand depends on the climatic conditions this approach does not appear to be a suitable one for this study.

In the second approach we take an average of both weather parameters and water usage for the days between the deliveries. Thus we convert the records representing the days between the deliveries into a single record having average values as shown in Figure 4. For example, let us assume that 100Mega Litre (ML) of water is delivered to a farm on the 1st of October and 400 ML of water is delivered on the 15th of October. In this approach we take the average weather parameters of the 15 records

and convert them into one record as shown in Figure 4. In this case the water usage of the record is 100/15 ML.

This approach appears to be a little better than the first one since we take average of the weather parameters along with the water usage. Therefore, water usage is not same among all the days having very different weather conditions. However, we introduce noise and also lose information due to the averaging out of the values. This is similar to the problem we usually face due to generalisation of data.

For example, let us consider two records R1 and R2. For R1 T-Max is 18^0 C and Humidity is 20%. For R2 T-Max is 38^0 C and Humidity is 80%. Let us also assume that the total water supply for the two days is 0.16 ML/ha. According to the first approach total water usage will be equally divided among the records. Both R1 and R2 will have 0.08 ML/ha water usage even though they have significantly different weather conditions. The second approach will merge the records into a new record say R3 having average values. R3 will have T-Max as 23^0 C, Humidity as 50% and Water Usage as 0.08 ML/ha. If the original water usage for R1 is 0.01 ML/ha and R2 is 0.15 ML/ha then both approaches appear to be unsuitable for extracting relationships between weather parameters and water usage. However, it is clear that the second approach preserves the relationship a little better than the first one.

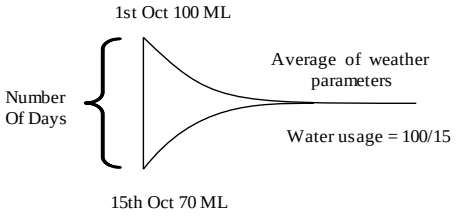


Figure 4: Averaging out of the parameters approach

Moreover, in the above example we assume that the water that was delivered previously has been used to irrigate the crops before the next delivery of water. However, in reality a farmer can actually order water while still having some water stored from the previous delivery. Similarly a farmer can also delay the water ordering. Therefore, dividing the whole amount of water that was delivered last time by the number of days between two deliveries may not get the right amount of water usage per day.

We come up with the third approach to resolve the issues with the first two approaches. In this approach we take Reference Evapotranspiration (ET_o) factor into consideration. Crop water usage can be calculated through Evapotranspiration (ET) which is the product of crop coefficient k_c and reference evapotranspiration (ET_o) (Al-Kaisi and Broner, 2009). Each crop has a constant crop coefficient value for a specific growth stage.

Let *n* be the number of days between two water deliveries for a farm, W_T be the amount of water delivered for the *n* days, and W_i be the water usage for the *i*th day. Note that W_T is the amount of water delivered at the beginning of the *n* days, and not the summation of two deliveries for *n* days. We obtain the daily ET_o values, for all *n* days, from our weather stations in the study area. We then calculate the coefficient x_i for the *i*th day, where $x_i = \frac{ET_o^i}{\sum_{j=1}^{n} ET_o^j}$. ET_o^i is the ET_o of the *i*th day. Finally W_i is calculate by multiplying x_i and W_T, i.e. $W_i = x_i \times W_T$.

There are several advantages of the third approach. Unlike the first approach here we do not use the same average water usage for the days having different weather conditions. Moreover, unlike the second approach it does not average out the weather parameters and water usage for the days in order to generalise the records into one. It estimates water usage, as accurate as possible, for each day and thereby uses each record of the training data set.

The final part of our data set is prepared by gathering the information from the spatial images for seasonal land use (cropping pattern) and to determine soil type of the farms. By using the spatial maps processed from satellite images we extract the crop type, cropping area and soil type information of every farm for a particular season.

In order to reduce the inconsistency of our data set we neglect the data from the farms having more than one soil type and consider only those farms with homogeneous soil type. Each record in our final data set holds daily average values of the weather parameters and crop water usage. However, values for crop type and soil type remain the same for whole growing season. Our data pre-processing technique uses a combination of the knowledge on data mining and irrigation engineering.

3.3 Application of a Decision Tree Algorithm on our Pre-processed Training Data Set on CIA

A decision tree algorithm is applied on the pre-processed data to extract the relationship between the non-class attributes and crop water usage. To generate a decision tree from our data set we consider crop water usage as the class attribute and all others as non-class attributes as shown in Table 2. Crop water usage is considered as a categorical attribute. While generating the decision tree, when an attribute is tested for a node if the attribute is numerical then there are two branches for the node (i.e. the data set is divided into two mutually exclusive horizontal parts). One branch contains all the records ">k" and the other contains all the records "<=k" of the data set, where k is a constant and it is one of the values of the attribute. However, if the attribute tested is categorical then there are *n* branches for the node, where *n* is the number of distinct values of that attribute.

We implement C4.5 algorithm to generate a decision tree on our pre-processed training data set. C4.5 algorithm takes a divide and conquers approach to build a decision tree from a training data set using information gain (Quinlan 1993).

We briefly introduce C4.5 algorithm as follows (Quinlan 1993, Islam 2010).

D – Whole data set. A two dimensional table where columns are attributes, and rows are records. Each record contains related information about the attributes.

T- An attribute with n number of mutually exclusive outcomes T_1, T_2……. T_n.

c - Number of classes i.e. domain size of the class attributes C.

p (D, j) - proportion of records in D belonging to the j^{th} class.

$D_i \subseteq D$ - the horizontal partition of the data set where all the records have T_i for the attribute T.

p (D_i, j) - proportion of records in D_i belonging to the j^{th} class.

|D| - size of the data set D.

$|D_i|$ - size of the partition D_i.

Step1: Entropy Calculation

The algorithm first calculates the entropy (a measure of the uncertainty associated with the class values of a set of records) of the whole dataset D using the following equation.

$$I(D) = -\sum_{j=1}^{c} p(D,j)\log_2\big(p(D,j)\big) \quad (1)$$

Gain Ratio Calculation for a Categorical Attribute T:

Step 2A: The algorithm then calculates entropy for a subset of the data set where all records have T_i for T as follows.

$$I(D_i) = -\sum_{j=1}^{c} p(D_i,j)\log_2\big(p(D_i,j)\big) \quad (2)$$

Step 3A: Weighted Entropy of the whole data set when attribute T is tested

$$I(D,T) = \sum_{j=1}^{n} \frac{|Di|}{|D|} \text{ X } I(Di) \quad (3)$$

Step 4A: Gain of an attribute T can be calculated by subtracting the weighted entropy from the total entropy of the dataset.

$$Gain(D,T) = I(D) - I(D,T) \quad (4)$$

Step 5A: Split Info of each attributed is calculated as follows.

$$SplitInfo(D,T) = -\sum_{j=1}^{n} \frac{|Di|}{|D|} \text{ X } \log_2\left(\frac{|Di|}{|D|}\right) \quad (5)$$

Step 6: Gain Ratio

$$GainRatio(D,T) = \frac{Gain(D,T)}{SplitInfo(D,T)} \quad (6)$$

Gain Ratio Calculation for a Numerical Attribute T:

When T is a numerical non-class attribute, the records are first rearranged so that all values of T are placed in ascending or descending order. Now the data set is horizontally divided into two parts, D_1 and D_2, based on a splitting point value *k* so that the domain of T in D_1 is [*l*, *k*], where *l* is the lowest value of the domain, and D_2 is [*k*+1, *u*], where *k*+1 is the next higher value to *k* and *u* is the upper value of the domain.

Step 2B: I(D,T) is calculated as follows.

$$I(D_i) = -\sum_{j=1}^{c} p(D_i,j)\log_2\big(p(D_i,j)\big), \text{for} 1 \le i \le 2 \quad (7)$$

$$I(D,T) = \sum_{j=1}^{2} \frac{|D_i|}{|D|} \text{ X } I(D_i) \quad (8)$$

The splitting point for which we achieve the minimum I(D,T) is considered as the best splitting point for T.

Step 3B: Gain can be calculated by subtracting the weighted entropy from the total entropy of the dataset.

$$Gain(D,T) = I(D) - I(D,T) \quad (9)$$

Step 4B: Split Info

$$SplitInfo(D,T) = -\sum_{j=1}^{2} \frac{|D_i|}{|D|} \text{ X } \log_2\left(\frac{|D_i|}{|D|}\right) \quad (10)$$

Step 5B: Gain ratio is calculated as follows.

$$GainRatio(D,T) = \frac{Gain(D,T) - (\log_2(N-1)/|D|)}{SplitInfo(D,T)} \quad (11)$$

where, N is number of distinct values of attribute T. Figure 5 shows a part of a decision tree obtained from our training data set.

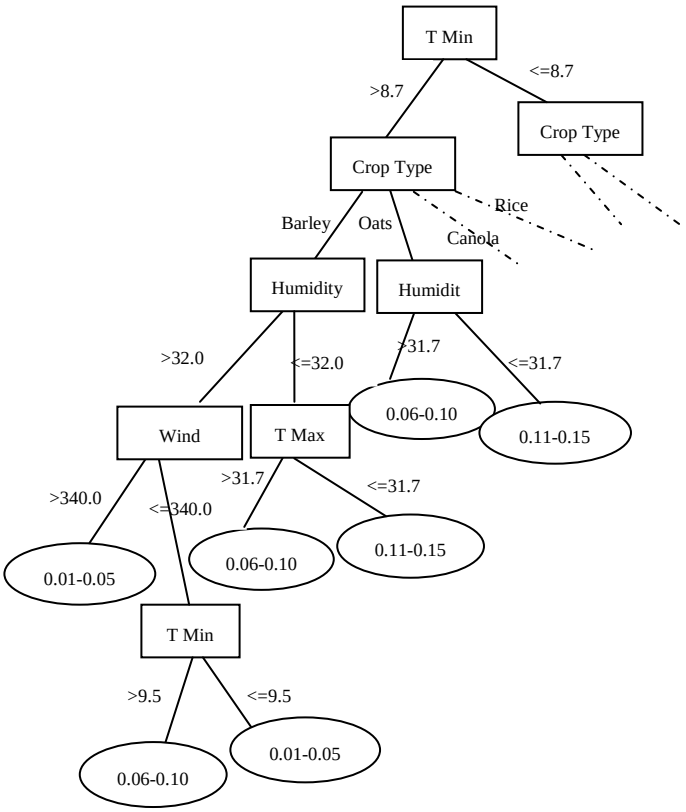


Figure 5 : Part of Decision Tree generated by C4.5 algorithm on our training data set

4 Demand Forecast Technique Implemented in our Web based Decision Support System (DSS)

The decision tree algorithm for irrigation water demand forecasting is developed in java. The demand forecasting model is incorporated in Coleambally IRIS DSS. Within the DSS, PHP (hypertext pre-processor) and java interacts with each other to generate a Decision tree and predict future water usage values from the decision tree. PHP calls the relevant java files which automatically generate a decision tree in order to perform a future prediction. The decision tree and the water demand prediction is displayed in the web pages. Description of the interaction between PHP and java is demonstrated by the conceptual diagram as shown in Figure 6.

Tmax (°C)	Tmin (°C)	Humidity (%)	Wind Speed km/day	Rainfall (mm)	Solar Radiation (MJ/m^2)	Soil Type	Crop Type	Crop Water Usage (ML/Ha/day)
18.1	3.8	80	122	0.2	9.5	SMC	Barley	0.01-0.05
16.4	6.7	48	481	0	16.6	RBE	Wheat	0.06-0.10
30.1	14.0	65	275	0.0	24.7	SMC	Rice	0.11-0.15
30.7	15.9	58	257	0.0	29.3	SMC	Corn	0.06-0.10

Table 2: Example of our training data set, Crop Water Usage is the Class attribute.

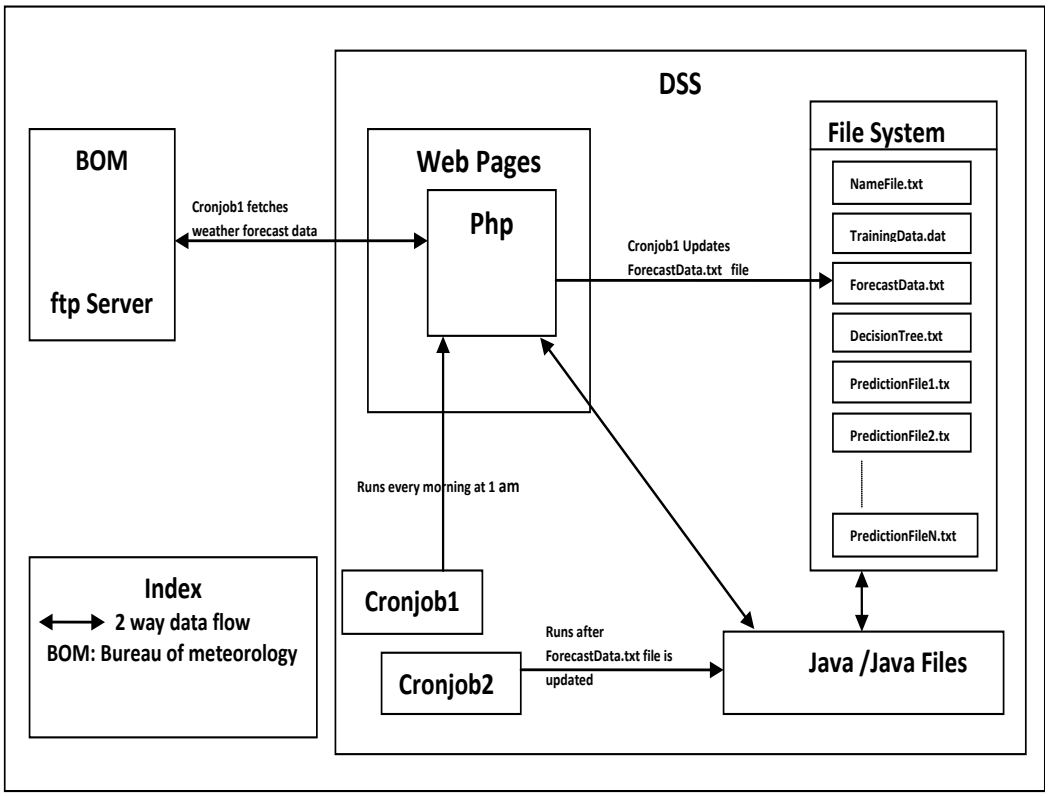


Figure 6: Conceptual Diagram Java-PHP Interaction

When the java code is invoked by PHP it connects to the file system in the DSS. To generate a decision tree, java requires two input files from the file system i) NameFile.txt and ii) TrainingData.dat.

NameFile.txt contains all information about the non-class and class attributes such as number of attributes, names of the attributes, types of attributes (numerical or categorical), possible values an attribute can have, and number of records in the training data set. TrainingData.dat contains the actual data on past records.

Every morning at 1am PHP code within the DSS automatically connects to the Bureau of Meteorology (BOM) ftp server through a scheduled cronjob1 (a cronjob is a command or a shell script which lets the users schedule jobs to run automatically at a certain time) as shown in Figure 6 and gets the weather forecast data for the next 7 days. It then replaces/overwrites only the weather data on the ForecastData.txt file in the file system, the other attributes in the ForecastData.txt file such as soil type and crop type remains same. When the contents of ForecastData.txt file is updated, the cronjob2 then executes the java code to generate the *i*th Predictionfile.txt for the *i*th farm where $1 \leq i \leq N$. The predictionfile contains the predicted values for the attribute 'crop water usage' along with all the other attributes. The prediction file will be generated for all the farms of CIA.

To generate the prediction files, java requires two input files from the file system they are i) DecisionTree.txt and ii) Forecastdata.txt. Java code first reads the DecisionTree.txt file to learn the pattern generated by the decision tree. In the second step it reads a record of the ForecastData.txt file and tests the record according to the decision tree to figure out the leaf where the record falls in. The class value of the leaf is considered as the predicted class value of the record. Thus java predicts the class value (water usage) of each record in the ForecastData.txt file.

To predict the water demand for a node (a set of a number of neighbouring farms), the water demand forecasted for the farms belonging to the node is accumulated. When a user wants to know the water

demand forecast for the next 7 days for his farm, the prediction file related to his farm is displayed on the web page. Similarly an irrigation manager can view the predictions for any individual farm, node and the whole CIA. Figure 7 shows a prediction file of a farm. Each row of the file contains all non-class attribute values and water demand prediction for the next 7 days.

Figure 7: Irrigation water demand forecast for 7days

The water demand forecast technique in Coleambally IRIS DSS will be useful for the farmers and irrigation managers of CIA. Farmers need to order the water from CICL. Once the order is made, it often takes 7 days to get the water at the farm from the upstream. Therefore, traditionally farmers need to guess future water demand (based on their past experiences) in order to request water for the next seven days. By using our web based demand forecast technique a farmer can learn more accurate water demand for his/her farm for the next 7 days in advance. Hence, a farmer can order more accurate amount of water they require resulting in a better water savings.

5 Results and Discussion

In order to evaluate our data pre-processing technique we build two training data sets D_1 and D_2. The data set D_1 is prepared taking the Equal Distribution approach and D_2 is prepared based on our Reference Evapotranspiration Based Estimate as explained in Section 3.2. In D_1we divide the water supply made on a particular day by the number of days between this delivery and the next water delivery. Moreover, in some farms there are more than one soil types. While preparing D_1we identify the soil type of the majority area of a farm and consider this as the value of the attribute "soil type" for the records representing the farm. However, in D_2 we calculate possible water usage of each day separately based on Reference Evapotranspiration coefficient as explained in Section 3.2. Therefore, we have more accurate water usage for each record in D_2. Moreover, in order to have more accurate data, in D_2 we use only records of those farms that have a single soil type. We realise that soil type has high influence on water usage and therefore accurate information on this attribute is crucial.

We divide D_1 into two data sets; training and testing data set. We build a decision tree on the training data set and then apply the tree on the testing data set in order to check the prediction accuracy of the tree. The accuracy check is carried out using 10 folds cross validation method. This is a method of testing the accuracy of the tree by dividing the data set into 10 equal parts that are also called as folds. Nine parts of the data set are used to train the tree and the one part is used to test the tree (Han and Kamber, 2001). This process is continued 10 times so that each part of the data set is used once for testing.

D_1 data set has 6070 records in total where 607 records are used for testing in each of the ten cross validations. Similarly, D_2 data set has 1500 records where 150 records are used for testing in each of the ten cross validations. In D_1 data set there are 4570 records representing farms having multiple soil types. These records are removed from D_2 for the purity of the data set.

First we perform 10 folds cross validation on D_1 data set. We have a low accuracy of around 43%. We then added another attribute called Reference Evapotranspiration, having high correlation with water usage, in D_1 in order to check whether it increases the accuracy. However, we find that the inclusion of the additional attribute does not increase the accuracy. We also perform a 10 folds cross validation on our D_2 data set and achieve significantly high accuracy of around 74% as shown in Table 3. The result clearly indicates the effectiveness of our data pre-processing based on irrigation engineering and data mining knowledge. Moreover, the result also indicates the appropriateness of the attributes selected using three different sources namely water delivery statement, meteorological data and remote sensing processed images obtained from satellite data.

We also compare the accuracy of the decision tree based prediction model with the accuracy of a traditional approach based on actual crop evapotranspiration (ET_c). Crop evapotranspiration ET_c is calculated using crop coefficient K_c (for a crop type and cropping stage) and reference evapotranspiration (ET_o). The empirical formula to calculate ET_c is $ET_c = K_c \times ET_o$ (FAO, 1998), and this is widely used globally to estimate water demand (Hunsaker et al 2005). The crop coefficient method was developed for the farmers and irrigation managers to calculate ET_c which helps them in making irrigation management decisions (Hunsaker et al 2005). The ET_c is the same as crop water usage (Al-Kaisi and Broner, 2009).

We use the data from the year 2007/08 and 2009/10 as training data set and the data from the summer season 2008/09 as the testing data set. Currently we do not have accurate data on water usage for the winter season of 2008/09. We build a decision tree on the training data set and use the tree to predict the class values on the testing data set. We also apply the ET_c based traditional approach to estimate the water usage of the records for the testing data set. The predicted class values (obtained by the decision tree approach and the traditional approach) are then checked against the actual class values of the testing data set.

Decision tree approach achieves significantly better accuracy than the accuracy of the ET_c based tradition approach. Table 4 shows a comparison among the actual water usage, water usage predicted by the decision tree approach and water usage predicted by the ET_c base approach for all the 22 nodes of CIA for the summer cropping season of 2008-09.

Folds	Total 6070 records using a pre-processing approach.			Total 1500 records using our pre-processing approach		
	Correctly classified records	Incorrectly classified records	Accuracy Percentage	Correctly classified records	Incorrectly classified records	Accuracy Percentage
1	186	421	31	111	39	74
2	212	395	35	108	42	72
3	176	431	29	107	43	72
4	254	353	42	110	40	73
5	302	305	50	118	32	79
6	236	371	39	110	40	73
7	311	296	51	110	40	73
8	289	318	48	109	41	73
9	337	270	56	112	38	75
10	303	304	50	108	42	72
Average:			**43**			**74**

Table 3: 10 folds cross validation on data sets based on two data pre-processing methods

Both demand forecast models are applied on all the farms of CIA to obtain the water demand for a whole cropping season. Finally the water requirement for each node is calculated by adding the water demand predicted for the farms belonging to the node.

Table 4 indicates that the predicted water usage values by decision tree approach are very similar to the actual water usage values. The prediction of the ET_c based approach is not as similar to the actual water usage as the prediction of decision tree based approach for the 22 nodes of the CIA. However, in few nodes such as "Coly 7", "Bundure_Main" and "Bundure 7_8", the actual water usage is significantly lower than the water usage predicated by the decision tree method. This is because only a few farms of the nodes were irrigating during the season. Moreover, the farms stopped irrigation for some reason half way through the season as it is evident from the water delivery statement. Moreover, "Coly 10" does not have any irrigation for the cropping season. We calculate the accuracy for each node as follows.

$$Accuracy = 1 - \frac{|Actual - Predicted\ Water\ Usage|}{Actual\ Water\ Usage} \times 100\%.$$

We find the average accuracy of the decision tree based approach and the traditional ET_c based approach as 89% and 74.5%, respectively. For the accuracy calculation we exclude the four exceptional nodes (Coly 7, Bundure_Main, Bundure 7_8 and Coly 10) where irrigation was not carried out for the complete cropping season. More interestingly out of 18 nodes (that were irrigated for the complete season) only one node ("Coly 11") has the water usage prediction made by our decision tree approach worse than the prediction made by the traditional ET_c approach.

Node	Actual Water Usage (ML)	Predicted Water Usage	
		Decision Tree (ML)	ET_c(ML)
Coly 1_2	407	344	284
Coly 3	1292	1103	777
Coly 4	800	746	570
Coly 5	879	945	666
Coly 6	4359	4158	3235
Coly 7	82	220.5	157
Coly 8	785	802	875
Coly 9	4501	4297	3232
Coly 10	0	0	0
Coly 11	2262	2877.5	2264
Tubbo	696	630	444
Boona 1	1201	1069	791
Boona 2	418	372	259
Boona 3	2438	2101	1652
Yamma Main	4299	3732	3098
Yamma 1	3333	3364	3085
Yamma 2_3_4	2926	3045	2772
Bundure Main	87	493	419
Bundure 3	763	768	653
Bundure 4	1597	1058	897
Bundure 5_6	961	798	677
Bundure 7_8	133	378	268.5

Table 4: Comparison of actual and predicted water usage made by decision tree and traditional method for all 22 nodes.

6 Conclusion

The main contributions of the paper are preparation of a data set, pre-processing of the data set, implementing a decision tree based demand forecasting technique, development of a web based decision support system for managers, farmers and researchers, and carrying out of necessary experiments. We discuss the contributions one by one as follows.

We prepare a data set by carefully selecting attributes from three different sources namely water delivery statement, meteorological data obtained from our weather stations installed in the study area, and remote sensing pre-processed images obtained from satellite data. For example, we obtain meteorological data on some attributes such as solar radiation, temperature and wind speed from our weather stations. We also obtain soil type and crop type data from pre-processed satellite images. Moreover, we obtain actual water usage data from Water Delivery Statement available from CICL. We carefully take irrigation engineering and data mining requirements into consideration for selecting relevant attributes that have high influence on water usage/demand.

