

CONFERENCES IN RESEARCH AND PRACTICE IN
INFORMATION TECHNOLOGY

VOLUME 134

DATA MINING AND ANALYTICS 2012
(AusDM'12)



DATA MINING AND ANALYTICS 2012

Proceedings of the
Tenth Australasian Data Mining Conference
(AusDM'12), Sydney, Australia,
5–7 December 2012

Yanchang Zhao, Jiuyong Li , Paul J. Kennedy and
Peter Christen, Eds.

Volume 134 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 2012. Proceedings of the Tenth Australasian Data Mining Conference (AusDM'12), Sydney, Australia, 5–7 December 2012

Conferences in Research and Practice in Information Technology, Volume 134.

Copyright ©2012, Australian Computer Society. Reproduction for academic, not-for-profit purposes permitted provided the copyright text at the foot of the first page of each paper is included.

Editors:

Yanchang Zhao

Department of Immigration and Citizenship, Australia;
and RDataMining.com
5 Chan St
Belconnen, ACT 2617, Australia
Email: yanchangzhao@gmail.com

Jiuyong Li

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

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

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 134.
ISSN 1445-1336.
ISBN 978-1-921770-14-2.

Document engineering by CRPIT, December 2012.

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

Table of Contents

Proceedings of the Tenth Australasian Data Mining Conference (AusDM'12), Sydney, Australia, 5–7 December 2012

Preface	vii
Conference Organisation	viii
AusDM Sponsors	x

Keynotes

Data = Normal + Anomalous + Noise.....	3
<i>Sanjay Chawla</i>	
Non-iidness: Coupled Object and Pattern Analysis	5
<i>Longbing Cao</i>	

Contributed Papers

Detecting Topic Labels for Tweets by Matching Features from Pseudo-Relevance Feedback	9
<i>Jing Zhang, Derek Liu, Kok-Leong Ong, Zhijie Li and Ming Li</i>	
Unsupervised Text Segmentation using LDA and MCMC	21
<i>Kaimin Yu, Zhe Li, Genliang Guan, Zhiyong Wang and David Feng</i>	
CRUDAW: A Novel Fuzzy Technique for Clustering Records Following User Defined Attribute Weights	27
<i>Md Anisur Rahman and Md Zahidul Islam</i>	
GOtoGene: A Method for Determining the Functional Similarity among Gene Products	43
<i>Kamal Taha</i>	
Functional Visualisation of Genes using Singular Value Decomposition	53
<i>Hamid Ghous, Paul J. Kennedy, Nicholas Ho and Daniel Catchpoole</i>	
A Non-Time Series Approach to Vehicle Related Time Series Problems	61
<i>Jonathan Wells, Kai Ming Ting and Chandrasiri Naiwala</i>	
Variance-wise Segmentation for a Temporal-Adaptive SAX	71
<i>Chao Sun, David Stirling, Christian Ritz and Claude Sammut</i>	
Coding of Non-Stationary Sources as a Foundation for Detecting Change Points and Outliers in Binary Time-Series	79
<i>Peter Sunehag, Wen Shao and Marcus Hutter</i>	
ABC-SG: A New Artificial Bee Colony Algorithm-Based Distance of Sequential Data Using