Since we prepare the data set from different sources we do not have all data in a desired format. For example, while we have average meteorological data for each day we do not have any information on daily water usage on a farm. In fact farmers typically do not irrigate a farm on a daily basis. From CICL water delivery statements we only learn how much water was delivered to a farm during each delivery. Once delivered a farmer can store the water in on-farm supply channels and irrigate according to his/her requirement. There is no information on the actual daily water usage by a farm. However, in order to build a useful decision tree we need daily data on weather parameters and water usage so that the tree can extract meaningful the relationship between them.

We consider a few options in order to estimate the actual daily water usage of a farm. We finally devise a technique to determine daily water usage based on the reference evapotranspiration. We use the proportion of the reference evapotranspiration of a day to the total reference evapotranspiration of the days between two water deliveries in order to estimate the possible water usage by the day. To the best of our knowledge this is a novel approach to estimate water usage in order to pre-process a data set for data mining purpose.

We then apply a decision tree based water demand forecasting approach. We also compare the approach with traditional water demand forecasting technique that is globally used by the irrigations managers and engineers.

We incorporate the decision tree based demand forecasting technique into our web based Decision Support System (DSS) so that managers and farmers can get more accurate future water requirement information from the web. Our DSS automatically collects information from the weather stations and Bureau of Meteorology (BOM). It also automatically prepares a decision tree from the processed training data set and applies the knowledge on the future data set in order to predict future water requirement. Our DSS uses PHP, java and cronjob in a combination to perform the task.

We carry out necessary experiments in order to evaluate the effectiveness of our data pre-processing approach. Our experimental results indicate a significant improvement of accuracy in water demand forecasting based on the proposed pre-processing technique when compared to the accuracy obtained from other possible techniques. Based on a 10 fold cross validation we obtain 74% accuracy on our pre-processed data set compared to 43% accuracy on the other pre-processed data set.

Moreover, we compare the decision tree based future water requirement prediction approach with a traditional evapotranspiration based technique. We experiment on all 22 nodes (made of all 473 farms) of our study area and discover that while the traditional approach has 74.5% accuracy our decision tree based technique has 89% accuracy. Our approach obtains better prediction in all except one node.

Farmers need to order water depending on their requirements. Often it takes around 7 days to get water at the farm gate after the order was made, due to propagation delay from the upstream to the farm gate. Therefore, a farmer needs to estimate the requirement of the next 7 days and request the water in advance in order to get it on time. Currently there is no reliable scientific tool available to the farmers to make an accurate estimate of future water requirements. Therefore, a farmer estimates the future water requirement purely based on his/her experience. In most of the cases they either heavily overestimate or underestimate the water requirement having adverse effect on crop production. Hence, the data mining based tool developed in the paper is crucial for the farmers to make the maximum use of limited water resource.

7 References

Al-Kaisi, M.M. and Broner, I. (2009): Crop Water use and Growth Stages, Colarado State University, leatlet no.4.715

Alvisi, S., Franchini, M. And Marinelli, A. (2007): A short-term, pattern-based model for water-demand forecasting, *Journal of Hydroinformatics*, **9**(1), 35-50.

Bontemps, C. And Couture, S. (2002): Irrigation wtar demand for the decision maker, *Environment and Development Economics*, **7:**643-657.

Coleambally Irrigation Company Limited (2010): Annual Compliance Report.

Douglas J. Hunsker., Paul J. Pinter Jr. and Bruce A. Kimball. (2005): Wheat basal crop coefficients determined by normalized difference vegetation index, Irrig Sci, **24,** 1-14.

Han, J., & Kamber, M. (2001): Data Mining: Concepts and Techniques, A Horcourt Science and Technology company, 525 B Street, Suite 1900, San Diego, CA 92101-4495, USA.

Hoogenboom, G., Paz, O.J., Salazar, Melba. And Garcia A.G. (2009): Agriculture Irrigation Water Demand Forecast, Procedures for Estimating Monthly Irrigation Demands, http://www.nespal.org/sirp/waterinfo/state/awd/AgWaterDemand_IrrAmt_Detail.htm, accessed on 19/05/2011

Hu, X. (2003): DB-HReduction: A data Preprocessing Algorithm for Data Mining Applications, *Applied Mathematics Letters*.

Islam, M. Z. (2010): EXPLORE: A Novel Decision Tree Classification Algorithm, *the 27th International Information Systems Conference, British National Conference on Databases*, June 29- July 01, 2010, Dundee, Scotland.

Islam, M. Z., Barnaghi, P. M. and Brankovic, L.(2003): Measuring Data Quality: Predictive Accuracy vs. Similarity of Decision Trees, *In Proceedings of the 6 th International Conference on Computer & Information Technology (ICCIT 2003),* Dhaka, Bangladesh, **2:** 457-462.

J.Ross Quinlann. (1993) C4.5: *Programs for machine Learning*.Morgan Kaufmann Publishers, San Mateo, California, USA.

J.Ross Quinlann. (1996): Learning Decision Tree Classifiers, *ACM Computing Surveys*, **28:**1

Jackson, T. (2009): An appraisal of the on-farm water and energy nexus in irrigated agriculture, Ph.D. thesis, Charles Sturt university, Wagga Wagga, Australia.

Khan, S., Rana, T., Dassanayake, D., Abbas, A., Blackwell, J., Akbar, S., and Gabriel, H. F. (2009): Spatially Distributed Assessment of Channel Seepage Using Geophysics and Artificial Intelligence, *Irrigation and Drainage* **58**: 307 – 320.

Miksovsky, P., Matousek, K. And Kouba, Zdenek (2002): Data Pre-Processing Support for Data Mining, *IEEE SMC*.

Pulido-Calvo, I., Roldan, J., Lopez-Luque, R. and Gutierrez-Estrada, J.C. (2003): Demand Forecasting for Irrigation Water Distribution Systems. *Journal of Irrigation and Drainage Engineering* **129**(6):422-431.

Pulido-Calvo, I. and Gutierrez-Estrada, J.C. (2009): Improved irrigation water demand forecasting using soft-computing hybrid model. *Biosystems Engineering*, 102, 202-218.

Shichao Zhang, Chengqi Zhang and Qiang Yang (2003): Data preparation for data mining, *Applied Artificial Intelligence*, **17**:5-6, 375-381.

Smith, M. (2000): The application of climatic data for planning and management of sustainable rainfed and irrigation crop production. *Agricultural and Forest Meteorology*, 102, 99-108.

Zhou, S.L., McMohan, T.A., Walton, A. and Lewis, J. (2002): Forecasting operational demand for an urban water supply zone, *Journal of Hydrology*, **259**, 189-202.

Knowledge Discovery through SysFor - a Systematically Developed Forest of Multiple Decision Trees

Md Zahidul Islam[1] **Helen Giggins**[2]

[1] Center for Research in Complex Systems (CRiCS),
School of Computing and Mathematics,
Charles Sturt University,
Boorooma Street, NSW 2678, Australia.
Email: zislam@csu.edu.au

[2] School of Architecture and Built Environment,
Newcastle University,
Callaghan, NSW 2308, Australia.
Email: helen.giggins@newcastle.edu.au

Abstract

Decision tree based classification algorithms like C4.5 and Explore build a single tree from a data set. The two main purposes of building a decision tree are to extract various patterns/logic-rules existing in a data set, and to predict the class attribute value of an unlabeled record. Sometimes a set of decision trees, rather than just a single tree, is also generated from a data set. A set of multiple trees, when used wisely, typically have better prediction accuracy on unlabeled records. Existing multiple tree techniques are catered for high dimensional data sets and therefore unable to build many trees from low dimensional data sets. In this paper we present a novel technique called *SysFor* that can build many trees even from a low dimensional data set. Another strength of the technique is that instead of building multiple trees using any attribute (good or bad) it uses only those attributes that have high classification capabilities. We also present two novel voting techniques in order to predict the class value of an unlabeled record through the collective use of multiple trees. Experimental results demonstrate that *SysFor* is suitable for multiple pattern extraction and knowledge discovery from both low dimensional and high dimensional data sets by building a number of good quality decision trees. Moreover, it also has prediction accuracy higher than the accuracy of several existing techniques that have previously been shown as having high performance.

Keywords: Data Mining, Classification Algorithm, Multiple Decision Tree, Prediction Accuracy.

1 Introduction

Huge amount of data are being collected these days in almost every sector of life. Collected data are generally processed and stored as data sets so that various data mining techniques can be applied on them. In this study we consider a data set as a two dimensional table where rows are records and columns are the attributes. We also consider that a data set can have two types of attributes; numerical (such as Price and Temperature) and categorical (such as Employer's Name, and Country of Origin). Numerical attribute values have a natural ordering among them whereas categorical values do not exhibit any natural ordering.

Various data mining techniques are applied to these data sets to extract hidden information. For example, a decision tree algorithm is applied on a data set to discover the logic rules (patterns) that represent a relationship between various classifier (non-class) attributes and the class attribute [18, 19, 10]. A class attribute, which is often also known as the label of a record, is a categorical attribute such as "Diagnosis" in a data set having patient records. Each row/record of such a data set stores values of various classifier attributes for instance Age, Blood Pressure and Blood Sugar Level of a patient, and the class attribute (Diagnosis) of the record. Note that in this study we do not consider multi-label scenario where each record contains more than one class values. An example of a multi-label case can be a patient record having multiple diagnosis such as "Fever" and "Diabetes" at the same time. On the contrary, in this study we only consider that each record has a single class value associated with it.

A decision tree has nodes and leaves as shown in Figure 1. The rectangles are nodes and the ovals are leaves which are numbered from 1 to 3 in the figure. An attribute is tested at each node. If the attribute tested (i.e. the test attribute) at a node is numerical (such as the attribute A348) then typically there are two edges from the node. One of the edges is labeled "$> c$" while the other edge is labeled "$<= c$", where c is a constant which is an element of the domain of the test attribute. The value "c" is called the splitting point for the attribute. If the attribute is categorical then there are typically as many edges from the node as the domain size of the attribute, each labeled by a categorical value drawn from the attribute domain. The edges protruding from a node divide the data set into mutually exclusive partitions.

Each leaf of a tree has information on the number of records (of the leaf) belonging to a class value, for all class values. For example, in Leaf 1 (Figure 1) there are eleven records having Class 1 and one record

The work was supported by the 1st Author's CRiCS Seed Grant 2010.

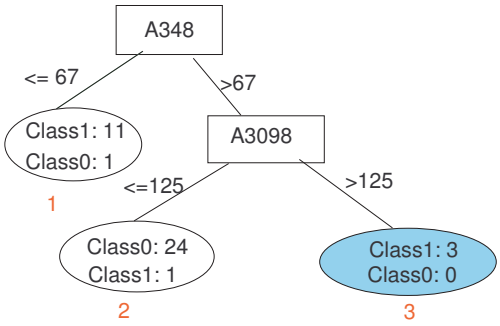


Figure 1: A decision tree obtained from the Central Nervous System data set

having Class 0. The class value of the maximum number of records of a leaf is called the majority class and other class values are called minority classes. If all records belonging to a leaf have the same class value then the leaf is called homogeneous, otherwise the leaf is called heterogeneous [13, 5]. In Figure 1, Leaf 3 is a homogeneous leaf and the other two leaves are heterogeneous. Each leaf has a logic rule that describes the test attributes and the split points (values) of the test attributes for the leaf. For example, the logic rule for Leaf 1 of Figure 1 is "A348 $\leq$ 67 $\Rightarrow$ Class1 (11) & Class0 (1)".

A decision tree is typically used for knowledge discovery as it extracts patterns (logic rules) from a data set. It can also be used to predict the class of an unlabeled record. For example, if an unlabeled record has attribute "A348" = 80 and "A3098" = 130 then from the tree shown in Figure 1 we can conclude that the record falls in Leaf 3 and therefore we can predict that the class value of the record is Class 1. The extraction of hidden pattern and the ability to make good predictions can be very useful in various decision making processes.

In a natural data set usually there are additional logic rules that are almost as good as the logic rules discovered by a single tree. Therefore, it is not unlikely to be able to build another good quality decision tree, which is different from the single tree generated by an algorithm such as C4.5 [18, 19] and Explore [10]. This is why we get different decision trees from a data set when we apply different algorithms.

The existence of multiple patterns is also evident from the experiments on privacy preserving data mining through noise addition [9, 12]. Controlled noise was added to a data set in such a way so that the gain ratios of all attributes tested at various nodes (of the single tree obtained from a data set) remain the same at the node levels even after noise addition. By gain ratio at the "node level" we mean the gain ratio, of the test attribute, for the horizontal data segment represented by the node. The tree shown in Figure 1 is built from a data set. Noise is added to the data set in such a way so that the gain ratio of the root attribute $A348$ remains unchanged for the entire data set, since a root node represents the whole data set. However, the gain ratio of the test attribute $A3098$ (tested by the node at Level 1) remains unchanged for the horizontal segment, represented by the node, where all records have $A348 > 67$ (see Figure 1).

It was expected to get a decision tree (from the perturbed data set) exactly the same as the decision tree from the original data set. However, the experiments frequently built trees that were different from the original tree. A possible explanation was that while adding a little amount of noise, although the gain ratio of the test attributes were preserved, gain ratios of some of the non test attributes were undeliberately improved to the level where they became even better than the original test attributes. Therefore, trees obtained from the perturbed data sets had different test attributes from the test attributes of the tree obtained from the original data set [9, 11]. Since only a very little amount of noise was added in the experiments, it is evident from the analysis that there are some attributes which are not tested in the original tree but still have almost as good gain ratios as the gain ratios of the attributes tested by the tree.

It was also noticed in the experimental results that the prediction accuracies of the trees obtained from the perturbed data sets were almost as good as the accuracy of the tree obtained from the original data set. Therefore, we argue that a data set can have multiple patterns, that can be extracted and represented by multiple trees. By multiple trees we mean a set of trees each having a set of logic rules as shown in Figure 1.

One obvious way to build multiple trees from a data set would be to apply a number of different single tree building algorithms on the data set. However, in that case a data miner needs to have access to a number of algorithms/softwares. He/she can only build as many trees as the number of algorithms he/she has access to. Therefore, with this approach a data miner can only build a limited number of trees. Moreover, the trees are not built systematically in order to extract multiple alternative patterns. Therefore, many of the trees can be very similar to each other. The existence of multiple patterns in a data set encourages us to systematically build multiple trees by deliberately choosing different test attributes for different trees. Multiple trees can be used collectively to achieve a better prediction accuracy and discover more logic rules/patterns.

In this paper we present a novel technique to systematically build a forest of decision trees using only those attributes having a high ability to classify a record. We apply our novel technique to build multiple trees on natural data sets. We also present two novel voting techniques in order to use the multiple trees collectively to predict the class values of unlabeled records. We implement several well known existing techniques and compare their prediction accuracy against SysFor. In Section 2 we discuss several existing techniques. Section 3 describes SysFor in more detail. Experimental results are presented in Section 4, and Section 5 presents concluding remarks.

2 Background Study

A technique called "Cascading and Sharing Trees (CS4)" [14, 15, 16] takes the number of trees to be generated as a user input. CS4 then orders the attributes by their gain ratios. It considers the ith best attribute as the root attribute of the ith tree. After the selection of the root attribute the data set is divided into mutually exclusive horizontal segments based on the values of the root attribute. For example, if the root attribute is numerical the data set is divided into two segments using the best split point of the root attribute, where all records in one segment have the values (of the root attribute) greater than the split point and all records in the other seg-

ment have values less than or equal to the split point. If the root attribute is categorical the data set is divided into segments by the values of the root attribute where all records in a segment have the same value for the root attribute, and records belonging to any two segments have different values.

A single tree building algorithm such as C4.5 [18, 19] is then applied on each segment to build the tree. A tree built on a segment is considered as a subtree of the ith tree. The root node of each subtree is therefore considered as a child node of the root node (of the ith tree) and thus all subtrees are combined to build the ith tree. CS4 builds the user defined number of trees only when the total number of non-class attributes of a data set is greater than or equal to the user defined number of trees. Otherwise, it builds as many trees as the number of non-class attributes in the data set. Each tree has a unique root attribute.

In order to predict the class value of an unlabeled record, CS4 uses all of the trees through a voting system [14, 15, 16], which we call *CS4 voting*. CS4 voting first calculates "coverage" of the leaves of each tree. Coverage of a leaf is the proportion of the records, in the leaf, having the majority class to the total number of records in the whole data set having the same class. For example, if a leaf has 20 records having the majority class say C_1 and the whole data set has 100 records having the class C_1 then the coverage of the leaf is 20%.

An unlabeled record falls in one and only one leaf of each tree. The majority class belonging to the leaf of a tree is the predicted class value of the unlabeled record according to the tree. Different trees may predict different class values. CS4 voting groups the trees such that trees belonging to the same group give the same prediction for the record, and trees belonging to different groups give different predictions. For an unlabeled record CS4 voting then sums up the coverage values of the leaves (belonging to the trees of a group) that the record falls in. The sum of the coverage values is considered as the total vote in favour of the class value that the group predicts. Similarly, sum of the coverage values are calculated for each group. The group having the highest sum wins. Therefore, the class predicted by the winning group is considered as the predicted class value for the record.

We argue that CS4 has several limitations. For example, it can only build (at most) as many trees as the number of non-class attributes of a data set. Therefore, for a low dimensional data set it is unable to build many trees. Another limitation is that it may build some poor quality trees since it uses the ith best attribute for the ith tree, regardless of the classification ability of the attribute in terms of gain ratio. If the ith best attribute has very low gain ratio we may get a poor quality tree having low accuracy and misleading logic rules.

In 2009 Abellan and Masegosa proposed a multiple tree building technique called "An Ensemble method using credal decision trees (Credal)" [3] which is very similar to CS4 except that it uses Imprecise Info-Gain (IIG) [1, 2] based ranking of the attributes as opposed to the Information Gain based ranking. It also has a slightly different voting system to predict the class value of a new record. From each tree it calculates a vote for each class value. That is for each tree it first identifies the leaf where the record falls in and calculates the proportion of records supporting a class value over the total number of records in the leaf. This proportion is considered as the vote for the class value from the tree. Similarly, vote for the class value is calculated from each tree and finally all votes for the class value are added to get the final vote. Final votes for all class values are also calculated. The class value having the highest final vote is considered as the predicted class for the record.

Another multiple tree building technique called "Maximally Diversified Multiple Decision Tree Algorithm (MDMT)" [7, 8] attempts to build multiple trees where an attribute is tested at most in one tree. Each tree tests a completely different set of attributes than the set of attributes tested in any other tree. MDMT builds the first decision tree using a traditional algorithm like C4.5 on a data set. All non-class attributes that have been tested in the first tree are then removed from the data set and C4.5 is again applied on the modified data set to build the second decision tree. The process continues until either the user defined number of trees are generated or all non-class attributes of the data set are removed. We argue that for a low dimensional data set MDMT may not be able to build sufficient number of trees. For a high dimensional dataset the technique may be able to build many decision trees. However, the quality of the later trees may not be high due to the removal of good attributes from the data set every time a tree is generated. A later tree can be generated from a data set having only attributes with poor gain ratios. Therefore, the technique may suffer from poor prediction accuracy as a result of insufficient number trees generated from a low dimensional data set and/or some poor quality trees generated.

In order to predict the class value of an unlabeled record, MDMT uses all generated trees through its voting system [7, 8], which we call the *MDMT voting*. The accuracies of the trees that have the same predicted class value for an unlabeled record are added and the result is considered as the vote in favour of the class value. Similarly, vote for each class value is calculated. The class value having the highest vote is considered as the final prediction for the record. MDMT uses the overall prediction accuracy of a tree in order to take into account the possible low quality of the later trees. We argue that a tree can have a high overall accuracy but a record can still fall in a leaf having high inaccuracy (i.e. having big proportion of minority records) resulting in a weak prediction for the record. Similarly, a tree can have a low overall accuracy but a record can still fall in a big leaf having high accuracy meaning a good confidence and support for the rule. Therefore, taking the accuracy of a tree (instead of the accuracy of a leaf) as an indicator of the prediction quality may not be a good idea. However, experimental results [7, 6] on several data sets indicate a superiority of CS4 and MDMT over several existing techniques including Random Forest, AdaBoostC4.5, SVM method, BaggingC4.5 and C4.5.

3 Our Technique

We now describe our novel technique to build *SysFor*: A Systematically Developed Forest of Multiple Decision Trees from a data set. We use the following steps to build a SysFor.

Step 1: Find a set of "good attributes" and corresponding split points based on a user defined "goodness" threshold and "separation" threshold. For a numerical attribute, the same attribute can be chosen more than once in the set of good attributes if the numerical attribute has high gain ratios with more than one different split points that are well separated and not adjacent to each other.
Step 2: If the size of the set of good attributes is less than a user defined number of trees then choose

each good attribute one by one as the root attribute (at Level 0) of a tree, and thereby build as many trees as the number of good attributes. Else, build user defined number of trees by considering as many good attributes as the number of trees.
Step 3: If the number of trees built in Step 2 is less than the user defined number of trees, then build more trees (until the user defined number of trees are built or the maximum number of possible trees are built) by using alternative good attributes at Level 1 of the trees built in Step 2. The alternative good attributes are chosen from the set of good attributes for the nodes at Level 1 of the trees built in Step 2.
Step 4: Return all trees built in Step 2 and Step 3 as a SysFor.

The whole process of building SysFor is introduced in Algorithm 1, and a couple of supplementary algorithms Algorithm 2 and Algorithm 3. Algorithm 1 takes (as user input) a data set D_f, user defined number of trees N, "goodness" threshold θ, "separation" threshold α, minimum gain ratio R, pruning confidence factor C_F and minimum number of records in each leaf N_L. The last three inputs are used by C4.5 [18, 19] for building a single tree. We now explain the steps as follows.

The data set D_f has a number of non-class attributes and a class attribute. In Step 1 we first calculate the gain ratio [19, 10], for the whole data set, of each non-class attribute one by one. The non-class attributes can be categorical or numerical. If a non-class attribute is categorical we then add the attribute and its gain ratio in two sets A^s and G^s, respectively. Since (unlike a numerical attribute) a categorical attribute does not have a numerical split point we add a negative number (say -100) for the categorical attributes in a set of split points P^s. On the other hand, for a numerical non-class attribute A_i we first calculate the best gain ratio and store it in a set of chosen gain ratios g^c. We also store the corresponding split point in a set p^c. After adding a split point in p^c, we next explore all available split points for the attribute such that any available split point p_l maintains $\frac{abs(p_l - p^c_m)}{|A_i|} > \alpha, \forall p^c_m \in p^c$. Among all available split points, the split point having the highest gain ratio is again added in the set p^c. Also the highest gain ratio is added in the set g^c (see Algorithm 2).

We continue the process of adding gain ratios and corresponding split points in g^c and p^c, and calculating available split points recursively until we have at least one available split point. Note that as the size of p^c grows the size of the set of available split points shrinks. The gain ratios in g^c and corresponding split points in p^c are then added in the set G^s and P^s. We also add the numerical attribute A_i in the set A^s as many times as the size of p^c. We continue the process of adding values in A^s, G^s and P^s for all non-class attributes. After the values are added the set G^s is sorted in a descending order of the gain ratios, and A^s and P^s are rearranged accordingly. All the gain ratios in G^s that have differences from the best gain ratio (i.e. the gain ratio stored in the first element $G^s[0]$) greater than a user defined goodness threshold θ are removed from G^s. A^s and P^s are also reorganised accordingly by removing the entries for which corresponding values in G^s have been removed. A^s is therefore the set of good attributes, G^s and P^s are the sets of gain ratios and split points of the good attributes. Note that with different split points a numerical attribute can appear more than once in the set of good attributes. However, the split points have to be different enough to satisfy the separation threshold requirement as explained before.

In Step 2, we pick the attributes and corresponding split points one by one from the set of good attributes and set of split points. Based on the gain ratios we first choose the best attribute, then the second best and so on. If the attribute picked is categorical then the whole data set is divided into as many mutually exclusive horizontal segments as the domain size, where all records in a segment have the same value for the attribute and records belonging to different segments have different values. If the attribute is numerical the data set is divided into two horizontal segments based on the split point. A gain ratio based decision tree algorithm such as C4.5 is then applied on each horizontal segment to build a tree. The root node of each of these trees is then joined as a child with the attribute picked from the set of good attributes as the parent. Thus we build a tree having a good attribute as the root (see Algorithm 3). If the number of good attributes is less than the user defined number of trees then choose each good attribute one by one as the root attribute (at Level 0) of a tree, and thereby build as many trees as the number of good attributes. Otherwise, build user defined number of trees by considering the necessary number of best attributes (from the set of good attributes) one by one as the root attributes.

If the number of trees built in Step 2 is less than the user defined number of trees, then SysFor algorithm builds more trees in Step 3 until the user defined number of trees are built or the maximum number of possible trees are built (see Algorithm 1). The algorithm first uses the root attribute of the first tree built in Step 2 in order to divide the whole data set in horizontal segments. It then prepares a set of good attributes, a set of corresponding split points and a set of gain ratios for each horizontal segment. Based on the numbers of good attributes of all segments a number of possible trees t_p is calculated using $t_p = \frac{\sum_{j=1}^{|\overline{D_f}|} |A^g_j| \times |D_j|}{\sum_{j=1}^{|\overline{D_f}|} |D_j|}$, where $|\overline{D_f}|$ is the number of data segments, $|D_j|$ is the number of records in the jth segment, and A^g_j is the set of good attributes for the jth segment. Note that t_p is the weighted average of the number of good attributes of the segments. Therefore, some segments may have number of good attributes less than or equal to t_p while other segments having number of good attributes greater than t_p.

If the number of good attributes of a segment is greater than or equal to t_p, then t_p number of trees are built from the segment by using the alternative good attributes for the segment. However, if number of good attributes is less than t_p then as many trees as the number of good attributes are first built using alternative good attributes. All remaining trees are then built using the best attribute from the list of good attributes. A tree is built at a time from a segment, for all segments and the trees are joined by connecting their roots (as children) to the root of the first tree built in Step 2. Therefore, the algorithm finally builds t_p number of trees having the root attribute same as the root attribute of the 1st tree built in Step 2. The algorithm then uses the root attribute of the 2nd tree built in Step 2 in order to build more trees using the good attributes at Level 1 of the 2nd tree. The process of building more trees continues until the requested number of trees are built or all trees built in Step 2 are used up.

Once SysFor is built we can use any voting system such as CS4 and MDMT voting (as explained in

Section 2) in order to predict the class values of unlabeled records. In this study we also present two novel voting systems for SysFor called SysFor Voting-1 and SysFor Voting-2. We now explain them one by one as follows.

For each new record SysFor Voting-1 first finds (from all trees) the list of leaves (logic rules) L that the record belongs to. There will be only one such leaf from each tree. We find the set of leaves $L^{100} \subseteq L$ where a leaf $L_i \in L^{100}$ has the following properties. The leaf L_i has 100% accuracy i.e. all records belonging to the leaf have the same class value. Moreover, the leaf has support greater than a user defined "Support threshold" or $\frac{|D_f|}{|N|} \times RulePercentage$, where $|D_f|$ = total number of records in the data set, $|N|$ = number of leaves of the tree. The Support threshold and Rule Percentage are user defined values. Default values of Support threshold and Rule Percentage can be chosen as 5 and 0.10, respectively. For each class value we take a voting among L^{100}. If the class attribute has a domain size $|C|$ then for a class value $C_i \in C$ we calculate the vote $Vote(C_i) = \sum_{j=1}^{|L^{100}|} \frac{|R_j^{C_i}|}{|R_j|}$, where $|R_j^{C_i}|$ is the number of records having C_i class value in the jth rule, and $|R_j|$ is the total number of records in the jth rule. If the $|L^{100}| = 0$ then for each class value we take a voting among all leaves L, instead of L^{100}, as $Vote(C_i) = \sum_{j=1}^{|L|} \frac{|R_j^{C_i}|}{|R_j|}$. The class value having the highest vote is the final prediction for the record.

In SysFor Voting-2 we take an aggressive approach where in order to predict the class value of a new record we first find all the leaves L (from all trees) that the record falls in. We then determine the leaf $L_i^{max} \in L$ that has the highest accuracy. Finally, the majority class value of L_i^{max} is considered as the predicted class value of the new record. If there are more than one leaves having the highest accuracy then the leaf having the highest support (number of records) among the leaves with the highest accuracy is considered to be the winner.

4 Experimental Results

We first apply our algorithm on a data set called Contraceptive Method Choice (CMC) available from UCI machine learning repository [17]. CMC data set has 1473 records, each record having 7 categorical non-class attributes, 2 numerical non-class attributes and one categorical class attribute. The class attribute "Contraceptive Method Used" has three categorical values, 1, 2 and 3. In this experiment we consider that the user defined number of trees is 60. We also use Goodness threshold = 0.3, and Separation threshold = 0.3. Moreover, for C4.5 (while used within our algorithm and individually) we assign Confidence factor = 25%, Minimum Gain Ratio = 0.01 and Minimum number of records in each leaf = 10. For SysFor Voting-1 we use Support threshold = 5 and Rule Percentage = 0.10. We apply CS4 [14, 15, 16] and MDMT [7, 8] on the data set with user defined number of trees equal to 60. We also build a decision tree from the data set by using C4.5 algorithm [18, 19].

The main purpose of multiple tree generation by SysFor is to perform a better knowledge discovery through the extraction of high quality multiple patterns. They are likely to give a better insight into the data set and therefore be useful in various decision making processes. In order to evaluate the quality of

Input: D_f, N, θ, α, R, C_F, N_L
Output: a set of trees T

initialize a set of decision trees T to null;
initialize a set of good attributes A^g to null;
initialize a set of split points P^g to null;
set A^g, and P^g by calling
GetGoodAttriubtes(D_f, θ, α);
initialise i to 1;
while $|T| < N$ *and* $|T| < |A^g|$ **do**
  T_i = **BuildTree**(D_f, $A_i \in A^g$, $P_i \in P^g$, R, C_F, N_L);
  $T = T \cup T_i$;
  $i = i + 1$;
end
$i = 1$;
initialise K to $|T|$;
while $|T| < N$ *and* $i \leq K$ **do**
  `/* divide the data set` D_f. `*/`
  if A_i *is categorical* **then**
    $D_f = \{D_1, D_2, \ldots D_{|A_i|}\}$;
  end
  if A_i *is numerical* **then**
    $D_f = \{D_1, D_2\}$;
  end
  `/*` $|\overline{D_f}|$ `= number of data segments in` D_f, $|D_j|$ `= number of records in the` j`th segment */`
  for $j = 1$ *to* $|\overline{D_f}|$ **do**
    initialise A_j^g, and P_j^g to null;
    A_j^g, and P_j^g =
    GetGoodAttributes(D_j, θ, α);
  end
  initialise number of possible trees, t_p to null;
  calculate, $t_p = \frac{\sum_{j=1}^{|\overline{D_f}|} |A_j^g| \times |D_j|}{\sum_{j=1}^{|\overline{D_f}|} |D_j|}$;
  initialise x to 1;
  while $|T| < N$ *and* $x \leq t_p$ **do**
    for $j = 1$ *to* $|\overline{D_f}|$ **do**
      if $|A_j^g| > x$ **then**
        t_j = **BuildTree**(D_j, $a_{x+1} \in A_j^g$, $p_{x+1} \in P_j^g$, R, C_F, N_L);
      end
      if $|A_j^g| \leq x$ **then**
        t_j = **BuildTree**(D_j, $a_1 \in A_j^g$, $p_1 \in P_j^g$, R, C_F, N_L);
      end
    end
    build a tree T_{new} by joining the root node of each t_j $(1 \leq j \leq |\overline{D_f}|)$ as a child node with the root node of T_i;
    $T = T \cup T_{new}$;
    $x = x + 1$;
  end
  $i = i + 1$;
end
return T;

Algorithm 1: Building a SysFor.

Input: D_f, θ, α
Output: A^s, P^s
GetGoodAttributes(D_f, θ, α)

initialise a set of attributes A^s to null;
initialise a set of split points P^s to null;
initialise a set of gain ratios G^s to null;
for *each attribute* A_i *in* A **do**
  /* A is the set of all non-class attributes */
  if A_i *is categorical* **then**
    calculate gain ratio of A_i, G_{A_i};
    add A_i into A^s;
    add -100 into P^s;
    add G_{A_i} into G^s;
  end
  if A_i *in numerical* **then**
    initialise a set of chosen split points p^c to null;
    initialise a set of chosen gain ratios g^c to null;
    create a set of available split points p^a of A_i;
    initialise a set of gain ratios g^a to null;
    calculate the gain ratio for each split point in p^a;
    add the gain ratios in g^a;
    while p^a *is not null* **do**
      find the highest gain ratio and corresponding split point from g^a and p^a, respectively;
      add the highest gain ratio and corresponding split point into g^c and p^c, respectively;
      recalculate available split points such that each available split point p_l maintains $\frac{abs(p_l - p^c_m)}{|A_i|} > \alpha$,$\forall p^c_m \in p^c$;
      reset p^a with the new set of available split points;
      reset g^a accordingly;
    end
    while p^c *is not null* **do**
      add A_i into A^s;
      add the best gain ratio of g^c and corresponding split point into G^s and P^s, respectively;
      remove the best gain ratio of g^c and corresponding split point from g^c and p^c, respectively;
    end
  end
end
sort G^s in the descending order of gain ratios in G^s, and also rearrange P^s and A^s accordingly;
remove the elements in G^s, and corresponding elements in A^s and P^s where the difference between a gain ratio and the best gain ratio in G^s is greater than θ;
return A^s, and P^s;

Algorithm 2: Finding a set of good attributes, and a corresponding set of split points.

Input: d, a, p, r, c_f, n_l
Output: a decision tree T
BuildTree(d, a, p, r, c_f, n_l)

if a *is categorical* **then**
  /* divide the data set into $|a|$ number of segments such that in each segment all records have only one value of a and records of any two segments have different values of a */
  $d = \{d_1, d_2, \ldots d_{|a|}\}$;
end
if a *is numerical* **then**
  /* divide the data set into two segments such that in one segment all records have value of a greater than the split point p and in the other segment all records have values for a less than or equal to p */
  $d = \{d_1, d_2\}$;
end
initialise i to 1;
initialise a set of trees t to null;
while $i \leq |d|$ **do**
  t_i = call C4.5(d_i, r, c_f, n_l);
  $t = t \cup t_i$;
  $i = i + 1$;
end
build a tree T by joining the root attribute of each $t_i \in t$ as a child node to the parent node that tests a as the root attribute of T;
return T;

Algorithm 3: Building a single tree given a data set and a particular root attribute.

the trees generated by different algorithms we compare the prediction accuracies of the trees on a testing data set. We use our voting techniques for trees generated by SysFor. Similarly, we use MDMT voting for MDMT trees and CS4 voting for CS4 trees. For C4.5 we use the conventional prediction system where the majority class value of the leaf that a record belongs to is predicted as the class value of the record. We use 3 fold cross validation for prediction purpose, i.e. we divide a data set into three equal sized mutually exclusive horizontal segments and build decision trees from two segments while test prediction accuracy on the third segment. Each segment is used once in turn as a testing set for prediction accuracy test. Prediction accuracies are presented in Table 1.

Data Set	Technique	Prediction accuracy			Avg.
		Cross Val. 1	Cross Val. 2	Cross Val. 3	
CMC	SysFor (Voting-1)	51.53	55.19	52.75	53.16
	SysFor (Voting-2)	52.34	50.31	50.92	51.19
	MDMT	47.45	46.23	48.27	47.32
	CS4	49.69	49.94	50.10	49.91
	C4.5	48.88	52.34	53.56	51.59

Table 1: Prediction Accuracy: 3 Fold Cross Validation on Contraceptive Method Choice (CMC) Data set.

It is clear from Table 1 that SysFor with its Voting-1 performs better than all other techniques in terms of prediction accuracy. SysFor Voting-2 also performs better than MDMT and CS4. Moreover, an important advantage of SysFor over CS4 and MDMT is its ability to build many trees even from a low dimensional data set such as CMC. For example, SysFor builds 60 trees from CMC data set since in this experiment the user defined number of trees is 60. However, CS4 and MDMT produce only 9 and 3 trees, respectively. CS4 produces 9 trees because there are altogether 9 non-class attributes in the data set. It uses each of the nine attributes one by one as a root attribute regardless of the classification ability (gain ratio) of the attribute. MDMT builds 3 trees because it runs out of attributes. All non-class attributes are used up by the three trees.

In order to explore the trend of accuracy change for different user defined number of trees, we build sets of 60, 50, 40, 30, 20 and 10 trees and estimate the prediction accuracy of each set. In Figure 2 we present the average prediction accuracies, using three fold cross validation, for different sets of trees. For SysFor we use its Voting-1 during this test. The prediction accuracies of MDMT and CS4 techniques remain unchanged over different number of trees because the techniques fail to produce more trees even if the user defined number of trees is high. Since SysFor can build as many trees as required by a user we get a variation in accuracies. However, it appears that higher number of trees do not always result in higher accuracy. We find the maximum accuracy (53.7%) of SysFor for 20 trees. Fifty trees produced better accuracy (53.49%) than 10, 30, 40 and 60 trees. The trend of accuracy change for different number of trees deserves a more thorough investigation. However, for any number of trees (10, 20, 30 etc.) SysFor results in higher accuracy than the accuracies of other techniques.

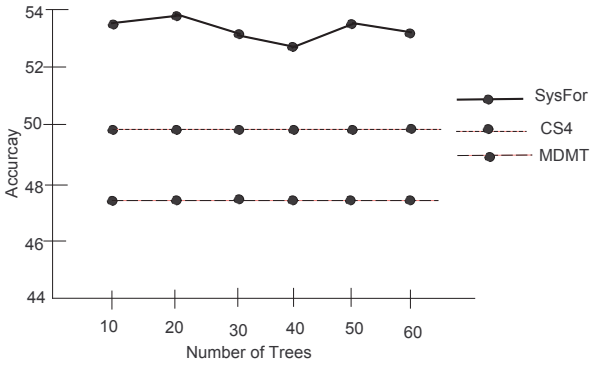


Figure 2: Prediction Accuracy VS User Defined Number of Trees

To evaluate the effectiveness of our voting technique we use our Voting-1 with trees generated by MDMT and CS4. We find an accuracy improvement in MDMT and CS4 when they use SysFor Voting-1. However, the accuracies are still lower than the accuracy of SysFor with its Voting-1 (see Table 1 and Table 2). Moreover, SysFor suffers from an accuracy drop when it is used along with CS4 and MDMT voting. The result is shown in Table 2.

Data Set	Technique	Prediction accuracy			Avg.
		Cross Val.1	Cross Val.2	Cross Val.3	
CMC	SysFor + CS4 Voting	50.92	52.75	51.12	51.60
	SysFor + MDMT Voting	51.73	54.18	51.93	52.61
	MDMT + SysFor Voting-1	46.44	52.34	49.69	49.49
	CS4 + SysFor Voting-1	52.14	53.56	51.53	52.41

Table 2: Effectiveness of Our Voting Technique

We also explore the quality of each individual tree generated by SysFor, CS4 and MDMT through the individual prediction accuracy of a single tree on the testing data set. In Figure 3 we present the individual prediction accuracy of the first 25 trees generated by SysFor, all 9 trees generated by CS4 and all 3 trees generated by MDMT, in cross validation number 2. SysFor trees maintain better accuracy than CS4 and MDMT trees. The 22nd SysFor tree has the highest accuracy of 54.38% among all 25 trees. This indicates that even after the generation of many trees SysFor is still capable of building high quality trees. Out of the 25 SysFor trees 7 have better accuracy than any tree generated by CS4 and MDMT. Both MDMT and CS4 trees suffer from quality drop after first few trees. Specially the quality of MDMT trees drop very sharp. The average accuracy of 25 SysFor trees, 9 CS4 trees and 3 MDMT trees are 50.40, 49.47 and 46.03, respectively. We also note that the combined accuracy (55.19%) of all SysFor trees is better than the accuracy of the best tree (the 22nd tree) indicating the

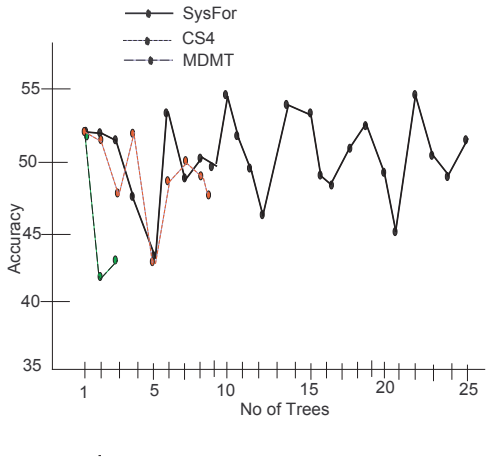


Figure 3: Prediction Accuracy of an Individual Tree

usefulness of SysFor Voting-1 in taking advantage of multiple trees.

We next apply SysFor, MDMT, CS4 and C4.5 algorithms on two high dimensional data sets, namely Lung Cancer (University of Michigan) data set and Central Nervous System data set available from Kent Ridge Biomedical Datasets [4]. The Lung Cancer data set has 96 records, each having 7129 non-class numerical attributes and a categorical class attribute. The class attribute has two values, cancer and normal. Out of 96 records, 86 are labeled as cancer and 10 as normal.The Central Nervous System data set has 60 records each having 7129 non-class numerical attributes and a categorical class attribute with class values Class 0 and Class 1. Out of 60 records 21 are for survivors and 39 are for failures. We build 20 trees from each data set using Goodness threshold = 0.2, and Separation threshold = 0.3. Moreover, for C4.5 we assign Confidence factor = 25%, Minimum Gain Ratio = 0.01 and Minimum number of records in each leaf = 2. The prediction accuracies of the techniques are presented in Table 3 and Table 4.

Data Set	Technique	Prediction accuracy			Avg.
		Cross Val.1	Cross Val.2	Cross Val.3	
Lung Cancer	SysFor (Voting-1)	100	100	100	100
	SysFor (Voting-2)	87.5	93.75	93.75	91.67
	MDMT	100	100	100	100
	CS4	100	100	100	100
	C4.5	100	96.88	100	98.96

Table 3: Prediction Accuracy on High Dimensional Lung Cancer Michigan Data set

For the Lung Cancer data set the accuracy of C4.5 single tree is very high and therefore, the multiple tree techniques do not have much room for further improvement. However, all of them improved the accuracy of a single tree from 98.96% to 100%. SysFor with its Voting-1 also achieves 100% accuracy. Voting-2 achieves 91.67% accuracy. Although Voting-2 does not perform very well on CMC and Lung Cancer data set it does very well on Central Nervous System data set. SysFor with Voting-2 achieves 65% accuracy whereas MDMT, CS4 and C4.5 achieve

Data Set	Technique	Prediction accuracy			Avg.
		Cross Val.1	Cross Val.2	Cross Val.3	
Cent. Nerv. Sys.	SysFor (Voting-1)	80	30	45	51.67
	SysFor (Voting-2)	70	70	55	65
	MDMT	60	65	45	56.67
	CS4	65	55	50	56.67
	C4.5	50	30	40	40

Table 4: Prediction Accuracy on High Dimensional Central Nervous System Data set

56.67%, 56.67% and 40% accuracy, respectively. SysFor with its two voting systems performs the best on all three data sets. However, finding an appropriate voting system appears to be an interesting problem. A possible solution to the problem can be to use an n Fold Cross Validation on a training data set in order to determine which voting system works better for the data set, and use the voting system for predicting the class value of a new record. We also plan to work on the voting system more deeply and present a better voting technique in future.

5 Conclusion

In this study we have presented a novel technique called SysFor for systematically building a forest of multiple trees. We have also presented two novel voting systems. A unique characteristic of the SysFor algorithm is that unlike existing techniques like CS4 and MDMT it only uses good attributes for building the trees. If it runs out of good attributes at a root node then unlike existing techniques it uses the good attributes available at level 1 of the best tree. Moreover, if it runs out of good attributes even at level 1 of the best tree it moves to the second best tree, third best tree and so on to use the good attributes at level 1 of those trees. Therefore, unlike the existing techniques, it does not require to build a tree that uses an attribute having poor gain ratio. Hence, the quality of logic rules and prediction accuracy of the trees should be better than the quality of logic rules and the prediction accuracy of the trees that are built using poor attributes. The algorithm also provides a user with the flexibility to adjust the extent of required "goodness" level of an attribute through an appropriate selection of the "goodness" threshold value that helps to evaluate whether the attribute qualifies as a good attribute.

Additionally, unlike existing techniques SysFor allows a numerical attribute to be chosen more than once if the split points are well separated and the gain ratios are high enough. We argue that well separated split points can actually extract different patterns/logic rules. For example, let us assume that an attribute "Age" has domain [0,120]. Logic rules R_1 : $Age > 70, Income > 80 \Rightarrow GoodCustomer$ and R_2 : $Age > 20, Suburb\_Code = Rich \Rightarrow GoodCustomer$ can be considered as significantly different since they have well separated split points for Age. SysFor algorithm aims to discover such patterns through multiple trees.

Another interesting characteristic of SysFor algorithm is that based on number of good attributes and number of records of horizontal segments at Level 1 the algorithm calculates possible number of trees

$t_p = \frac{\sum_{j=1}^{|D_f|} |A_j^g| \times |D_j|}{\sum_{j=1}^{|D_f|} |D_j|}$ such that bigger size segments have greater influence on number of possible trees calculation. Therefore, SysFor explores more alternative patterns when bigger size segments support this.

An important advantage of SysFor over existing techniques is its ability to build many trees even from a low dimensional data set due to its unique characteristic of using good attributes both at Level 0 and Level 1. Moreover, SysFor is easily extendable in the sense that it is not restricted to Level 0 and Level 1 nodes only, rather it can use the same approach on Level 2, Level 3 and so on nodes, if that is necessary and desirable.

We have also presented two novel voting systems. The main distinctive characteristic of our Voting-1 is that if among the multiple trees there is one or more trees having (pure) logic rule/s with high support (as defined by the Support threshold) and 100% accuracy (high confidence) for an unlabeled record then the class value of the unlabeled record is predicted based on only the pure logic rule/s. In that case the voting system overlooks the logic rules having heterogeneity as they are nonconclusive/uncertain. However, in the absence of such pure logic rules our voting technique uses all logic rules to predict the class value. SysFor Voting-2 uses the most accurate logic rule as it is considered to be our best bet.

Our initial experiments indicate that both SysFor and its voting systems are better than existing techniques and their voting systems. SysFor with its two voting systems has better accuracy than the accuracy of CS4, MDMT and C4.5, whereas experimental results [7, 6] indicate superiority of CS4 and MDMT over several other existing techniques including Random Forest, AdaBoostC4.5, SVM method, BaggingC4.5 and C4.5.

SysFor suffers from performance drop when it uses other voting systems such as CS4 Voting and MDMT Voting. However, other techniques make a performance improvement when they use our voting technique (Table 2) indicating a superiority of our voting techniques. Besides, SysFor always achieves better accuracy than the accuracy of other techniques, when an identical voting technique is used for all of them. Therefore, both SysFor and its voting techniques appear to be better than others.

From our initial experiments it appears that even on a low dimensional data set SysFor can build significantly bigger number of trees than the number of trees generated by CS4 and MDMT. When a single tree reveals a set logic rules (patterns) for a data set, a systematically developed set of multiples trees can reveal more logic rules in order to get much better information on the data set and thereby help better knowledge discovery.

Moreover, the quality of the trees built by SysFor is also consistently good as opposed to the trees generated by MDMT and CS4. SysFor builds trees one by one. Even a tree built at a later stage also has good accuracy on the testing data set. Within just the first 25 trees, there are seven SysFor trees which have better accuracy on testing data set than the accuracy of any (even the best) CS4 and MDMT trees (Figure 3). It also appears that regardless of the total number of trees generated SysFor always achieves better accuracy than the accuracy of MDMT and CS4 (Figure 2).

SysFor along with its Voting-1 performs the best on CMC and Lung Cancer data sets. However, it performs the best with its Voting-2 on Central Nervous System data set. Therefore, a data miner may want to test both voting systems on a training data set with an n-Fold Cross Validation and then for prediction purpose he/she can use the one that appears to be better on the data set. Our future work plans include the plan to improve the voting techniques, explore the relationships between number of trees and accuracy, improve SysFor algorithm, and explore if any type of data sets suits a voting system better.

References

[1] J. Abellan and S. Moral. Building classification trees using the total uncertainty criterion. *International Journal of Intelligent Systems*, 18(12), 2003.

[2] J. Abellan and S. Moral. Upper entropy of credal sets. application to credal classification. *International Journal of Approximate Reasoning*, 39(2-3), 2005.

[3] Joaquin Abellan and Andres R. Masegosa. An ensemble method using credal decision trees. *European Journal of Operational Research*, December 2009.

[4] Institute for Infocomm Research Data Mining Department. Kent ridge bio-medical dataset. available from http://datam.i2r.a-star.edu.sg/datasets/krbd/index.html. visited on 10.03.11.

[5] Jiawei Han and Micheline Kamber. *Data Mining Concepts and Techniques.* Morgan Kaufmann Publishers, San Diego, CA 92101-4495,USA, 2001.

[6] Hong Hu, Jiuyong Li, Ashley Plank, Hua Wang, and Grant Daggard. A comparative study of classification mehtods for microarray data analysis. In Jiuyong Li Simeon Simoff Peter Christen, Paul Kennedy and Graham Williams, editors, *Proceedings of the Australasian Data Mininig Conference (AusDM 2006)*, volume 61, pages 33–37, December, 2006.

[7] Hong Hu, Jiuyong Li, Hua Wang, Grant Daggard, and Mingren Shi. A maximully diversified multiple decision tree algorithm for microarray data classification. In *Proceedings of the Workshop on Intelligent Systems for Bioinformatics (WISB 2006), Conferences in Research and Practice in Information Technology (CRPIT)*, volume 73, 2006.

[8] Hong Hu, Jiuyong Li, Hua Wang, Grant Daggard, and Li-Zhen Wang. Robustness analysis of diversified ensemble decision tree algorithms for microarray data classification. In *Proceedings of the seventh International conference on Machine Learning and Cybernetics*, pages 115–120, 12-15 July 2008.

[9] Md Zahidul Islam. *Privacy Preservation in Data Mining through Noise Addition.* PhD thesis, School of Electrical Engineering and Computer Science, The University of Newcastle, Australia, June 2008.

[10] Md. Zahidul Islam. Explore: A novel decision tree classification algorithm. In *Proceedings of the 27th International Information Systems Conference, British National Conference on Databases (BNCOD 2010), Springer LNCS 6121 (in press)*, Dundee, Scotland, June 29 - July 01, 2010.

[11] Md Zahidul Islam, Payam Mamaani Barnaghi, and Ljiljana Brankovic. Measuring data quality: Predictive accuracy vs. similarity of decision trees. In *Proceedings of the 6th International Conference on Computer & Information Technology (ICCIT 2003)*, volume 2, pages 457–462, Dhaka, Bangladesh, 2003.

[12] Md Zahidul Islam and Ljiljana Brankovic. Privacy preserving data mining: A noise addition framework using a novel clustering technique. *Knowledge-Based Systems (accepted)*.

[13] Md Zahidul Islam and Ljiljana Brankovic. Noise addition for protecting privacy in data mining. In *Proceedings of of the 6th Engineering Mathematics and Applications Conference (EMAC 2003)*, volume 2, pages 85–90, Sydney, Australia, 2003.

[14] Jinyan Li and Huiqing Liu. Ensembles of cascading trees. In *Proceedings of the third IEEE international conference on Data Mining (ICDM 03)*, pages 9–17, Maebashi City, Japan, 2003. Australian Computer Society, Inc.

[15] Jinyan Li, Huiqing Liu, See-Kiong Ng, and Limsoon Wong. Discovery of significant rules for classifying cancer diagnosis data. *Bioinformatics*, 19(2):ii93–ii102, 2003.

[16] Jinyan Li and Kotagiri Ramamohanarao. A tree-based approach to the discovery of diagnostic biomarkers for ovarian cancer. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 04)*, pages 682 – 691, Maebashi City, Japan, 2004. Springer-Verlag Berlin Heidelberg.

[17] UCIrvine University of California. Uci machine learning. available from http://www.ics.uci.edu/∼mlearn/MLRepository.html. visited on 12.10.06.

[18] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, California, USA, 1993.

[19] J. Ross Quinlan. Improved use of continuous attributes in c4.5. *Journal of Artificial Intelligence Research*, 4:77–90, March 1996.

A novel hybrid neural learning algorithm using simulated annealing and quasisecant method

JOHN YEARWOOD[1] **ADIL BAGIROV**[1] **SATTAR SEIFOLLAHI**[1,2]

[1]School of Science, Information Technology and Engineering, University of Ballarat, Victoria 3353, Australia
[2] School of Civil, Environmental and Mining Engineering, University of Adelaide, South Australia 5005, Australia

Emails: j.yearwood@ballarat.edu.au, a.bagirov@ballarat.edu.au, s.seifollahi@ballarat.edu.au

Abstract

In this paper, we propose a hybrid learning algorithm for the single hidden layer feedforward neural networks (SLFNs) for data classification. The proposed hybrid algorithm is a two-phase learning algorithm and is based on the quasisecant and the simulated annealing methods. First, the weights between the hidden layer and the output layer nodes (output layer weights) are adjusted by the quasisecant algorithm. Then the simulated annealing is applied for global attribute weighting. The weights between the input layer and the hidden layer nodes are fixed in advance and are not included in the learning process. The proposed two-phase learning of the network is a novel idea and is different from that of the existing ones. The numerical results on some benchmark data sets are also reported and these results are promising.

Keywords: Classification, Quasisecant method, Simulated annealing, Attribute weighting

1 Introduction

Among different types of artificial neural network (ANN), single hidden layer feedforward neural networks (SLFNs) have become an extremely interesting topic of research because of their high learning ability and robustness. There are two main variations for SLFNs networks, those with additive hidden nodes and those with radial basis function (RBF) hidden nodes. Using both non-linear transfer functions provides the power of non-linearity to the network. Moreover, it has been proven that this class of networks is capable of universal approximation (Girosi et al. 1995).

A learning algorithm is at the heart of a neural based system. Learning can be considered as a weight updating rule of the ANN. Error Back Propagation (EBP), developed by Rumelhart et al., is probably the most cited learning algorithm. EBP is based on the gradient descent minimization method. Also, most of the neural learning algorithms depend on the gradient information of the error surface, which may not always be available or may be expensive to calculate. Moreover, the algorithm may easily be trapped in a local minima (Masters 1995).

One of the alternative learning techniques, that is more attractive, is the use of metaheuristics from global optimization. However, one of the problems though, with most of the global optimization methods is the time complexity of these methods, in particular for very large scale problems. For instance, finding the hidden layer weights in an SLFN, the weights between the input layer and the hidden layer nodes, can be handled by these methods, however when the application size or the number of hidden layer nodes is large, these methods need a further improvement in terms of the time complexity. Another alternative is the use of hybrid methods. The work of Ghosh et al., 1995, suggest different hybrid techniques, in which a local search method in conjunction with an evolutionary learning is used to update the hidden and output layer weights.

The ANN is one of the most popular algorithms used for data classification (Ghosh et al. 1995, Haykin 1999, Huang et al. 2006, Looney 1997). A neural network maps the input attribute vector to the network output consisting of classes. Most of the existing algorithms based on neural networks suffer from one or more drawbacks such as long training time, multiple local minima, high dependence on random initial weights and the need for tuning of parameters such as learning rate and momentum. In a classification process the contribution of attributes may be different. Giving weights to attributes may correct this imbalance and improve classification accuracy.

In this paper, we propose a new hybrid algorithm for the learning of the SLFN for classification. The proposed hybrid learning method is applied for updating the output layer weights and attribute weighting. It is based on the quasisecant algorithm (recently proposed by Bagirov and Ganjehlou, 2010) and the simulated annealing algorithm (Kirkpatrick et al. 1983). The simulated annealing is shown to be efficient for solving global optimization problems with box constraints.

We consider a two-phase learning algorithm. In the first phase of the algorithm, the output layer weights are found by the least square approach using the quasisecant method. In the second phase, the attribute weights are determined using the simulated annealing. The learning procedure here is different from that of existing ones, for instance the extreme learning machine (ELM) algorithm proposed by Huang et al. in 2006. In the ELM algorithm, the hidden layer parameters (hidden layer weights and biases) are randomly assigned and the only unknown parameters that need to be determined are the output linear weights, which can be handled using a least square method. However, the similarity between the

proposed algorithm and the ELM algorithm is that in both algorithms the hidden parameters are first decided by using random numbers and they are not adjusted during learning process. This process reduces the number of unknown parameters in the network and hence it mitigates the complexity of the problem.

The rest of this paper is organized as follows. Section 2 gives a brief review on the simulated annealing method and then the quasisecant method. In Section 3, we propose a hybrid method, which is based on the optimization methods introduced in the previous section. In Section 4, practical test results of the proposed algorithm by employing some test problems from literature are illustrated. We conclude the paper in section 5 followed by a few directions for future work.

2 Methods used in hybrid algorithm

2.1 The simulated annealing method

The simulated annealing procedure derives its name from the physical process of "annealing" or cooling of heated metals in which many final crystalline configurations which correspond to different energy states are possible depending upon the rate of the associated cooling process. According to (Kirkpatrick et al. 1983), this procedure can be traced to Metropolis who originally attempted to simulate the behavior of an ensemble of atoms in equilibrium at a given temperature. Metropolis constructed a mathematical model of the behavior of such a system that contained a method for minimizing the total energy of the system.

It is a fact that the atoms of a molten metal when cooled to a freezing temperature will tend to assume relative positions in a lattice in such a way as to minimize the potential energy of their mutual forces. Because of the huge number of atoms and resulting possible lattice arrangements (a combinatorial problem), the final derived state will typically correspond to only a local optimum and not a global one. Computationally, simulated annealing has been devised as a general optimization methodology and it can find global minima of general Lipschitz functions. It is not required that the function is smooth, that is, continuously differentiable.

The simulated annealing method consists of two main iterations: outer and inner iterations. In outer iterations the temperature $\mathcal{T}$ is updated. In order to do so we take any initial value $\mathcal{T}_0$ for temperature and a number $\alpha \in (0,1)$ and use the following schedule for temperature: $\mathcal{T}_{k+1} = \alpha\mathcal{T}_k, k = 0,1,2,\ldots$. In inner iterations we update the solution. Unlike many other optimization algorithms the simulated annealing method may accept not only downhill moves but also uphill moves. In order to generate a new solution in the inner iteration we randomly generate a new point x and also a uniformly distributed random number $p \in [0,1]$. Then we calculate the following number:

$$\beta = \min\left(1, \exp((f_{best} - f(x))/\mathcal{T})\right).$$

If $p \leq \beta$ then we accept x as a new solution, otherwise we repeat inner iterations. f_{best} is the best function value obtained by the simulated annealing. The number of inner iterations is restricted by some number provided by the user. If this number is reached then we go to implement the outer iteration.

The simulated annealing is a stochastic method and it can deal with both discrete and continuous variables. It is shown to be efficient for solving global optimization problems with box constraints.

2.2 The quasisecant method

The quasisecant method was introduced in (Bagirov and Ganjehlou 2010). It is a local method for solving nonsmooth, including nonconvex optimization problems. In general, this method is applicable for solving the following unconstrained minimization problem:

$$\text{minimize} \quad f(x) \tag{1}$$

where $x \in R^n$ and the objective function f is assumed to be locally Lipschitz. Formally, quasisecants are defined as follows. Let $S_1 = \{x \in R^n : \|x\| = 1\}$ be the unit sphere. A vector $v \in R^n$ is called a quasisecant of the function f at the point x in the direction $g \in S$ with the length $h > 0$ if

$$f(x + hg) - f(x) \leq h\langle v, g\rangle.$$

Here $\langle v, g\rangle$ is the inner product of vectors v and g in R^n. The above inequality is called a quasisecant inequality. Quasisecants provide overestimation to the function f in some neighborhood of a point x. There are many vectors v satisfying the quasisecant inequality. We consider only those which provide approximation to the function. Subgradient-related quasisecants introduced in (Bagirov and Ganjehlou 2010) provide such approximations and they converge to tangents of the graph of the function f. Any quasisecant is defined with respect to a given direction $g \in S_1$ and with given length $h > 0$. The choice of h allows one to compute descent directions with different lengths. Therefore, one can compute descent directions even from some shallow local minimizers using quasisecants. This observation makes the quasisecant method applicable to nonconvex problems and compute a "deep local minimizers". On the other hand, the quasisecant method uses a bundle of quasisecants at a given point to compute descent directions which makes it similar to the well-known bundle methods in nonsmooth optimization (Frangioni 2002, luksan and Vlcek 1998). Therefore, it is applicable to solve nonsmooth optimization problems. Results presented in (Bagirov and Ganjehlou 2010) demonstrate that the quasisecant method is efficient and robust method for solving nonsmooth, nonconvex optimization problems.

A brief description of the quasisecant method is followed. For more and detailed description of the method, the definitions and also the settings of the algorithms, see (Bagirov and Ganjehlou 2010).

Let the numbers $h > 0$, $c_1 \in (0,1)$, $c_2 \in (0, c_1]$ and a small enough number $\delta > 0$ be given.

Algorithm 1 Computation of the descent direction.

Step 1. Choose any $g^1 \in S_1$ and compute a quasisecant $v^1 = v(x, g^1, h)$ in the direction g^1. Set $V_1(x) = \{v^1\}$ and $k = 1$.

Step 2. Compute $\|\bar{v}^k\|^2 = \min\{\|v\|^2 : v \in \operatorname{co} V_k(x)\}$. If

$$\|\bar{v}^k\| \leq \delta, \tag{2}$$

then stop. Otherwise go to Step 3.

Step 3. Compute the search direction by $g^{k+1} = -\|\bar{v}^k\|^{-1}\bar{v}^k$.

Step 4. If

$$f(x + hg^{k+1}) - f(x) \leq -c_1 h \|\bar{v}^k\|, \quad (3)$$

then stop. Otherwise go to Step 5.

Step 5. Compute a quasisecant $v^{k+1} = v(x, g^{k+1}, h)$ in the direction g^{k+1}, construct the set $V_{k+1}(x) = \text{co}\left\{V_k(x) \bigcup \{v^{k+1}\}\right\}$, set $k = k + 1$ and go to Step 2.

Algorithm 2 The quasisecant method for finding (h, δ)-stationary points.

Step 1. Choose any starting point $x^0 \in \mathbb{R}^n$ and set $k = 0$.

Step 2. Apply Algorithm 1 for the computation of the descent direction at $x = x^k$ for given $\delta > 0$ and $c_1 \in (0, 1)$. This algorithm terminates after a finite number of iterations $m > 0$. As a result, we get the set $V_m(x^k)$ and an element v^k such that

$$\|v^k\|^2 = \min\left\{\|v\|^2 : \ v \in V_m(x^k)\right\}.$$

Furthermore, either $\|v^k\| \leq \delta$ or for the search direction $g^k = -\|v^k\|^{-1} v^k$

$$f(x^k + hg^k) - f(x^k) \leq -c_1 h \|v^k\|. \quad (4)$$

Step 3. If

$$\|v^k\| \leq \delta \quad (5)$$

then stop. Otherwise go to Step 4.

Step 4. Compute $x^{k+1} = x^k + \sigma_k g^k$, where σ_k is defined as follows

$$\sigma_k = \arg\max\left\{\sigma \geq 0 : \ f(\overline{x}) - f(x^k) \leq -c_2 \sigma \|v^k\|\right\},$$

where $\overline{x} = x^k + \sigma g^k$. Set $k = k + 1$ and go to Step 2.

Let $\{h_k\}$, $\{\delta_k\}$ be sequences such that $h_k \to +0$ and $\delta_k \to +0$ as $k \to \infty$.

Algorithm 3 The quasisecant method.

Step 1. Choose any starting point $x^0 \in \mathbb{R}^n$, and set $k = 0$.

Step 2. If $0_n \in \partial f(x^k)$, then stop. ∂f stands for subdifferential of f.

Step 3. Apply Algorithm 2 starting from the point x^k for $h = h_k$ and $\delta = \delta_k$. This algorithm terminates after finite many iterations $M > 0$, and as a result, it computes (h_k, δ_k)-stationary point x^{k+1}.

Step 4. Set $k = k + 1$ and go to Step 2.

It is proved in (Bagirov and Ganjehlou 2010) that under mild assumptions the sequence of points generated by the quasisecant method converges to stationary point of locally lipschitz continuous functions.

3 The proposed hybrid algorithm

3.1 The network formulation

Let us assume a data set $\mathcal{D} = \{(\mathcal{X}_s, y_s) | s = 1, \cdots, N\}$ of N arbitrary samples, where $\mathcal{X}_s$ is an $n-$dimensional vector of decision-making attributes, $\mathcal{X}_s = [x_{s1}, x_{s2}, \cdots, x_{sn}]^T$, and y_s is the desired output corresponding to the input $\mathcal{X}_s$, then the output of an SLFN, with H number of additive nodes, can be mathematically modeled as:

$$\sum_{j=1}^{H} \beta_j \phi_j(\mathcal{X}_s) = o_s, \quad s = 1, \cdots, N \quad (6)$$

$\beta_j, j = 1, \cdots, H$, is the weight vector connecting the jth hidden node to the output nodes, o_s is the network response to the sth sample and ϕ_j is the jth node in the hidden layer.

Let us consider sigmoid nodes in the hidden layer, then the function ϕ_j can be written in the following form:

$$\phi_j(\mathcal{X}_s) = \phi(w_j.\mathcal{X}_s + w_{0j}) = \frac{1}{1 + \exp(-w_j.\mathcal{X}_s - w_{0j})}$$

where $w_j = [w_{1j}, w_{2j}, \cdots, w_{nj}]^T$ and w_{0j} are the learning parameters of the jth hidden node, $w_j.\mathcal{X}_s$ denotes the inner product of vectors w_j and $\mathcal{X}_s$ in R^n.

If an SLFN with H number of hidden nodes can approximate these N samples with zero error, meaning that $\sum_{s=1}^{N} \|y_s - o_s\| = 0$, it then implies that there exist β_j, w_j such that:

$$\Phi\beta = Y \quad (7)$$

where

$$\Phi = \begin{bmatrix} \phi_1(\mathcal{X}_1) & \cdots & \phi_H(\mathcal{X}_1) \\ \vdots & \cdots & \vdots \\ \phi_1(\mathcal{X}_N) & \cdots & \phi_H(\mathcal{X}_N) \end{bmatrix}_{N \times H},$$

$$\beta = [\beta_1, \cdots, \beta_H]^T$$

and

$$Y = [y_1, y_2, \cdots, y_N]^T.$$

The system (7) is a linear system with respect to the output weights of the network. It can be estimated as the least square problem:

$$f(\beta) = \sum_{s=1}^{S} \Big[\sum_{k=1}^{K} (y_{ks} - o_{ks})^2\Big] \quad (8)$$

where y_{kp} is the desired value of the kth output and the sth sample, o_{kp} is the actual value of the kth output and the sth sample, S is the number of samples, and K is the number of the network outputs. Equation (8), also, can be written in the following matrix form.

$$f(\beta) = E^T E$$

where E is an $K \times S$ matrix with entries $e_{ks} = y_{ks} - o_{ks}, \quad k = 1, \cdots, K, \quad s = 1, \cdots, S$.

For a typical binary classification problem, if the desired output is coded as "1" for samples from class

1 and "-1" for samples from class 2, the classifier determines the class label of the input vector $\mathcal{X}_s$ as

$$\widehat{Y}(\mathcal{X}_s) = sign(o_s) = sign(\sum_{j=1}^{H} \beta_j \phi_j(\mathcal{X}_s)) \quad (9)$$

where the scaler o_s is the network output corresponding the input vector $\mathcal{X}_s$.

3.2 Attribute weighting

Good attribute weighting can eliminate the effects of noisy or irrelevant attributes. There are some attribute weighting methods in the literature (Ozsen and Gunes 2009, Wu and Cai 2011). In this section, we propose weights for attributes, in which each attribute has its own power as a weight. The idea of our weighting method is the same as the work of Wu and Cai, for attribute weighting in attribute weighted Naive bayes (AWNB). More precisely, assuming the attributes are independent conditionally to the class variable, the AWNB classifier assigns to each sample $\mathcal{X}_s$ the class value having the highest conditional probability as:

$$\widehat{Y}(\mathcal{X}_s) = \arg\max_{y \in Y} P(y) \prod_{i=1}^{n} P(x_{si}|y)^{\xi_i} \quad (10)$$

where x_{si} is the value of the ith attribute, $P(y)$ is known as the apriori probability of the class, $P(x_{si}|y)$ are conditional attribute-value probabilities and ξ_i is the weight for the ith attribute. In this work, similar to the attribute weighting of Wu and Cai, we consider the weights for each attribute. Then they are updated by a global optimization procedure using the simulated annealing method in the second phase of our algorithm.

If we consider weights for attributes as powers, then, the output of the hidden layer nodes, ϕ, can be formulated as:

$$\phi_j(\mathcal{X}_s) = \phi(\sum_{i=1}^{n} w_{ij} x_{si}^{\xi_i} + w_{0j}), \quad j = 1, \cdots, H \quad (11)$$

where ξ_i is the weight for the ith attribute. So, the cost function of (8), by considering these attribute weights, can be considered as a function of two sets of variables, that is, the output layer weights and the weights for attributes. More precisely, the purpose here is minimizing the cost function with respect to the sets of variables so that the attribute weights lie in a hyperbox $[a, b]$. Therefore, the minimization problem of (8) can be reformulated as:

$$\text{minimize} \quad f(\beta; \xi) \quad (12)$$

subject to

$$\xi \in [a, b], \quad a, b \in R_+^n. \quad (13)$$

Lower bound a_i and upper bound $b_i, i = 1, \cdots, n$, ith components of a and and b corresponding to the ith attribute, are positive and $a_i = \ell$ and $b_i = \xi_i + u$. ℓ and u are constants; here we set them 0.1 and 1, respectively.

3.3 The learning algorithm

A new heuristic hybrid algorithm for the learning of the SLFN is considered for classification. The proposed algorithm is a new two-phase learning algorithm and differs from that of the existing ones. In the first phase of the algorithm, the output layer weights are adjusted by a local optimization, which here is the quasisecant method. Then, a global optimization method, which here is the simulated annealing, is applied to find the attribute weights. The weights and biases between the input layer and the hidden layer nodes, the hidden layer weights, are fixed in advance and they are not updated during the learning process. The algorithm proceeds until an improvement in the objective function value occurs and a pre-specified number of iteration is not reached. The hybrid algorithm proceeds as follows:

Algorithm 4 Hybrid method.

Step 1. Fix all the hidden layer weights with random numbers uniformly distributed from $[0, 1]$ and initialize the attribute weights, $\xi_i = 1, i = 1, \cdots, n$. Choose any starting point $\beta^0 \in R^H$ and set $a^0 = a$, $b^0 = b$, $k := 0$.

Phase 1. Updating output layer weights

Step 2. Apply the quasisecant method to find a stationary point for the output weights of the network starting from the point β^k.

Phase 2. Updating attribute weights

Step 3. Update a hyperbox $[a^{k+1}, b^{k+1}] \subset [a, b]$, where a and b are positive, i.e. $a, b \in R_+^n$.

Step 4. Apply the simulated annealing method to find the attribute weights in order to globally minimize the function value in the hyperbox $[a^{k+1}, b^{k+1}]$.

Step 5. Let $\bar{\xi}$ be obtained by minimizing of the function f using the simulated annealing on $[a^{k+1}, b^{k+1}]$. If $f(\beta^k; \bar{\xi}) < f(\beta^k; \xi^k)$ then set $\xi^k = \bar{\xi}$ and go to Phase 1 (Step 2), otherwise the algorithm terminates.

4 Experiments

4.1 Data collections

We use 10 real world data sets to test the proposed algorithm. The first nine data sets used in these experiments can be found in the UCI repository of machine learning databases (Asuncion and Newman 2007), and the last data set is downloadable on the tools page of LIBSVM (Chang and Lin 2001). These data sets have been analyzed more frequently by the current data mining approaches. Another reason for selecting of these data sets were that conventional approaches have analyzed them with variable success. Table 1 shows the brief description of the data sets used.

Table 1: Brief description of the experimental data sets

Data set	# Attributes	# Samples
Breast Cancer	10	683
Congres Voting	16	435
Credit Approval	14	690
Diabetes	8	768
Haberman's Survival	3	306
Ionosphere	34	351
Liver Disorders	6	345
Phoneme CR	5	5404
Statlog (Heart)	13	270
Svmguide3	21	1284

4.2 Results and discussion

We consider a two-phase learning algorithm, in which the output layer weights are found by the least square approach using the quasisecant method and then the attribute weights are determined using the simulated annealing. The hidden layer weights of the networks are fixed in advance. The proposed algorithm terminates until no improvement in the objective function value occurs and/or a prespecified number of iteration has not been reached.

In the following table, "ELM" stands for the network with the ELM learning algorithm, proposed by Huang et al. in 2006. In the ELM, the parameters of hidden layer nodes, such as weights and biases between the input and the hidden layer nodes, are randomly assigned. Therefore, the only parameters which need to be determined are the output layer weights. These weights are found by the least square method using the Moore-Penrose generalized inverse (Huang et al. 2006, Rao and Mitra 1971). The activation functions used in the hidden layer are sigmoid functions.

Notation "RBFN" represents the RBF network, i.e. a three-layer feed-forward neural network with RBF kernels in the hidden layer nodes (Abdel at al. 2008, Haykin 1999, Golbabai et al. 2009). In the RBF network, all the hidden layer weights are assigned one and other parameters in the hidden layer (the centers and widths of the RBFs) can often be pre-fixed. Here, in "RBFN", the centers of the RBFs are chosen randomly from the training samples and the jth width, σ_j, is defined as the distance between the jth center, $\mathcal{X}_j$, and its nearest center (Abdel at al. 2008):

$$\sigma_j = \gamma \min\{\|\mathcal{X}_j - \mathcal{X}_k\| : \ k = 1, \cdots, H, \ k \neq j\}$$

where γ has to be set heuristically (the suggested value is $\gamma = 1.5$). The only unknown output weights, in "RBFN", are found by the least square method using the Moore-Penrose generalized inverse. Here, the multiquadratic function is used in the hidden layer nodes which is of the form:

$$\phi(\|\mathcal{X} - \mathcal{X}_j\|) = (\|\mathcal{X} - \mathcal{X}_j\|^2 + \sigma_j^2)^{1/2},$$

where $\|\mathcal{X} - \mathcal{X}_j\|$ is the Euclidean distance between the input vector $\mathcal{X}$ and the center $\mathcal{X}_j$.

"QSM" represents the network with considering the first phase of the algorithm, i.e. the quasisecant method for updating the output layer weights. The attribute weights are not considered in this case. "HYBRID" stands for the network with the proposed hybrid algorithm, in which in the first phase the output layer weights and in the second phase the attribute weights are updated. Here, we used sigmoid function in the hidden layer nodes of the networks "QSM" and "HYBRID".

Table 2 shows the average accuracy results of the networks "ELM", "RBFN', "QSM" and "HYBRID" described above; using 15 nodes in the hidden layer. The first column of the table shows the data set used in our experiments and the subsequent columns show the average accuracies (in percentage) of 5 independent runs. Also, in each run, we use 5-fold cross validation method with random orders in partitioning training and test data sets to have more reliable results. More precisely, each fold contained 20% of the data set randomly selected (without replacement).

According to the results in Table 2, the proposed hybrid method, "HYBRID", outperforms the "ELM" and "RBFN"; improving almost 4.9%, 2.8% on the averages, respectively. Also, the numerical results show that the quasisecant, "QSM", method works well, and it can be a good alternative to update the weights in ANNs. Attribute weighting could not remarkably improve the results obtained by the first phase of the algorithm; improving almost 1.3% on the average. Of course, it should be noted that we repeat the algorithm twice and after that we terminate the algorithm. Moreover, the results obtained by the RBF kernels, i.e. "RBFN", are more accurate than those of "ELM"; improving almost 2% on the average.

Table 2: The average accuracy results (%) of the networks; using five-fold cross validation and 15 nodes in the hidden layer

Data set	ELM	RBFN	QSM	HYBRID
Breast Cancer	96.80	96.98	97.79	**98.38**
Congres Voting	84.19	91.91	91.40	**93.93**
Credit Approval	84.18	85.81	**90.00**	**90.0**0
Diabetes	77.31	77.39	78.88	**78.93**
Haberman's Survival	75.87	74.62	76.85	**78.36**
Ionosphere	80.23	**90.29**	83.14	86.00
Liver Disorders	71.77	68.12	76.52	**77.97**
Phoneme CR	75.80	76.80	86.48	**86.50**
Statlog (Heart)	77.70	82.67	81.63	**82.69**
Svmguide3	78.69	78.45	76.64	**78.98**
Average	80.25	82.3	83.93	**85.17**

We have coded the "QSM" and "HYBRID" in Fortran and "ELM" and "RBFN" in Matlab. The time complexity of the networks are not included in this work due to different programming languages, however it is clear that the time complexity of the proposed hybrid method are more than the others, which is followed by the "QSM". Also, it is note that the time complexity of the methods including the proposed methods are less than the standard BP network since the hidden weights in all these networks are fixed in advance.

5 Conclusion

In this paper, we proposed a classifier based on single hidden layer feedforward neural networks (SLFNs). A novel two-phase learning algorithm was used for learning of the SLFNs. The proposed algorithm uses the quasisecant and the simulated annealing algorithms. In the first phase, the output layer weights are updated by the quasisecant method, then the simulated annealing is utilized for attribute weighting. The hidden layer weights are fixed in advance and are not included in the learning process. The proposed algorithm is different from that of the existing ones and also the ELM algorithm. The proposed algorithm is a new general learning idea and also can be extended to multiclass. It is noted that other alternative optimization methods can be used in the hybrid algorithm to update the proposed weights.

We carried out a number of experiments on some different data sets obtained from the UCI repository and also from tools page of LIBSVM. The numerical results show that the proposed algorithm has positive effects on the network performance. If the number of hidden nodes increases beyond a certain threshold, the network performance may degrade because of the network complexity and overlapping. It will not be as bad as other traditional networks since their complexity increases due to increasing the number of the hidden layer weights. Research concerning the complexity is not included in this work and it is future challenging work.

References

Abdel Hady, M. F., Schwenker, F., & Palm, G. (2008), Semi-supervised learning of three-structured RBF networks using co-training, ICANN 2008, Part I, LNCS 5163, pp. 79-88.

Asuncion, A., & Newman, D. (2007), UCI machine learning repository. School of Information and Computer Science, University of California, Irvine (http://www.ics.uci.edu/mlearn/MLRepository.html).

Bagirov, A.M & Nazari Ganjehlou, A. (2010), A quasisecant method for minimizing nonsmooth functions, Optimization Methods and Software 25, pp. 3-18.

Bishop, C.M. (1995), Neural networks for pattern recognition, Oxford: Clarendon Press.

Chang, C., & Lin, C. (2001), LIBSVM: A library for support vector machines, Software available at http://www.csie.ntu.edu.tw/cjlin/libsvm.

Frangioni, A. (2002), Generalized bundle methods, SIAM Journal on Optimization, 113(1), 117-156.

Ghosh, R., Yearwood, J., Ghosh M., & Bagirov, A. (2006), A hybrid neural learning algorithm using evolutionary learning and derivative free local search method, International Journal of Neural Systems, 16, 201-213.

Girosi, F., Jones, M., & Poggio, T. (1995), Regularization theory and neural networks architectures. Neural Computation, 7, 219-269.

Golbabai, A., Mammadov, M., & Seifollahi, S. (2009), Solving a system of nonlinear integral equations by an RBF Network, Computers & Mathematics with Applications, 57, 1651-1658.

Haykin, S. (1999), Neural networks: a comprehensive foundation, Englewood Cliffs, NJ: Prentice-Hall.

Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2006), Extreme learning machine: Theorey and applications, Neurocomputing 70, 489-501.

Kirkpatrick, S., Gelatt, C. D., Jr., & Vecchi, M. P. (1983), Optimization by Simulated Annealing, Science, 220, 671-680.

Looney, C.G. (1997), Pattern recognition using neural networks: Theory and algorithm for Engineers and Scientist, Oxford University Press, New York.

Luksan. L., & Vlcek, J. (1998), A bundle Newton method for nonsmooth unconstrained minimization, Mathematical Programming, 83, 373-391.

Masters, T. (1995), Advanced Algorithms for Neural Networks: A C++ Sourcebook, Wiley, New York.

Özşen, S., & Güneş, S. (2009), Attribute weighting via genetic algorithms for attribute weighted artificial immune system (AWAIS) and its application to heart disease and liver disorders problems, Expert Systems with Applications, 36, 386-392.

Rao, C.R., & Mitra, S.K. (1971), Generalized inverse of matrices and its applications, New York: Wiley.

Wu, J., & Cai, Z. (2011), Attribute Weighting via Differential Evolution Algorithm for Attribute Weighted Naive Bayes (WNB), Journal of Computational Information Systems, 1672-1679.

Seed-Detective: A Novel Clustering Technique Using High Quality Seed for K-Means on Categorical and Numerical Attributes

Md Anisur Rahman, and Md Zahidul Islam
Centre for Research in Complex Systems, School of Computing and Mathematics,
Charles Sturt University, Wagga Wagga, NSW 2678,
Australia.

{arahman,zislam}@csu.edu.au

Abstract

In this paper we present a novel clustering technique called Seed-Detective. It is a combination of modified versions of two existing techniques namely Ex-Detective and Simple K-Means. Seed-Detective first discovers a set of preliminary clusters using our modified Ex-Detective. The modified Ex-Detective allows a data miner to assign different weights (importance levels) for all attributes, both numerical and categorical. Centers of the preliminary clusters are then considered as initial seeds for the modified Simple K-Means, which unlike existing Simple K-Means does not randomly select the initial seeds. Centers of the preliminary clusters are naturally expected to be better quality seeds than the seeds that are chosen randomly. Having better quality initial seeds as input the modified Simple K-Means is expected to produce better quality clusters. We compare Seed-Detective with several existing techniques including Ex-Detective, Simple K-Means, Basic Farthest Point Heuristic (BFPH) and New Farthest Point Heuristic (NFPH) on two publicly available natural data sets. BFPH and NFPH were shown in the literature to be better than Simple K-Means. However, our initial experimental results indicate that Seed-Detective produces better clusters than other techniques, based on several evaluation criteria including F-measure, entropy and purity. Another contribution of this paper is the experimental result on Ex-Detective which was never tested before.

Keywords: Clustering, Classification, Decision Tree, Data Mining.

1 Introduction

Due to the recent technological advancement huge amount of data is being collected in almost all sectors of life. Various data mining tasks including clustering is often applied on huge data sets in order to extract hidden and previously unknown information which can be helpful in decision making processes. For example, world's largest retailer Wal-Mart Stores, Inc. captures point-of-sale transactions from each of its retail outlets, and stores them in its massive data warehouse. Wal-Mart performs various data mining tasks including clustering on the collected data by using different tools such as NeoVista's Decision Series (TM) and Decision Series 2.1 (Wal-Mart).

Clustering techniques assemble similar records in a cluster and dissimilar records in different clusters (Han and Micheline 2006, Tan, Steinbach, and Kumar 2006). Clustering has wide range of applications such as medical imaging, gene sequence analysis, market research, social network analysis, crime analysis, software evaluation, machine learning and search result grouping (Antonellis, Antoniou, Kanellopoulos, Markis, Theodoridis, Tjortjis, and Tsirakis 2009, Songa and Nicolae 2008, Grubesic and Murray 2001, Chui-Yu, Yi-Feng I-Ting 2008, and He 2008, Jian and Heung 2011, Haung and Pan 2009). Therefore, for an effective decision making purpose it is important to obtain a set of high quality clusters.

In this study we present a novel clustering technique called Seed-Detective which is a combination of modified versions of two existing techniques; Ex-Detective (Islam and Brankovic 2005, Islam 2008, Islam and Brankovic 2011), and Simple K-Means (SimpleKMeans, Han and Micheline 2006, K-Means). Seed-Detective takes weights (importance/significance levels) for all attributes, both numerical and categorical, as a user input and accordingly produces a set of preliminary clusters applying the modified Ex-Detective. We then find the centers of the preliminary clusters, where a cluster centre is a made-up record having average values for each numerical attribute and mode values for each categorical attribute of all records belonging to the cluster. The cluster centers are then considered as initial seeds and given as an input to the modified Simple K-Means for further clustering. Since the initial seeds are chosen from the preliminary clusters (instead of selecting them randomly as it is done in Simple K-Means) it is expected that they are of better quality. Having better quality initial seeds as input the modified Simple K-Means is also expected to produce better quality clusters.

We compare the cluster quality of Seed-Detective with several existing techniques including Ex-Detective, Simple K-Means, Basic Farthest Point Heuristic (BFPH) and New Farthest Point Heuristic (NFPH) (Islam and Brankovic 2005, Islam 2008, Islam and Brankovic 2011, SimpleKMeans, Han and Micheline 2006, K-Means, Haung 1997, He 2006) on two publicly available data sets based on a range of evaluation criteria. BFPH and NFPH were shown in the literature to be better than Simple K-

This work was supported by the 2nd author's CRiCS Seed Grant from the Centre for Research in Complex Systems (CRiCS), Charles Sturt University, Australia.

Means (He 2006). Our experimental results indicate that Seed-Detective produces better quality clusters than the clusters produced by other techniques. Another contribution of this study is the presentation of experimental results on Ex-Detective, which was never tested before.

The structure of the paper is as follows. In Section 2 some existing clustering techniques are discussed. In Section 3 we present our novel clustering technique. Experimental results are presented in Section 4. Finally, we give concluding remarks in Section 5.

2 Related work

DEcision TreE based CaTegorIcal ValuE clustering technique (Detective) (Islam and Brankovic 2005, Islam 2008, Islam and Brankovic 2011) explores similarity among attribute values instead of similarity among the records of a dataset, unlike most of the existing techniques. Moreover, it explores similarities among values within a horizontal segment (partition) of a data set, instead of the whole data set. It assumes that two values belonging to an attribute may not be similar within a whole data set but still can appear to be very similar within a horizontal segment of the data set.

It first builds a decision tree (Quinlan 1993, Quinlan 1996, Islam 2010) considering the attribute (the values of which need to be clustered) as the class attribute. The class values belonging to a heterogeneous leaf are considered to be similar to each other, where within a heterogeneous leaf records have different class values. The class values of a heterogeneous leaf are considered to be similar only within the records belonging to the leaf. The values may not be similar for other records. Detective also considers the class values of two sibling leaves (i.e. the leaves having the same parent node) similar to each other.

Ex-Detective is an extended version of Detective (Islam and Brankovic 2005, Islam 2008, Islam and Brankovic 2011). Unlike Detective it clusters records of a dataset. It first builds a decision tree (DT) for each categorical attribute separately considering the attribute as the class attribute. (Note that if categorical attributes have big domain sizes then the tree can be wide.) It next creates clusters of records by using the decision trees in such a way so that two records belonging to the same leaf in each decision tree end up in the same cluster. For a dataset having a combination of categorical and numerical attributes, Ex-Detective first obtains clusters based on only categorical attributes. Within a cluster, a distance based clustering technique such as Simple K-Means (SimpleKMeans, Han and Micheline 2006, K-Means) is then used for the numerical attributes.

An advantage of Ex-Detective is that it allows a data miner to assign different weights (levels of importance) on different categorical attributes instead of considering all categorical attributes equally important for clustering. Using our example dataset (Figure 1) we now explain the steps of Ex-Detective as follows. Two decision trees are built, one by one, considering the categorical attributes A1 and A3 as the class attributes, respectively as shown in Figure 2 and Figure 4.

The depth of the decision trees for attributes A1 and A3 are 3 and 2, respectively. A decision tree is then trimmed (pruned) based on the user defined weight for the class attribute of the tree. The weights can vary from 0 to 1. Let us consider that the weights assigned on A1 and A3 are 0.6 and 0.6, respectively. The depth of the tree (Figure 2) that considers A1 as the class attribute is therefore reduced from 3 to 2 by multiplying the depth (3) by the class attribute's weight (0.6) as $3*0.6=1.8 \cong 2$. The pruned tree is shown in Figure 3, where Leaf 12 contains all records belonging to Leaf 3 and Leaf 4 of Figure 2. Similarly, the tree (Figure 4) considering attribute A3 as the class attribute is pruned to a tree as shown in Figure 5.

Ex-Detective produces clusters based on attribute A1 and A3 by taking intersections of sets of records belonging to the leaves of the pruned trees for A1 and A3 (Figure 3 and Figure 5). For example, the intersection of Leaf 10 and Leaf 14 gives the property "A1= {a12}, A2 = any value, A3= {a31, a32}, and A4>7" which is satisfied by records R4, R6 and R9. Hence, Ex-Detective considers R4, R6 and R9 to be a cluster of records following the trees for A1 and A3. An interesting observation is that R9 has different value (a32) for attribute A3 than the value (a31) of R4 and R6 for A3. This is possible because according to Leaf 14, a31 and a32 are considered to be similar.

Within each cluster a distance based clustering technique such as Simple K-Means (SimpleKMeans, Han and Micheline 2006, K-Means) is applied for numerical attributes A2 and A4. Therefore, Ex-Detective further divides the records into a number of clusters.

If attribute A1 is assigned zero weight then the tree that considers A1 as the class (see Figure 2) is pruned to Level 0 having only one leaf containing all records of the data set. Therefore, A1 has zero influence in the clustering process through intersections of the leaves since based on the attribute all records are considered to be similar. However, we note that the attribute can still be used in another tree and therefore can still have some influence in the clustering process. We argue in this study that the influence of A1 in final clustering is zero (due to the zero weight assigned to it) as long as it is not tested in any other trees i.e. it is not influential in exploring similarity of any other attributes. If a categorical attribute is un-related to other attributes and a user assigns zero weight to the attribute then according to Ex-Detective the attribute does not have any influence on the final clustering.

Record \ Attribute	A1	A2	A3	A4
R1	a11	5	a31	10
R2	a13	7	a31	5
R3	a11	7	a32	7
R4	a12	5	a31	10
R5	a13	3	a32	4
R6	a12	15	a31	10
R7	a11	5	a32	3
R8	a13	6	a32	10
R9	a12	6	a32	10

Figure 1: An Example dataset having nine records

Simple K-Means randomly selects as many records as the user defined number of clusters and considers the records as initial seeds (SimpleKMeans, Han and Micheline 2006, Haung 1997, K-Means). Any other record of the dataset is assigned to the cluster, the seed of which has the smallest distance with the record. The distance is measured by adding the differences of the values belonging to each attribute of the records. For a numerical attribute the absolute difference between the two normalised values is used. However, for a categorical attribute if both records have the same value for the attribute then their difference is considered to be zero, and otherwise it is considered to be one (Haung 1997).

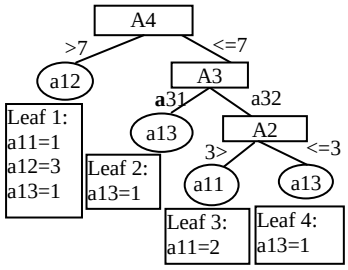


Figure 2: Decision Tree considering attribute A1 as a class attribute.

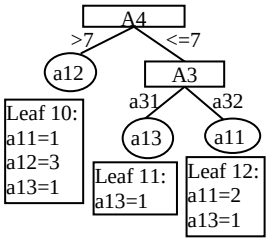


Figure 3: Pruned Decision Tree on attribute A1

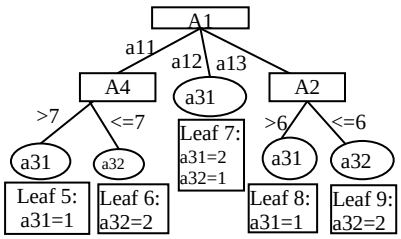


Figure 4: Decision Tree considering attribute A3 as the class attribute

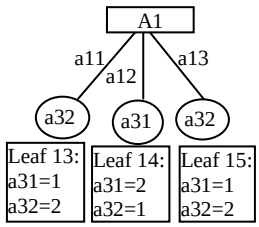


Figure 5: Pruned Decision Tree on attribute A3

In the next iteration, for each cluster Simple K-Means calculates a new seed which is the cluster center of the cluster. All records are again reorganized such that a record is assigned to the cluster the seed of which has the minimum distance with the record. The process of reorganizing records and finding new seeds continues recursively until a termination conditions is satisfied. Typically, the number of iteration and a minimum difference between the seeds are considered as termination conditions.

In Basic Farthest Point Heuristic (BFPH), the first initial seed is selected randomly. Other seeds are then selected deterministically so that the second seed is the record having the maximum distance with the first seed (He 2006). For all other records, distances between a record and both seeds are then calculated. The record having the maximum distance with the seed closer to it is then considered as the third seed. The new seed selection process continues until as many seeds as the user defined number of clusters is chosen. After the initial seed selection BFPH follows the same approach as the Simple K-Means.

In New Farthest Point Heuristic (NFPH), the first initial seed is also selected deterministically (He 2006). Frequencies of all values appearing in a record are added to determine the score of the record. The record having the highest score is considered as the first seed of the dataset. Other seeds are selected using the same approach taken by BFPH. An advantage of NFPH is that unlike Simple K-Means and BFPH, it produces the same set of clusters every time it is applied on a data set.

3 A Novel Clustering Technique

Seed-Detective is a combination of modified Ex-Detective and modified Simple K-Means. We first give a high level pseudo code of Seed-Detective as follows and then explain the steps.

Step 1: Produce a set of preliminary clusters using our modified Ex-Detective.

Step 2: Calculate seeds of the preliminary clusters.

Step 3: Input the seeds in a modified Simple K-Means in order to produce final clusters.

In Step 1, a preliminary set of clusters is produced by using a modified version of Ex-Detective. There are two modifications of Ex-Detective. The first modification allows a user to assign weights on numerical attributes, in addition to categorical attributes. We then build a decision tree (Quinlan 1993, Quinlan 1996, Islam 2010) for each attribute (both categorical and numerical) considering the attribute as the class. If the attribute is numerical we first categorize (generalize) the values belonging to the attribute into a user defined number of categories. The categorization is done only for the purpose of building a decision tree where the class attribute is originally a numerical attribute. All other non-class numerical attributes are considered in their original numerical form.

After the trees are built, Seed-Detective then prunes them and produces intersections of the leaves in the same way as Ex-Detective. The intersections are considered as the set of preliminary clusters. The second modification of Ex-Detective is that we do not apply K-Means in each intersection separately.

In Step 2, seeds of the preliminary clusters are calculated. Our objective function for initial seed selection aims to minimize the sum of the squared error

(SSE) of the distances between the initial seed of a cluster and the records belonging to the same cluster. However, we only consider the significant attributes while calculating SSE. Moreover, for a significant attribute if there are more than one values that are similar to each other (as it is the case for attribute A3 of R4, R6 and R9 for Leaf 14, see Figure 1 and Figure 5) then we consider the difference between the similar values to be zero while calculating SSE. Let C_j be the *j*th cluster, c_j be the seed of the *j*th cluster, *k* be the number of clusters, x_i be the *i*th record belonging to the *j*th cluster, *m* be the number of records of the *j*th cluster, and $dist(x_i, c_j)$ be the distance between x_i and c_j based on the significant attributes only. We represent the objective function as follows.

$$SSE(C_1, C_2, \ldots\ldots C_k) = \sum_{j=1}^{k} \sum_{i=1}^{m} dist(x_i, c_j)^2. \qquad (1)$$

In order to achieve this objective function we first build a decision tree (as mentioned in Step 1). Since a decision tree algorithm such as C4.5 aims to minimize the entropy for the class values within a leaf it attempts to increase the possibility of having the same class value among all records belonging to a leaf. Let, *H* be the entropy of the class values in a leaf, $|p|$ be the domain size of the class attribute, $p(q_l)$ be the probability of the *l*th class value in the leaf. *H* can be represented as follows.

$$H = -\sum_{l=1}^{|p|} p(q_l) log_2 p(q_l). \qquad (2)$$

Therefore, it is clear that in order to minimize *H* we need to maximize the proportion of a class value q_l. When *H* is equal to zero then all records of a leaf have the same class value. Let us assume that we build a tree that considers attribute A1 as the class attribute as shown in Figure 6. Each leaf of the tree contains a set of records that have the same or similar class value/s. Therefore, the records belonging to a leaf can be considered as a cluster based on the class attribute. These clusters are informally shown in Figure 7 where dots represent records. Let us also assume that we get another set of clusters based on attribute A2 as shown in Figure 8 and Figure 9. Finally we intersect the clusters and get another set of clusters (Figure 10) where all records in a cluster have the same or similar value/s for each of the attributes A1 and A2. Therefore, the seed of a cluster also have the same or similar values for the significant attributes resulting in a zero distance (based on the attributes A1 and A2) between the seed and any record belonging to the cluster.

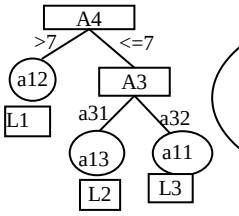


Figure 6: DT on attribute A1

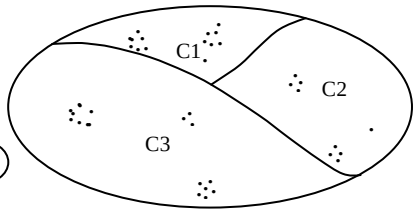


Figure 7: Clusters based on A1

Since we categorize a numerical significant attribute, all records belonging to a cluster have values (for the attribute) belonging to the same or similar category. Therefore, the seed is calculated by taking the average of the actual numerical values in order to minimize the SSE.

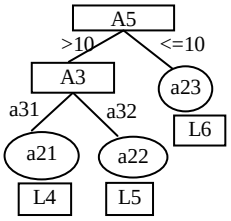


Figure 8: DT on attribute A2

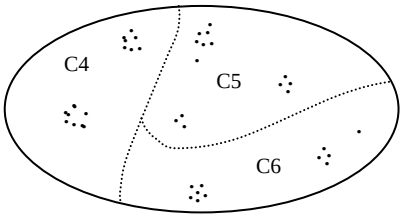


Figure 9: Clusters based on A2

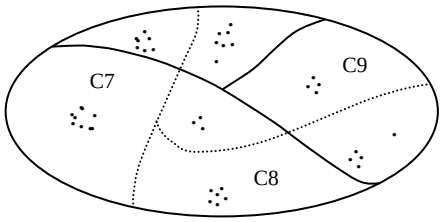


Figure 10: Clusters based on A1 and A2

In Step 3, the data set and the seeds are first normalized and then given as inputs to a modified Simple K-Means. It considers the number of clusters same as the number of initial seeds. As an output of the Simple K-Means we finally obtain a set of clusters based on both numerical and categorical attributes. We then de-normalize the records and the cluster definitions. A cluster definition is considered to be similar to a logic rule (such as "A1= {a12}, A2 = any value, A3= {a31, a32}, and A4>7"), which defines the boundary conditions of a cluster. Each record belongs to a cluster.

The distance between a record and a seed in the modified Simple K-Means can be calculated in one of the two different ways. The first way is the conventional way where the distance is calculated based on all attributes. We propose a second way where distance can be calculated based on only the significant attributes using their level of significance as follows.

$$dist(x_i, c_j) = \frac{\sum_{a=1}^{n} w_a |x_i - c_j|}{\sum_{a=1}^{n} w_a}. \qquad (3)$$

Here x_i is the ith record, c_j is the centre (seed) of the jth cluster, w_a is the user defined weight (significance level) for an attribute *a*, and *n* is the number of attributes. Therefore, the attributes having zero weight are ignored in the Simple K-Means process. Moreover, the attributes having high weights have more influence in distance calculation than the attributes having low weights. Note that attribute weights used in distance calculation in equation 3 can be different from the attribute weights used in initial seed selection. For initial seed selection a user may want to assign non zero weights on low number of attributes in order to reduce the complexity while getting high quality seed. However, for distance

calculation purpose a user may want to assign non-zero weights on many important attributes. We consider in this approach that a user knows his data set well and therefore, can identify the important attributes and guess appropriate weights. If a user does not know the important attributes he/she can try different weight sets and explore the results.

Computational complexity of Seed-Detective can be greater than an existing technique. However, we consider that it will be used for static data set (not live data stream) where computation time is not a big problem. Moreover, a user may not assign non-zero weights on many attributes in order to reduce the computational time required, if that is necessary. As an extreme example, if a user assigns zero weights on all attributes for initial seed selection purpose then he/she can reduce the complexity for initial seed selection to zero.

If a user assigns inappropriate weights on the attributes (i.e. non-zero weights on insignificant attributes only) we may have a set of poor quality preliminary clusters. In Ex-Detective, a record does not have an opportunity to move to another preliminary cluster even if it suits better in another cluster. The preliminary clusters only get divided into sub clusters due the application of K-Means within each cluster separately. However, since Seed-Detective uses the preliminary clusters just to determine the initial seeds and then allows a record to move to any cluster, it is supposed to produce better quality final clusters even from a set of poor quality preliminary clusters.

The random initial seed selection process of Simple K-Means (SimpleKMeans, Han and Micheline 2006, Haung 1997, K-Means) is a limitation of the technique. Therefore, BFPH and NFPH (He 2006) attempt to improve the seed selection process by choosing seeds that are well separated. However, BFPH still selects the very first seed randomly and NFPH tends to select it as the record having values that are in a way the most common in a data set. The next seed is the farthest point of the first seed, and this approach of selecting the second seed is difficult to justify. Moreover, we argue that the record having the highest score (as it is calculated in NFPH) may not be in the densest area. Besides, the records belonging to a dense area may be close to each other but they may not have the same natural class value. Seed-Detective improves the quality of initial seeds by taking the centers of a set of preliminary clusters. Moreover, unlike Simple K-Means and BFPH Seed-Detective produces the same set of clusters from a data set whenever it is applied on the data set.

4 Experimental Results

We implement Seed-Detective (SD), Ex-Detective (ED), Simple K-Means (SK), Basic Farthest Point Heuristic (BFPH), and New Farthest Point Heuristic (NFPH). Seed-Detective uses modifications of Ex-Detective and Simple K-Means. Therefore, we test Seed-Detective's performance against them in order to justify the modifications made. Moreover, since it was shown in the literature (He 2006) that BFPH and NFPH perform better than Simple K-Means we test Seed-Detective's performance against them as well.

For the experiments here we calculate distance (in Step 3 of Seed-Detective) using all attributes instead of using the significant attributes only as shown in Equation 3. We run the techniques on the two natural datasets, namely Credit Approval and Contraceptive Method Choice (CMC). The datasets are publicly available in UCI Machine Learning Repository (UCI).

The CMC dataset has 1473 records with 10 attributes including the class attribute. There are all together 8 categorical attributes, and 2 numerical attributes. None of the records has any missing values. The Credit Approval dataset has 690 records with 16 attributes including the class attribute. There are 10 categorical attributes, and 6 numerical attributes. We remove all records with one or more missing values, and thereby work on 653 records with no missing values.

Before applying the clustering techniques on a data set we first remove the class attribute. After the clusters are produced we reconsider the original class values of the records in order to evaluate the clustering quality using a few well known evaluation criteria namely F-measure (FM), Purity and Entropy (Chuang 2004, Tan, Steinbach, and Kumar 2006). Note that there are many natural data sets without having class attributes. We remove the class attributes from the data sets that we use in the experiments in order to simulate of the data sets without a class attribute. We then use the actual class attribute to test the cluster quality.

We now define the evaluation criteria briefly (Chuang 2004, Tan, Steinbach, and Kumar 2006). Let $p_{i,j}$ be the probability of a record belonging to cluster *i* having class value *j*, $m_{i,j}$ be the number of records belonging to cluster *i* having class *j*, and m_i be the number of records in cluster *i*. Precision of a cluster *i* with respect to a class value *j*, $precision(i,j) = p_{i,j} = {}^{m_{i,j}}/_{mi}$. Recall of a cluster *i* with respect to class value *j*, $recall(i,j) = {}^{m_{i,j}}/_{m_j}$ where m_j is the number of records having class value *j* in the whole data set. F-measure of a cluster *i* with respect to a class value *j* is calculated as follows.

$$F(i,j) = \frac{2 * precision(i,j) * recall(i,j)}{precision(i,j) + recall(i,j)}. \qquad (4)$$

If for a data set we have *u* distinct class values and discover *v* number of clusters ($u < v$) then we convert *v* number of original clusters into *u* number of clusters by merging all original clusters having the same majority class into one cluster. We then calculate the F-measure of each of the *u* merged clusters with respect to the majority class of the cluster. Finally we calculate the average F-measure of all *u* clusters to indicate the overall quality of all clusters produced by a clustering technique.

Purity of a cluster *i*, $p_i = max_j(p_{i,j})$ i.e. the probability of a record having the majority class in a cluster. Overall purity of all *v* number of clusters, $purity = \sum_{i=1}^{v} \frac{m_i}{m} p_i$, where *m* is the total number of records in a data set. Entropy of a cluster *i*, $e_i = -\sum_{j=1}^{u} p_{i,j}\, log_2 p_{i,j}$. Therefore, the overall entropy for all *v* number of clusters, $e = \sum_{i=1}^{v} \frac{m_i}{m} e_i$. Higher value of F-

		Seed-Detective				Ex-Detective			
Weight		**CLN**	**FM**	**Purity**	**Entropy**	**CLN**	**FM**	**Purity**	**Entropy**
All	All 1	27	0.857	0.858	0.522	52	0.696	0.704	0.776
	All 2	85	0.857	0.859	0.473	144	0.813	0.823	0.522
	All 3	201	0.893	0.894	0.325	256	0.87	0.868	0.356
	All 4	265	0.91	0.911	0.252	315	0.875	0.883	0.316
	All 5	270	0.913	0.914	0.241	320	0.887	0.889	0.289
Best Five	BA 1	27	0.857	0.858	0.522	52	0.696	0.704	0.776
	BA 2	85	0.857	0.859	0.473	144	0.813	0.823	0.522
	BA 3	118	0.882	0.884	0.353	182	0.837	0.845	0.453
	BA 4	176	0.893	0.894	0.324	244	0.863	0.86	0.361
	BA 5	181	0.895	0.896	0.314	247	0.858	0.86	0.366
Worst Four	WA 1	1	0.353	0.546	0.994	3	0.505	0.615	0.937
	WA 2	1	0.353	0.546	0.994	3	0.505	0.615	0.937
	WA 3	6	0.783	0.793	0.718	11	0.526	0.617	0.893
	WA 4	6	0.783	0.793	0.718	11	0.526	0.617	0.893
	WA 5	6	0.783	0.793	0.718	11	0.526	0.617	0.893
Best Two	BT 1	1	0.353	0.546	0.994	3	0.505	0.615	0.937
	BT 2	2	0.799	0.799	0.715	5	0.603	0.612	0.902
	BT 3	4	0.824	0.828	0.655	8	0.742	0.75	0.79
	BT 4	6	0.796	0.806	0.668	12	0.754	0.751	0.775
	BT 5	6	0.796	0.806	0.668	12	0.754	0.751	0.775
Best Two & Worst Two	BW 1	1	0.353	0.546	0.994	3	0.505	0.615	0.937
	BW 2	2	0.799	0.799	0.716	5	0.603	0.612	0.902
	BW 3	6	0.825	0.828	0.65	12	0.747	0.759	0.774
	BW 4	10	0.820	0.827	0.633	19	0.753	0.754	0.766
	BW 5	10	0.820	0.827	0.633	19	0.753	0.754	0.766

Table 1: Evaluation of Seed-Detective and Ex-Detective on Credit Approval dataset

measure and purity indicate better clustering whereas for entropy a lower value indicates better clustering.

In Seed-Detective a user can assign different weights on different attributes. Typically a data miner has good understanding on the data set being used and therefore knows the attributes that need to be given more weights than others in order to get a good clustering output. However, due to various reasons a data miner may not always assign weights on good attributes only. Therefore, in order to simulate different user attitude towards weight assignments we allocate weights on the attributes in different categories. For example, we assign non-zero weights on all attributes (All), only the best attributes (BA), only the worst attributes (WA), best two attributes (BT) and a mixture of the best and worst attributes (BW). Moreover, while assigning non-zero weights on all attributes we use various weight patterns. In "All 1" pattern we assign 0.2 weights and in "All 2" pattern we use 0.4 weights on all attributes. Similarly in "All 3", "All 4" and "All 5" patterns we assign 0.6, 0.8 and 1.0 weights, respectively. In "BA 1" and "BA 2" patterns we assign 0.2 and 0.4 weights, respectively on the best attributes. In the experiments of this study we only assign non-zero weights on categorical attributes. In this study All, BA and WA are called weight categories whereas All 1, and All 2 are called weight patterns.

We now describe the process of discovering the best and the worst attributes. We assign full weight (1.0) on the attributes one at a time. That is, in one arrangement we assign full weight on an attribute say A1 and zero weight to all other attributes. Similarly in another arrangement we assign full weight on another attribute say A2 and zero weight to all other attributes. We then produce clusters using Seed-Detective for all arrangements separately. We calculate the average F-measure of all the clusters for an arrangement. The attribute having the full weight for the arrangement that produces the best average F-measure is considered as the best attribute. This way we divide the attributes into two equal groups where in one group we have all the attributes that are better than all attributes in the other group. Attributes in the former group are called the best attributes (BA) and attributes in the later group are called the worst attributes (WA). In BW category non-zero weights are assigned on two best and two worst attributes.

We then use different weight patterns in Seed-Detective and Ex-Detective (see Table 1). Ex-Detective generally produces more clusters than Seed-Detective for any weight pattern. Having more clusters can result in better F-measure, Purity and Entropy for a clustering. Therefore, while comparing two clustering techniques we need to match up the number of clusters as well.

For similar number of clusters Seed-Detective produces significantly better F-measure (FM), Purity and Entropy than those produced by Ex-Detective. For example, in Credit Approval data set for "All 2" weight pattern (Table 1) Seed-Detective produces 85 clusters (CLN = 85) while Ex-Detective produces 140 clusters. However, Seed-Detective still produces better result than Ex-Detective. We also get similar result for CMC data

	F-measure (FM)				Purity				Entropy			
Weight pattern	**SD**	**NFPH**	**BFPH**	**SK**	**SD**	**NFPH**	**BFPH**	**SK**	**SD**	**NFPH**	**BFPH**	**SK**
All	**.886[3]**	**.880[1]**	**.889[4]**	**.883[2]**	**.887[3]**	**.882[1]**	**.889[4]**	**.884[2]**	**.362[2]**	**.371[1]**	**.343[4]**	**.354[3]**
Best Five	**.877[4]**	**.867[1]**	**.874[3]**	**.872[2]**	**.878[4]**	**.869[1]**	**.875[3]**	**.873[2]**	**.397[4]**	**.425[1]**	**.398[3]**	**.399[2]**
Worst Four	**.611[2]**	**.617[4]**	**.611[2]**	**.615[3]**	**.694[3]**	**.694[3]**	**.690[1]**	**.695[4]**	**.828[2]**	**.827[3]**	**.828[2]**	**.818[4]**
Best Two	**.714[4]**	**.706[3]**	**.684[2]**	**.684[2]**	**.757[4]**	**.745[3]**	**.726[1]**	**.727[2]**	**.740[4]**	**.772[3]**	**.793[1]**	**.787[2]**
Best Two & Worst Two	**.723[4]**	**.709[3]**	**.689[1]**	**.697[2]**	**.765[4]**	**.749[3]**	**.730[1]**	**.739[2]**	**.725[4]**	**.765[2]**	**.772[1]**	**.762[3]**
Total:	**17**	**12**	**12**	**11**	**18**	**11**	**10**	**12**	**16**	**10**	**11**	**14**

Table 2: Evaluation on Credit Approval dataset

	F-measure (FM)				Purity				Entropy			
Weight pattern	**SD**	**NFPH**	**BFPH**	**SK**	**SD**	**NFPH**	**BFPH**	**SK**	**SD**	**NFPH**	**BFPH**	**SK**
All	**.479[3]**	**.484[4]**	**.466[1]**	**.470[2]**	**.499[3]**	**.495[2]**	**.490[1]**	**.502[4]**	**1.372[4]**	**1.385[2]**	**1.385[2]**	**1.383[3]**
Best Four	**.455[4]**	**.454[3]**	**.447[2]**	**.444[1]**	**.474[2]**	**.476[3]**	**.471[1]**	**.479[4]**	**1.426[4]**	**1.431[1]**	**1.430[2]**	**1.428[3]**
Worst Three	**.399[4]**	**.363[3]**	**.348[1]**	**.354[2]**	**.461[4]**	**.444[1]**	**.447[2]**	**.450[3]**	**1.470[4]**	**1.486[1]**	**1.478[3]**	**1.479[2]**
Best Two	**.401[3]**	**.405[4]**	**.392[1]**	**.398[2]**	**.460[4]**	**.457[2]**	**.454[1]**	**.460[3]**	**1.448[4]**	**1.459[1]**	**1.457[3]**	**1.458[2]**
Best Two & Worst Two	**.426[3]**	**.436[4]**	**.415[1]**	**.422[2]**	**.486[4]**	**.472[2]**	**.470[1]**	**.480[3]**	**1.410[4]**	**1.418[3]**	**1.424[1]**	**1.423[2]**
Total:	**17**	**18**	**6**	**9**	**17**	**10**	**6**	**17**	**20**	**8**	**11**	**12**

Table 3: Evaluation on Contraceptive Method Choice dataset

Dataset	**Seed-Detective (SD)**	**NFPH**	**BFPH**	**Simple K-Means**
Credit Approval	**51**	**33**	**33**	**37**
CMC	**54**	**36**	**23**	**37**
Total:	**105**	**69**	**56**	**74**

Table 4: Score Comparison (Summary of Result)

Weights	**Seed-Detective**		**Ex-Detective**		**NFPH**		**BFPH**		**Simple K-Means**	
	CA	CMC	CA	CMC	CA	CMC	CA	CMC	CA	CMC
All1	15611	175884	10586	175332	11623	4510	8113.4	4405.1	6372.9	4061.5
BA1	14401	45401	10586	175332	11623	4510	8113.4	4405.1	6372.9	4061.5
WA1	1396	131302	1110	131167	308	134	207.1	72.7	199.3	71.1
BT1	869	20924	583	19832	308	1482	207.1	1282.1	199.3	1297.9
BW1	1665	162134	1379	160981	308	1482	207.1	1282.1	199.3	1297.9

Table 5: Computational Time in milliseconds on Credit Approval (CA) and CMC dataset

set. Although Ex-Detective was proposed in the past (Islam and Brankovic 2011, Islam 2008) no experimental result was published on Ex-Detective. Therefore, the experimental result on Ex-Detective is also one of the contributions of this study.

We next apply Simple K-Means (SK), Basic Farthest Point Heuristic (BFPH) and New Farthest Point Heuristic (NFPH) on the data sets. SK, BFPH and NFPH require a desired number of clusters as inputs. We use the number of clusters produced by Seed-Detective as the user defined number of clusters for SK, BFPH and NFPH. Since Simple K-Means and BFPH (if they are applied more than once) can produce different sets of clusters, we run them ten times for each weight pattern such as "All 2" and calculate the average of the F-measures, purities and entropies.

Seed-Detective has some additional complexity as it combines the modified Ex-Detective and Simple K-Means. We therefore test whether Simple K-Means, BFPH and NFPH can result in better clustering even if they are allowed to use unlimited number of iterations in order to balance the additional complexity required by Seed-Detective. We set the maximum number of iteration very high and ensure that the techniques never terminate due to a limited number of iterations allowed. In Table 5 we present computational time required by Seed-

Detective, Ex-Detective, NFPH, BFPH and Simple K-Means on Credit Approval and CMC dataset. We have calculated the computational time for weight patterns All1, BA1, WA1, BT1 and BW1. We use Intel (R) Core (TM) i5 CPU M430 2.27 GHz processor with 4 GB RAM in this experiment.

In Table 2 and Table 3 we present average F-measure, Purity and Entropy of all weight patterns for a weight category. We then use a scoring rule to assign 4 points to the best, 3 points to the 2nd best, 2 points to the 3rd best and 1 point to the worst F-measure. We also assign similar points for Purity and Entropy. The points are shown within parenthesis in the tables. For Credit Approval data set Seed-Detective scores the highest total among all techniques for all three evaluation criteria; F-measure, Purity and Entropy. Seed-Detective also scores the highest for all three evaluation criteria for "Best Five", "Best Two" and "Best Two and Worst Two" weight categories (Table 2). This indicates that for the Credit Approval data set Seed-Detective performs well if a user assigns higher weights on the best attributes.

In this approach we assume that a user has good understanding on the schematic design of his/her data set. Note that we do not assume that a user already knows the clusters, rather we assume that a user knows which attributes can be more interesting to use for clustering records. For example, if a data set has many attributes such as "Pump Manufacturer", "Name of the Pump Installer", "Pump Age", "Pump Type (such as reciprocal and centrifugal)", and "Pump Location (such as underground and over ground)" then a user may want to cluster the records using more weights on the last three attributes than on the 1st two attributes since he would naturally know the significance of the attributes in exploring say "possibility of pump failure". If he/she does not have this understanding he/she may want to assign same weight to all attributes or explore different weight combinations. Our technique provides more opportunity to play with the data set and explore different patterns of clustering.

For the CMC data set (Table 3) Seed-Detective scores the highest (20 out of 20) on the basis of Entropy test. It also scores the highest on the basis of Purity and the second highest on the basis of F-measure for the CMC data set.

In Table 4 we present the total score of the techniques on the two data sets. Seed-Detective (SD) clearly scores the best among all techniques for both data sets.

5 Conclusion

In this study we have presented a novel clustering technique that uses a combination of modified Ex-Detective and Simple K-Means in order to take advantage of high quality initial seeds for better clustering. Our clustering technique also allows a data miner to assign weights on the attributes (both categorical and numerical) and thereby completely or partially consider a set of attributes while clustering records. We have compared our technique with several existing techniques namely Ex-Detective, Simple K-Means, Basic Farthest Point Heuristic, and New Farthest Point Heuristic based on several evaluation criteria such as F-measure, purity and entropy. Our initial experimental result strongly indicates a superiority of the novel technique over the existing techniques. A limitation of our novel technique can be its complexity. However, we consider that the technique can be useful for static data sets where complexity may not a big problem. Moreover, a user can always adjust the complexity through suitable weight assignments on the attributes. Our future research plans include further development of the technique to reduce complexity and increase usefulness. We also plan to test the technique with more existing techniques using additional evaluation criteria.

6 References

Antonellis, P., Antoniou, D., Kanellopoulos, Y., Makris, C., Theodoridis, E., Tjortjis, C., Tsirakis, N. (2009): Clustering for Monitoring Software Systems Maintainability Evolution. In Electronic Notes in Theoretical Computer Science, vol. 233, pp. 43-57.

Chuang, K. T. and Chen, M. S. (2004): Clustering Categorical Data by Utilizing the Correlated-Force Ensemble. Proc. 4th SIAM Intern'l Conference on Data Mining (SDM-04).

Chui-Yu, C., Yi-Feng, C., I-Ting, K.., He, C. K. (2008) : An intelligent market segmentation system using k-means and particle swarm optimization. Journal of Expert Systems with Applications.

Grubesic, T. H. and Murray, A. T. (2001): Detecting Hot Spots Using Cluster Analysis and GIS. Proc. 5th Annual International Crime Mapping Research Conference, Dallas, TX.

Han, J. and Micheline, K. (2006): Data Mining Concepts and Techniques, 2nd ed., San Francisco, Morgan Kaufmann.

Haung, D. and Pan, W.(2009): Incorporating biological knowledge into distance-based clustering analysis of microarray gene expression data. Proc. 6th International conference on Fuzzy systems and knowledge discovery, vol. 5, pp. 52-56.

Haung, Z. (1997): Clustering Large Data Sets with Mixed Numeric and Categorical Values. Proc. First Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 21-34, Singapore.

He, Z. (2006): Farthest-Point Heuristic based Initialization Methods for K-Modes Clustering. In Computing Research Repository, vol. abs/cs/0610043.

Islam, M. Z. and Brankovic, L. (2005): DETECTIVE: A Decision Tree Based Categorical Value Clustering and Perturbation Technique in Privacy Preserving Data Mining. Proc. 3rd International IEEE Conference on Industrial Informatics, Perth.

Isam, M. Z. (2008): Privacy Preservation in Data Mining through Noise Addition. Ph.D. thesis. in Computer Science, School of Electrical Engineering and Computer Science, The University of Newcastle, Australia.

Islam, M. Z. (2010): Explore: A Novel Decision Tree Classification Algorithm. Proc. 27th Internation Information Systems Conference, British National

Conference on Databases (BNCOD 2010), Springer LNCS, Vol. 6121, In press, Dundee.

Islam, M. Z. and Brankovic, L. (2011): Privacy Preserving Data Mining: A Noise Addition Framework Using a Novel Clustering Technique. Journal of Knowledge-Based Systems, in press.

Jian, F. C., Heung, S. C. (2011): Applying agglomerative hierarchical clustering algorithms to component identification for legacy systems. Journal of Information and Software Technology.

K-Means: K-Means Clustering in WEKA,

http://maya.cs.depaul.edu/classes/ect584/WEKA/k-means.html. Accessed 20th Nov 2010.

Quinlan, J. R. (1993): C4.5: Programs for Machine Learning. San Francisco, Morgan Kaufmann.

Quinlan, J. R. (1996): Improved Use of Continuous Attributes in C4.5. Journal of Artificial Intelligence Research, Vol. 4, pages 77-90

SimpleKMeans: SimpleKMeans,

http://weka.sourceforge.net/doc.dev/weka/clusterers/SimpleKMeans.html. Accessed 20th Nov 2010.

Songa, J. and Nicolae, D. L. (2008): A sequential clustering algorithm with applications to gene expression data. Journal of Korean Statistical Society.

Tan, P. N., Steinbach, M. and Kumar, V. (2006): Introduction to Data Mining. Addison-Wesley, Boston.

UCI: UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/. Accessed 15th Oct 2010.

Wal-Mart: Wal-Mart Deploys Data Mining Software Into Its Production Support Environment http://walmartstores.com/pressroom/news/4008.aspx. Accessed 5th Dec 2010.

OO-FSG: An Object-Oriented Approach to Mine Frequent Subgraphs

Bismita Srichandan **Rajshekhar Sunderraman**

Department of Computer Science
Georgia State University
Atlanta, USA
Email: bsrichandan1@student.gsu.edu, raj@cs.gsu.edu

Abstract

Frequent subgraph mining (FSG) has always been an important issue in data mining. Several frequent subgraph mining methods have been developed for mining graph data. However, most of these are main memory algorithms in which scalability is a bigger issue. A few algorithms have opted for a relational approach that stores the graph data in relational tables. However, relational databases have their own space as well computing constraints when it comes to storing large databases. Moreover, relational databases do not preserve semantic information as they represent simple entities and in order to preserve the relationship between two entities additional tables are necessary. Object-oriented databases, on the other hand, do not have these constraints. In this paper, we present an object-oriented database approach to mining frequent sub-graphs. We use Db4o, a popular open-source object database system, to store the input graph data as well as intermediate results. Db4o can save all the information about an entity in a single class in an object form. Application domains such as protein-protein interaction data, social network data, and chemical compound structure data require mining frequent subgraphs while preserving the meaning. This paper proposes a novel idea for using object oriented database db4o to store graph data, which can support large graph data as well as preserve semantic information.

Keywords: Data Mining, Frequent Subgraphs, Object Oriented Database

1 Introduction

There are many efficient algorithms for finding frequent itemsets in very large transaction databases (Agrawal et al. 1994, Agarwal et al. 1998, Zaki et al. 2001, Han et al. 2000). We can use these itemsets for discovering association rules, for extracting prevalent patterns that exist in the datasets, or for classification. However, we can't apply these techniques over datasets which are not itemsets. In recent years, there has been an increased interest in developing data mining algorithms that operate on graphs. Such graphs arise naturally in a number of different application domains, including computer networks, bioinformatics, semantic web, chemical compound and social networks. All these domains mentioned require mining of frequent subgraphs over large data sets. However, the algorithms developed so far are not scalable. Most works done on frequent subgraph mining have focused on algorithms that assume graph data is stored in main memory. Memory dependent algorithms could be very inefficient if the dataset is large. In this paper we propose to store the input graph data set as well as intermediate results in an object-oriented database and extend the FSG mining algorithms to work with object-oriented databases. This approach scales nicely for large data sets that cannot fit in main memory. We also show that using object-oriented databases over relational systems had an advantage in performance and scalability.

2 Related Work

We now discuss related work dealing with both in-memory as well as disk storage algorithms.

2.1 In-memory Methods

Cook et al. (2000) proposed SUBDUE to discover the best compressing structures. Inokuchi et al. (2003) proposed an Apriori based algorithm to discover all frequent substructures. Kuramochi et al. (2001) proposed FSG algorithm which represents graphs as sparse adjacency matrix and uses canonical labeling to determine subgraph isomorphism. Han et al.(2002) proposed gSpan algorithm which uses the depth-first search and generates lesser candidate items than FSG. Kuramochi et al. (2005) proposed an algorithm to find frequent patterns from a large sparse graph. Jiang et al. (2009) tried to find globally frequent subgraphs on a single labeled graph. All the above mentioned algorithms are not scalable because when the graph data is too large, it consumes the memory completely and reduces the efficiency.

2.2 In-disk method, DB-FSG

Chakravarthy et al. (2004) proposed an algorithm DB-Subdue which is the first attempt to implement the database approach for graph mining. Following DB-Subdue, Chakravarthy et al. (2008) proposed an SQL-based approach for frequent subgraph mining (DB-FSG). In their work, they have used relational tables to store graph data and subgraphs. Their approach is briefly as follows: the method has two tables to begin with: one for the vertices and other for the edges which contain individual vertices and edges. The individual tables are joined to obtain size-1 subgraphs. Each time the candidates are generated, the columns in the table grow depending on the size of the graphs. This will eventually place a limit on the size

of the maximum substructure that can be detected, as there is a limit on the number of columns a relation can have in a relational database. The algorithm described in DB-FSG can discover substructures of size 165 at the most. After implementing their algorithm, we figured out that it poses many difficulties for larger datasets, and efficiency is the major drawback when the dataset size is large.

Another issue related to relational database is storing semantic information. As graph databases, like chemical compounds and protein-protein interactions, have explicit relations between elements, semantic information must be taken care of by the data model.

We implemented the algorithm by Chakravarthy et al. (2008) using the same datasets to analyze the pros and cons of both approaches. The algorithm is as follows:

Algorithm 1 DB-FSG - Chakravarthy et al. (2008)

Input: A graph dataset Gs and min_*sup*
Output: The frequent subgraph set S
Method:

1. Create oneedge (instance_1) table by joining vertex table and edge table
2. Remove the edges with instance count less than support from the oneedge table
3. for n=2 to MaxSize do
   (a) Join instance (n-1) with oneedge table to generate instance n
   (b) Eliminate pseudo duplicates from instance n table
   (c) Canonically order instance n table on vertex labels
   (d) Project distinct vertex label, edge label and gid to obtain one instance per substructure for each graph and store in dist n table.
   (e) Group dist n table by vertex label and edge label to obtain substructures and its count
   (f) Retain only the instances of substructure satisfying support and store it in instance n table
   (g) If there are no instances of substructure satisfying support then stop.
4. end for

The rest of the paper is organized as follows. Section 3 describes the object-oriented approach. Section 4 describes the OO-FSG algorithm. Section 5 shows the experimental evaluation. Section 6 concludes the paper and points some of the future works in progress.

3 An OO-approach to Mine FSGs

We are proposing to use db4o, an object-oriented database (http://www.db4o.com/) to store the graph dataset. db4o stores everything as objects. The advantage of using db4o as the data storage is because it's highly scalable and do not put burden on memory. To begin with, our approach includes the following basic classes: Vertex, Edge, SingleEdge and Subgraph_1 shown in Fig. 1. The classes are extended as the size of subgraphs increase. For example, for size-2 subgraphs, TwoEdge and Subgraph_2 classes are used. Note that the paper focuses on directed labeled graphs where the direction is assumed to be from a smaller vertex number to the larger vertex number. For example, if the vertices are given the numbers as 0, 1, 2, 3 etc., then the direction of the edges are considered to be from 0 to 1, 1 to 2 or 2 to 3 but not 3 to 1. Hence our method does not need any specific field to keep track of direction between the vertices.

The Vertex class represents nodes in the graph. In the Vertex class, each object has a unique object identity which is 'VertexNo' and label as 'VertexLabel'. Fig. 2 shows a simple subgraph where the numbers 1 and 2 represent vertex numbers which are allocated for ease of use, but these do not have any significance for subgraph mining. A and B represent labels of the vertices and C is the edge label. Each vertex object represents a node of the given graph; the Edge class is similar to the Vertex class. Each edge object represents an edge of the given graph. The SingleEdge class is the combination of the Vertex and Edge classes. It contains all the single-edged subgraphs. The Vertex and Edge classes are constructed separately as the Edge class does not contain the label details of the vertices. The Subgraph_1 class includes all the subgraphs satisfying minimum support which is described in the subsequent sections. The Subscript '1' in Subgraph_1 represents the size-1 subgraphs. As the sizes of the subgraphs increase the subscript changes.

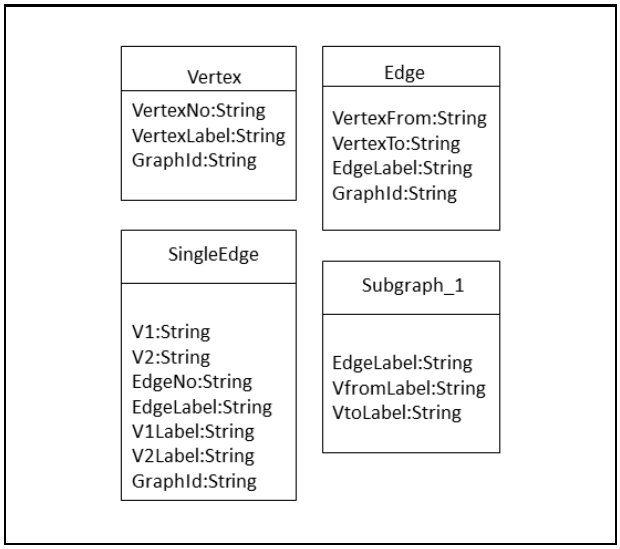


Figure 1: All major classes

Figure 2: An Example Subgraph

In the following sub-sections, we now present the subgraph construction and FSG determination, optimization techniques used in the object-oriented approach, and comparison of DB-FSG with OO-FSG with respect to implementation. Experimental comparison comes later in the paper.

3.1 Sub-graph Construction and FSG Determination

In this sub-section, we discuss the sub-graph construction process and determination of FSGs. We start with a definition.

Definition 1: A *labeled graph* is represented by a 4-tuple, G = (V, E, L, l), where

V is a set of vertices (or nodes)

E $\subseteq$ V $\times$ V is a set of edges, they can be directed or undirected

L is a set of labels

l: V $\cup$ E $\rightarrow$ L, l is a function assigning labels to the vertices and the edges

3.1.1 Sub-graph Support

In Fig. 3, three transaction graphs are shown. The numbers 0, 1, 2 and 3 are the numbers assigned for programming purpose. The labels (names) A, B, C, D, E, F and G are important to the algorithm. Let *nGraph* be the total number of graphs in the dataset and *nSubGraph* be the number of times a particular sub-graph appears in the dataset. Then, the support 'Sup' of a particular subgraph is defined as:

$$Sup = nSubGraph \div nGraph$$

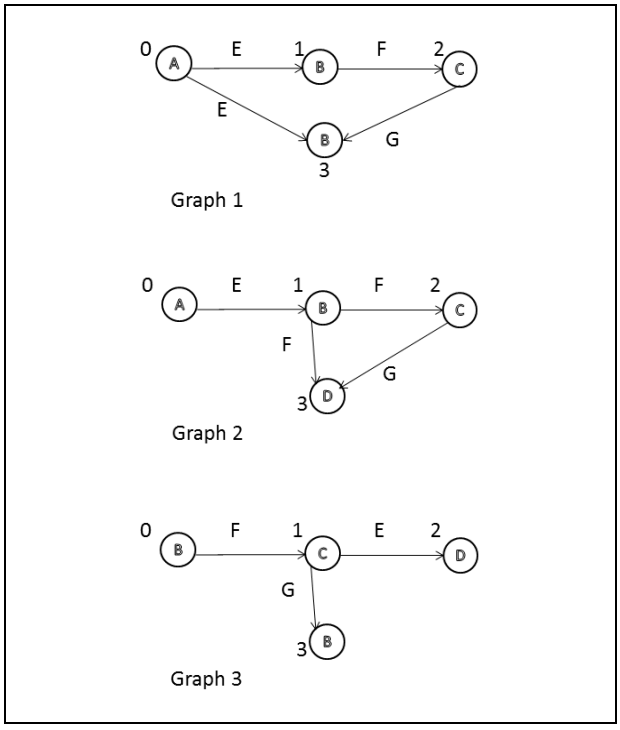


Figure 3: Representation of graphs in the dataset

Graph isomorphism problem needs to be tackled while counting the support of subgraphs in the dataset. Two instances are isomorphic if the vertex and edge labels are same and directions are same. In our experiment, the direction is assumed to be from the lower numbered vertex to the higher numbered vertex. For example, if we count the number of occurrences of subgraph A-E-B in the three graphs, the count is 3, but in reality it is 2. Graph 1 contains the subgraph A-E-B twice. That must be counted once. This problem is eliminated by finding the distinct subgraphs per graph. Note that though we count only one instance of the subgraph per graph, we do not discard the other instances before pruning. The problem could be the instance omitted might have significance in the discovery of the subgraph of size 2. If we remove the pair 0-1 (vertex numbers) from graph 1 instead of 0-3, then in the next level, construction of subgraph_2 would not generate the subgraph A-B-C (0-1-2). So we store the other instances too.

3.1.2 Subgraph construction

This section elaborates the process of subgraph construction. To begin with, we save the vertices and edges in different classes named Vertex and Edge classes. Since Edge class does not have information on the labels of the vertices, we join the Vertex and Edge classes based on the vertex numbers and the graph id to create the size-1 subgraphs stored in SingleEdge class. The graph id must be same during joining as the expansion happens in the same graph. SingleEdge class contains all the information on the vertices and edge labels. Considering the graphs shown in Fig. 4, the size-1 edges are shown in the diagrams. In order to provide a detailed view of the relational method (Chakravarthy et al., 2008) and object-oriented method, we have provided the tables along with the graph structures. Subsequent stages of construction are also shown in the figures. The edges are assigned a number to keep track of the edges joined during candidate generation. Actually, they are stored as objects in the db4o database.

Table 1: Vertex Table

VertexNo	VertexLabel	GraphId
0	A	1
1	B	1
2	C	1
3	B	1
0	A	2
1	B	2
2	C	2
3	D	2
0	B	3
1	C	3
2	D	3
3	B	3

Table 2: Edge Table

V1	V2	EdgeLabel	GraphId
0	1	E	1
0	3	E	1
1	2	F	1
2	3	G	1
0	1	E	2
1	2	F	2
1	3	F	2
2	3	G	2
0	1	F	3
1	2	E	3
1	3	G	3

Table 3: SingleEdge Table

V1	V2	ENo	ELabel	V1L	V2L	GId
0	1	1	E	A	B	1
0	3	2	E	A	B	1
1	2	3	F	B	C	1
2	3	4	G	C	B	1
0	1	5	E	A	B	2
1	2	6	F	B	C	2
1	3	7	F	B	D	2
2	3	8	G	C	D	2
0	1	9	F	B	C	3
1	2	10	E	C	D	3
1	3	11	G	C	B	3

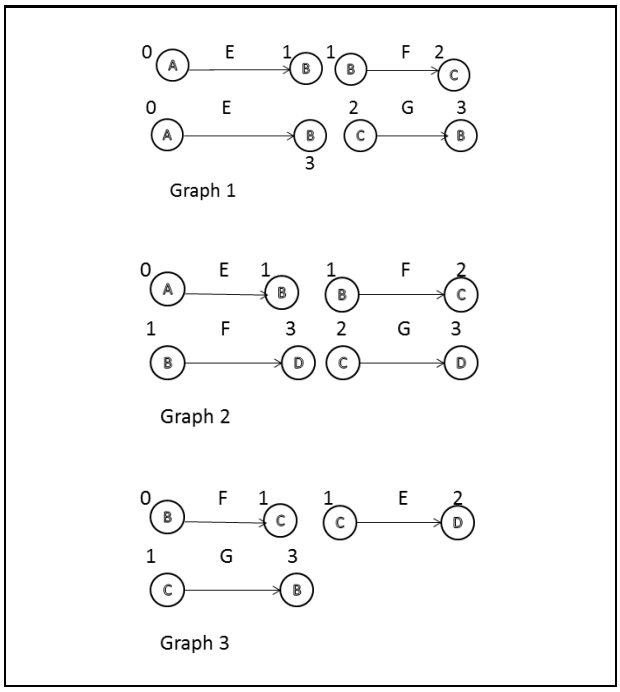


Figure 4: Objects of SingleEdge class

Table 1 contains all the objects in the Vertex class where each row represents the individual objects of the Vertex class. Vertex and Edge classes are not shown in graphical format. Similarly, Table 2 has all the objects which are individual edge objects. After joining the Vertex and Edge classes, we obtained the SingleEdge class shown in Fig. 4. In order to generate size-2 subgraphs, each object of SingleEdge class is joined with itself where V2 of the first edge is same as the V1 of the other object. In all cases, joining happens within the same graph. Unlike, relational databases where there is a defined join query using SQL; db4o does not have such join queries. Instead, it supports a query called 'Native Query' which constrains the class to be joined and has a keyword called 'descend' which goes down to the field level to query the data.

The TwoEdge class is shown in Fig. 5. The notations in the corresponding TwoEdge table are squeezed to fit more columns and are as follows: E1L-label of edge 1, E2L-label of edge 2, V1L-label of vertex 1, V2L-label of vertex 2, V3L-label of vertex 3. The ThreeEdge class is constructed from TwoEdge and SingleEdge classes. Note here that the SingleEdge, TwoEdge and ThreeEdge classes at this stage, do not contain the pruned data, and these are the possible sub-structures before considering pruning. An example of the states of the SingleEdge and TwoEdge classes after pruning is shown in section 4 under pruning. In order to construct the three-edge subgraphs, the query searches for the match for the V3 node from table 4 with the node V1 from table 3 with the same graph id. ThreeEdge class is shown in Fig 6.

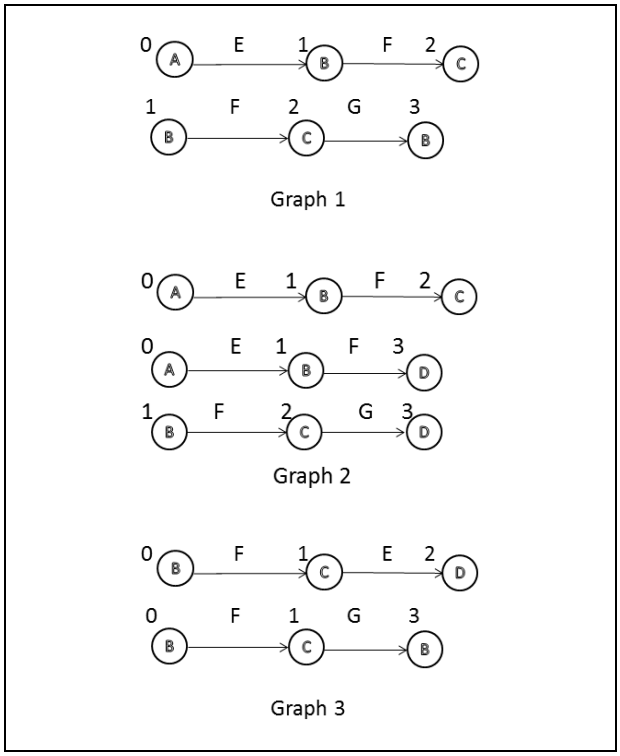


Figure 5: Objects of TwoEdge class

Table 4: TwoEdge Table

V1	V2	V3	E1L	E2L	V1L	V2L	V3L	GId
0	1	2	E	F	A	B	C	1
1	2	3	F	G	B	C	B	1
0	1	2	E	F	A	B	C	2
0	1	3	E	F	A	B	D	2
1	2	3	F	G	B	C	D	2
0	1	2	F	E	B	C	D	3
0	1	3	F	G	B	C	B	3

Table 5: ThreeEdge Table

v1	v2	v3	v4	e1L	e2L	e3L	v1L	v2L	v3L	v4L	GId
0	1	2	3	E	F	G	A	B	C	B	1
0	1	2	3	E	F	G	A	B	C	D	2

3.1.3 Determining frequent subgraphs

Frequent subgraphs are determined based on the number of times it appears in the whole dataset. If we consider the single-edge subgraphs shown in Fig 4, there are eleven of them, but AEB (Graphs 1 and 2), CGB (Graphs 1 and 3) and BFC (Graph 1, 2

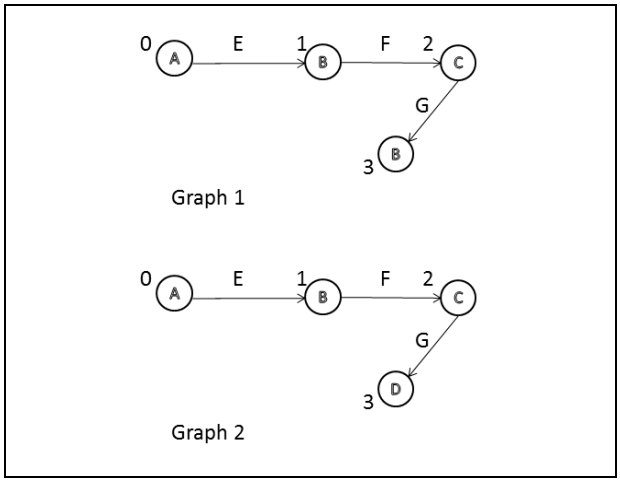


Figure 6: Objects of ThreeEdge class

and 3) appear more than once. Hence it is obvious that the other subgraphs except AEB, CGB and BFC are insignificant. Our purpose is to find the subgraphs which occur more than a specific number of times (min_*sup* provided by the user) in the dataset. Let's consider the minimum support as 2, which mean a subgraph must be appearing in at least two graphs. Subgraph_1 class contains the following size-1 subgraphs shown in the Fig 7. Notice that the Subgraph_1 class does not have the numbers of the vertices. Only significant details, the labels are stored. Fig. 8 shows the frequent subgraphs (minimum support-2) of size-2.

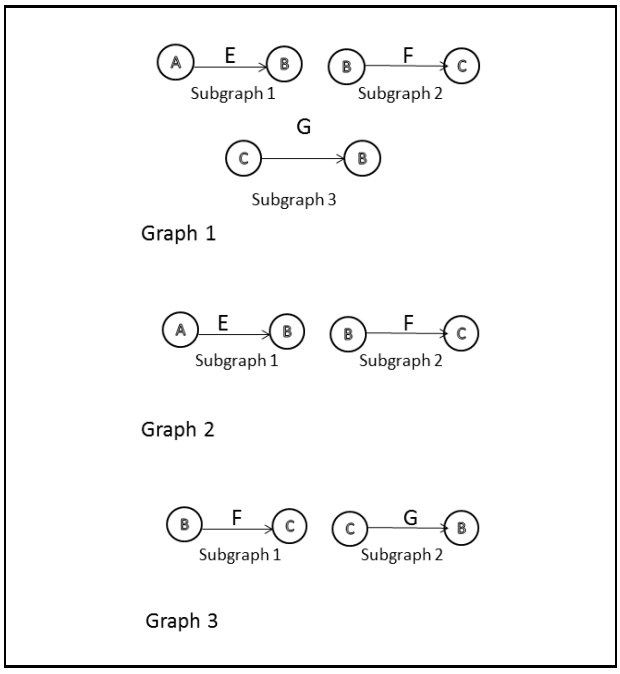


Figure 7: Objects of Subgraph_1 class

3.2 Optimization Techniques

This section discusses various optimization techniques used in object-oriented approach. We have used available data structures across the application to avoid frequent querying of the object database and hence increasing efficiency. Though data structures are used to make the processes faster, the applications are independent of each other, in other words, the graph dataset is always in db4o store. In order to retrieve the distinct instances (to tackle graph isomorphism), we used the data structure "hash sets". In many places, common Java data structures are used to make application process faster. Subgraph counting time has been dramatically improved by using "MultiKey" and "MultiValueMap" common collections data structure available from apache.org. MultiKey can store the same sub-structure instances more than once; in other words, the keys do not need to be unique. For example, considering Fig. 7 we can save the sub-structure A-E-B from both the graphs 1 and 2.

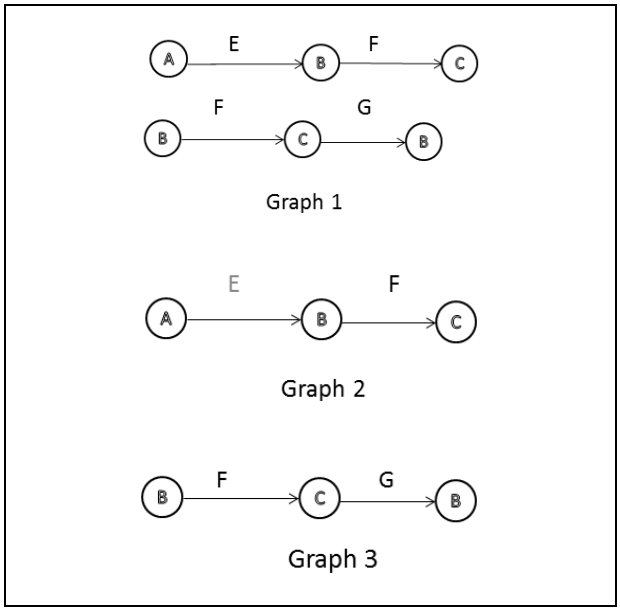


Figure 8: Objects of Subgraph_2 class

3.3 DB-FSG vs OO-FSG: Implementation

The coding of Algorithm 1 was done in Java using Oracle 11g. The tables have the same name as the classes in db4o. We tried to optimize the relational method as much as possible by using indexes, and prepared statements for the insert statements. We noticed a significant time delay while inserting millions of records, whereas in db4o approaches it takes significantly less time. Initial data loading was quite time consuming, so we used Perl script to minimize the time by separating raw input data to Vertex, Edge and SingleEdge files to load into the relational database. For small sized datasets, the efficiency of relational and db4o approach are nearly same, but as the dataset size increases, the performance of db4o over relational increases dramatically. The only problem with db4o approach is it needs strong programming skills whereas relational approach solves things with simple queries. But, at the same time manipulating millions of database records through queries has a huge drag on efficiency. Also the join queries of SQL get messy when we join more than 2 tables. The queries of db4o database are quite simple. A comparison of both queries is given below.

The following is an example of a query used to join the matching vertices of the SingleEdge class/table with itself in order to obtain TwoEdge class/table objects/rows. In the SODA (db4o query) query the "vertexFrom" from the SingleEdge class is joined with the matching "vertexTo" of the same SingleEdge

class.

In the second statement of the code snippet the keyword "constrain" constrains the SingleEdge class. In the third statement the "descend" keyword means starting from the class level the query goes down to one level to the "VertexFrom" field and "constrain" keyword is used to match the "VertexTo" of the SingleEdge class. VertexFrom and VertexTo are the fields in the SingleEdge class and they are named so to indicate the direction of the edges. The last statement executes the query which retrieves all the matching objects in the class. We have also shown the SQL version of the query.

db4o version of a join query

```
query = db.query();
query.constrain(DB4OSingleEdge.class);
query.descend("VertexFrom")
         .constrain(VertexTo)
         .and(GraphId.constrain(GraphId));
query.execute();
```

SQL version of the same query

```
select distinct
  I1.VertexFrom as V1,
  I1.VertexTo as V2,
  I2.VertexTo as V3,
  I1.EdgeNo as E1No,
  I2.EdgeNo as E2No,
  I1.EdgeLabel as E1Label ,
  I2.EdgeLabel as E2Label,
  I1.VfromLabel as V1Label,
  I1.VtoLabel as V2Label,
  I2.VtoLabel as V3Label,
  I1.GraphId
from
  oneedge I1, oneedge I2
where
  I1.VertexTo = I2.VertexFrom and
  I1.GraphId = I2.GraphId;
```

4 Details of OO-FSG Algorithm

OO-FSG algorithm has two major aspects. One is generating candidates and another one is pruning the insignificant edges from the graphs. Each step of the algorithm is discussed in detail. In the algorithm, first step is for the construction of SingleEdge class from Vertex and Edge classes. In the second step, the distinct single edges are separated to get rid of isomorphic structures and stored in Subgraph_1 class. Counting of the distinct edges is done using MultiKey and MultiValueMap on the whole dataset with the user provided minimum support(min_*sup*). In the third step, we remove the edges which fail to satisfy the minimum support value from the SingleEdge class. Step 4 is the looping condition, looping occurs from steps 4 (a) through 4 (e) until size-n which is 5 for our experiment. Step 4 (a) combines the SingleEdge class with itself based on the matching vertices and graph id. Step 4 (b) removes the redundant subgraphs to find the distinct instances and stores in the temporary class Subgraph_*Distinct*_2 class. In Step 4 (c), we count the subgraphs. When we say subgraphs, means only the edge labels and vertex labels not the numbers given to the nodes and edges. Steps 4 (d) and 4 (e) are self-explanatory. In the second iteration of the loop, we combine TwoEdge class with SingleEdge class and follow the steps accordingly. We keep repeating the loop until we get a subgraph of size-5.

Algorithm 2 OO-FSG Algorithm

Input: A graph dataset Gs and min_*sup*
Output: The frequent subgraph set S
Method:

1. construct SingleEdge class by joining Vertex and Edge class.
2. select distinct single edges and store the subgraphs which satisfies min_*sup* in Subgraph_1 class.
3. remove the edges with count less than the min_*sup* from SingleEdge.
4. repeat steps a through e until a candidate subgraph of size-N with min_*sup* is generated.
   (a) join (N-1)Edge class with SingleEdge class to generate *(N)Edge.
   (b) eliminate the redundant subgraphs from (N)Edge and store the size-N subgraphs in Subgraph_*Distinct*_*N* class.
   (c) count the unique vertex and edge labels in the Subgraph_*Distinct*_*N* class.
   (d) eliminate the subgraphs from Subgraph_*Distinct*_*N* with count less than min_*sup* and store it in Subgraph_*N* class.
   (e) remove the edges with count less than min_*sup* from (N)Edge class.
5. end loop.

*(N)Edge: represents the TwoEdge, ThreeEdge, FourEdge and FiveEdge classes etc.

Candidate generation : this process is same as the subgraph construction described in section 3. First time the SingleEdge class is combined with itself. In subsequent iterations it is combined with TwoEdge, ThreeEdge and FourEdge classes as we are running the loop until size-5 subgraphs are generated.

Frequency counting and Pruning: The subgraphs from the Subgraph_*Distinct*_1 class are searched for frequency counting on the vertex labels, edge labels. The subgraphs which meet the support value (user defined) are stored in a class called Subgraph_1. Edges are retained in the SingleEdge class where there is a matching; all other edges are pruned from the SingleEdge class. An example of pruning would be good here to understand the process. If we consider the Figures 4 and 5, the states are shown before pruning. Let's assume the minimum support provided by the user is 2, i.e. a sub-structure must be appearing in at least two graphs. The states of the SingleEdge and TwoEdge classes after pruning are shown in Fig. 9 and 10 respectively.

5 Experimental Details

The experiments were conducted on a Linux machine with 2 GB memory. The OO-FSG algorithm was coded in Java. The experimental results are shown in Table 6 as well as in graphical format. The graphs in the figures 11, 12, 13 and 14 show the efficiency comparison between DB-FSG and OO-FSG w.r.t minimum supports 1, 3, 5 and 7 respectively. We observed that using db4o database, efficiency is much

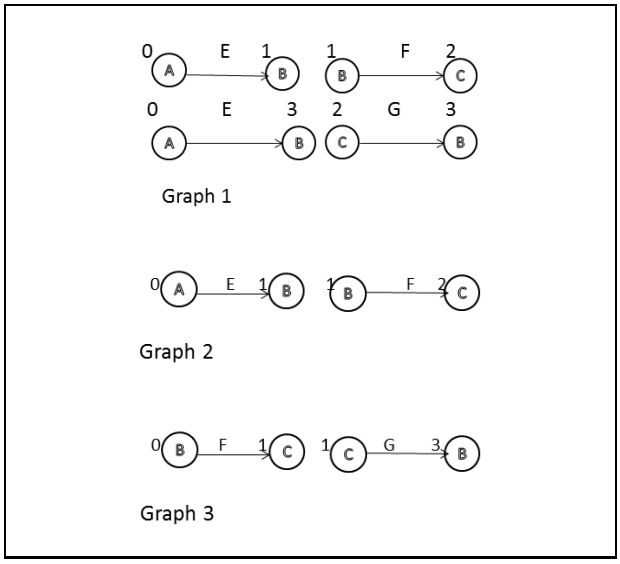


Figure 9: Objects of SingleEdge After Pruning

higher than relational database. Also, scalability of db4o is higher than relational database.

For the comparison of both the methods, we performed the experiments on datasets containing 50,000 to 400,000 graphs. These are transaction graphs. Each graph contains 30-50 edges and 30-50 vertices. Tests were conducted with varying minimum support values 1% , 3%, 5% and 7%. The maximum size of the sub-structures is taken as five. Datasets have millions of records. For example, 50K dataset has approximately 2 million size-1(single) edges. We observed that DB-FSG and OO-FSG performs the same for 50K data (min_*sup*: 1%). Hence, we ignored to evaluate further comparisons with other minimum support values. But as the dataset size grows big, the performance improved dramatically. For datasets of size 100K, the improvement is around 84% and for 400K it is around 230% for minimum support 1%.

Table 6: DB-FSG vs OO-FSG Performance

Dataset size	Min_*sup*	DB-FSG	OO-FSG
50K	1%	357	353
100K	1%	1349	731
100K	3%	1220	656
100K	5%	1061	563
100K	7%	827	484
200K	1%	2439	1331
200K	3%	2002	1206
200K	5%	1717	1117
200K	7%	1622	1030
300K	1%	5887	2221
300K	3%	5394	2141
300K	5%	5137	2019
300K	7%	4164	1863
400K	1%	9502	2879
400K	3%	8228	2457
400K	5%	7156	2426
400K	7%	6962	2313

Note: All run-times in the DB-FSG and OO-FSG columns are shown in seconds.

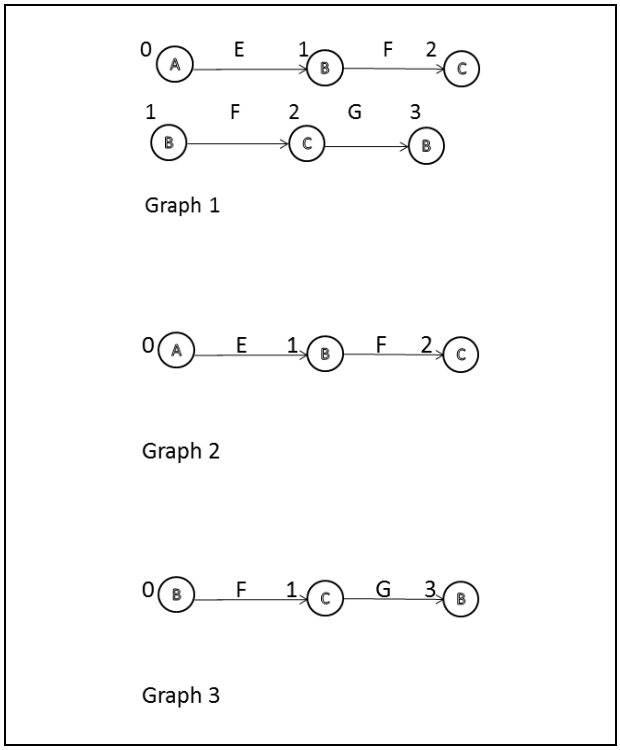


Figure 10: Objects of TwoEdge After Pruning

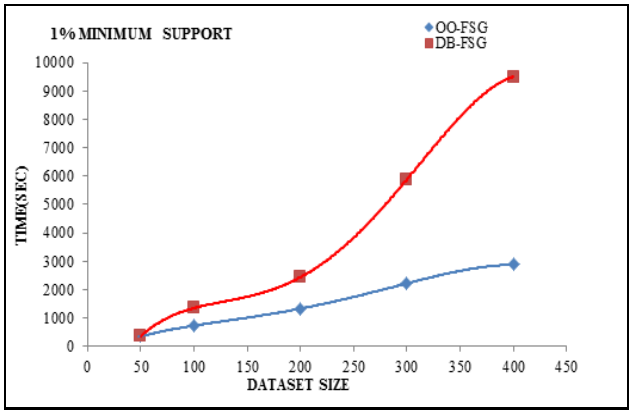


Figure 11: Comparison with 1% minimum support

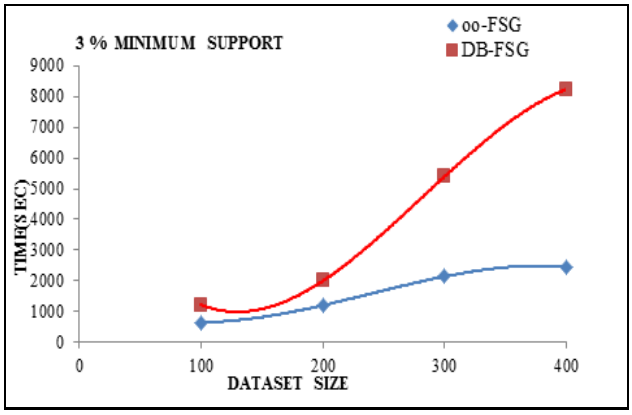


Figure 12: Comparison with 3% minimum support

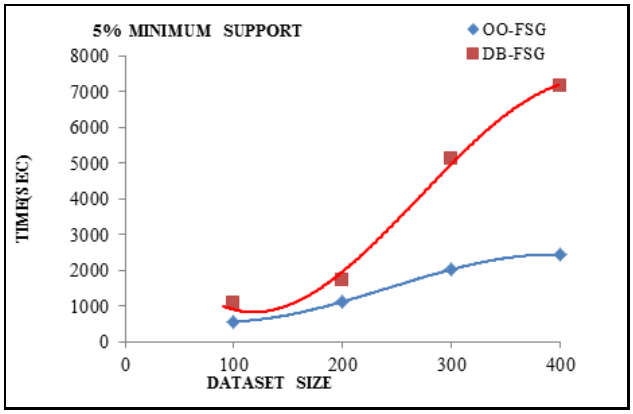


Figure 13: Comparison with 5% minimum support

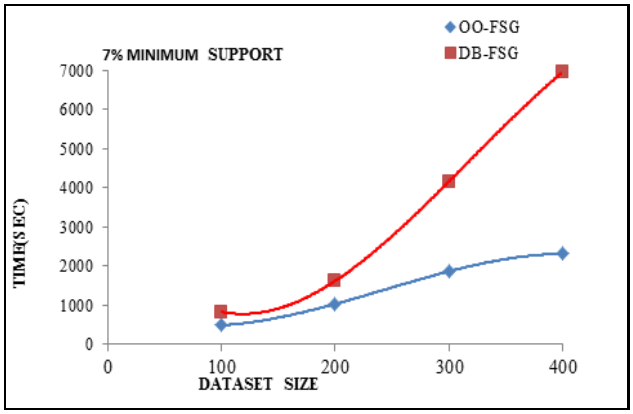


Figure 14: Comparison with 7% minimum support

6 Conclusion and Future Work

In this paper we proposed a new approach for mining frequent subgraphs by storing the graph data sets in an object-oriented database, db4o. Using db4o, large graph data sets can be stored, thus eliminating the constraint of memory resident graph data sets. Also, db4o overcomes the space constraints of relational databases. Furthermore, retaining the semantic information is an added advantage of db4o. To the best of our knowledge, our method is the fastest amongst the entire frequent subgraph mining methods so far. Currently, we are implementing our algorithm on both directed and undirected graph data sets. Our work is in progress to resolve the graph isomorphism problem and storing the ontology in undirected graphs (which this paper does not support currently) in an efficient manner.

7 Acknowledgments

This research has been supported by The Molecular Basis of Disease Program, Georgia State University, USA.

References

Cook, D.J., Holder, L.B. & (2000), Graph-based data mining. *in* 'IEEE Intelligent Systems', 15(2), 32-41.

Inokuchi, A., Washio, T. & Motoda, H. (2003), Complete mining of frequent patterns from graphs. *in* 'Mining graph data. Mach. Learn.', 50(3).

Kuramochi, M., Karypis, G. & (2001), Frequent subgraph discovery. *in* ' ICDM 2001: Proc. of the 2001IEEE International Conference on Data Mining, Washington, DC, USA.', IEEE Computer Society, Los Alamitos pp. 313-320.

Han, J., Yan, X. & (2002), gSpan: Graph-based substructure pattern mining. *in* 'ICDM 2002:Proc. of the 2002 IEEE Int. Conf. on Data Mining', pp. 721-731.

Kuramochi, M., Karypis, G. & (2005), Finding frequent patterns in a large sparse graph. *in* 'Data Min. Knowl. Discov.' 11 (3) (2005) 243-271.

Jiang, X., Xiong, H. & Wang, C., Tan, A. (2009), Mining Globally Distributed Frequent Subgraphs in A Single Labeled Graph. *in Data and Knowledge Discovery*, 68(10), pp: 1034-1058 2209.

Chakravarthy, S., Beera, R. & Balachandran, R. (2004), Database approach to graph mining. *in* 'PAKDD Proceedings', Sydney, pp. 341-350.

Chakravarthy, S., Pradhan, S. & (2008), DB-FSG: An SQL-Based Approach for Frequent Subgraph Mining. *in* 'DEXA 2008', LNCS 5181, pp. 684-692.

Agrawal, R., Srikant, R. & (1994),Fast algorithms for mining association rules. *in* 'J. B. Bocca, M. Jarke, and C. Zaniolo, editors, Proc. of the 20th Int. Conf. on Very Large Databases (VLDB)', pages 487-499. Morgan Kaufmann.

Agarwal, R. C., Aggarwal, C. C. & Prasad, V. V. V., Crestana, V. (1998), A tree projection algorithm for generation of large itemsets for association rules. *in* 'IBM Research Report RC21341'.

Zaki, M. J., Gouda, K. & (2001),Fast vertical mining using diffsets. *in* 'Technical Report 01-1, Department of Computer Science, Rensselaer Polytechnic Institute'.

Jiang, X., Xiong, H. & Wang, C., Tan, A. (2000), Mining frequent patterns without candidate generation. *in* 'Proc. of ACM SIGMOD Int. Conf. on Management of Data', 68(10), Dallas, TX.

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 102 - Computer Science 2010
Edited by Bernard Mans, Macquarie University, Australia and Mark Reynolds, University of Western Australia, Australia. January, 2010. 978-1-920682-83-5.
Contains the proceedings of the Thirty-Third Australasian Computer Science Conference (ACSC 2010), Brisbane, Queensland, Australia, January 2010.

Volume 103 - Computing Education 2010
Edited by Tony Clear, Auckland University of Technology, New Zealand and John Hamer, University of Auckland, New Zealand. January, 2010. 978-1-920682-84-2.
Contains the proceedings of the Twelfth Australasian Computing Education Conference (ACE 2010), Brisbane, Queensland, Australia, January 2010.

Volume 104 - Database Technologies 2010
Edited by Heng Tao Shen, University of Queensland, Australia and Athman Bouguettaya, CSIRO ICT Centre, Australia. January, 2010. 978-1-920682-85-9.
Contains the proceedings of the Twenty-First Australasian Database Conference (ADC 2010), Brisbane, Queensland, Australia, January 2010.

Volume 105 - Information Security 2010
Edited by Colin Boyd, Queensland University of Technology, Australia and Willy Susilo, University of Wollongong, Australia. January, 2010. 978-1-920682-86-6.
Contains the proceedings of the Eight Australasian Information Security Conference (AISC 2010), Brisbane, Queensland, Australia, January 2010.

Volume 106 - User Interfaces 2010
Edited by Christof Lutteroth, University of Auckland, New Zealand and Paul Calder, Flinders University, Australia. January, 2010. 978-1-920682-87-3.
Contains the proceedings of the Eleventh Australasian User Interface Conference (AUIC2010), Brisbane, Queensland, Australia, January 2010.

Volume 107 - Parallel and Distributed Computing 2010
Edited by Jinjun Chen, Swinburne University of Technology, Australia and Rajiv Ranjan, University of New South Wales, Australia. January, 2010. 978-1-920682-88-0.
Contains the proceedings of the Eigth Australasian Symposium on Parallel and Distributed Computing (AusPDC 2010), Brisbane, Queensland, Australia, January 2010.

Volume 108 - Health Informatics and Knowledge Management 2010
Edited by Anthony Maeder, University of Western Sydney, Australia and David Hansen, CSIRO Australian e-Health Research Centre, Australia. January, 2010. 978-1-920682-89-7.
Contains the proceedings of the Fourth Australasian Workshop on Health Informatics and Knowledge Management (HIKM 2010), Brisbane, Queensland, Australia, January 2010.

Volume 109 - Theory of Computing 2010
Edited by Taso Viglas, University of Sydney, Australia and Alex Potanin, Victoria University of Wellington, New Zealand. January, 2010. 978-1-920682-90-3.
Contains the proceedings of the Sixteenth Computing: The Australasian Theory Symposium (CATS 2010), Brisbane, Queensland, Australia, January 2010.

Volume 110 - Conceptual Modelling 2010
Edited by Sebastian Link, Victoria University of Wellington, New Zealand and Aditya Ghose, University of Wollongong, Australia. January, 2010. 978-1-920682-92-7.
Contains the proceedings of the Seventh Asia-Pacific Conference on Conceptual Modelling (APCCM2010), Brisbane, Queensland, Australia, January 2010.

Volume 112 - Advances in Ontologies 2009
Edited by Thomas Meyer, Meraka Institute, South Africa and Kerry Taylor, CSIRO ICT Centre, Australia. December, 2009. 978-1-920682-91-0.
Contains the proceedings of the Australasian Ontology Workshop 2009 (AOW 2009), Melbourne, Australia, December, 2009.

Volume 113 - Computer Science 2011
Edited by Mark Reynolds, The University of Western Australia, Australia. January 2011. 978-1-920682-93-4.
Contains the proceedings of the Thirty-Fourth Australasian Computer Science Conference (ACSC 2011), Perth, Australia, 1720 January 2011.

Volume 114 - Computing Education 2011
Edited by John Hamer, University of Auckland, New Zealand and Michael de Raadt, University of Southern Queensland, Australia. January 2011. 978-1-920682-94-1.
Contains the proceedings of the Thirteenth Australasian Computing Education Conference (ACE 2011), Perth, Australia, 17-20 January 2011.

Volume 115 - Database Technologies 2011
Edited by Heng Tao Shen, The University of Queensland, Australia and Yanchun Zhang, Victoria University, Australia. January 2011. 978-1-920682-95-8.
Contains the proceedings of the Twenty-Second Australasian Database Conference (ADC 2011), Perth, Australia, 17-20 January 2011.

Volume 116 - Information Security 2011
Edited by Colin Boyd, Queensland University of Technology, Australia and Josef Pieprzyk, Macquarie University, Australia. January 2011. 978-1-920682-96-5.
Contains the proceedings of the Ninth Australasian Information Security Conference (AISC 2011), Perth, Australia, 17-20 January 2011.

Volume 117 - User Interfaces 2011
Edited by Christof Lutteroth, University of Auckland, New Zealand and Haifeng Shen, Flinders University, Australia. January 2011. 978-1-920682-97-2.
Contains the proceedings of the Twelfth Australasian User Interface Conference (AUIC2011), Perth, Australia, 17-20 January 2011.

Volume 118 - Parallel and Distributed Computing 2011
Edited by Jinjun Chen, Swinburne University of Technology, Australia and Rajiv Ranjan, University of New South Wales, Australia. January 2011. 978-1-920682-98-9.
Contains the proceedings of the Ninth Australasian Symposium on Parallel and Distributed Computing (AusPDC 2011), Perth, Australia, 17-20 January 2011.

Volume 119 - Theory of Computing 2011
Edited by Alex Potanin, Victoria University of Wellington, New Zealand and Taso Viglas, University of Sydney, Australia. January 2011. 978-1-920682-99-6.
Contains the proceedings of the Seventeenth Computing: The Australasian Theory Symposium (CATS 2011), Perth, Australia, 17-20 January 2011.

Volume 120 - Health Informatics and Knowledge Management 2011
Edited by Kerryn Butler-Henderson, Curtin University, Australia and Tony Sahama, Qeensland University of Technology, Australia. January 2011. 978-1-921770-00-5.
Contains the proceedings of the Fifth Australasian Workshop on Health Informatics and Knowledge Management (HIKM 2011), Perth, Australia, 17-20 January 2011.

Volume 122 - Advances in Ontologies 2010
Edited by Thomas Meyer, UKZN/CSIR Meraka Centre for Artificial Intelligence Research, South Africa, Mehmet Orgun, Macquarie University, Australia and Kerry Taylor, CSIRO ICT Centre, Australia. December 2010. 978-1-921770-00-5.
Contains the proceedings of the Sixth Australasian Ontology Workshop 2010 (AOW 2010), Adelaide, Australia, 7th December 2010.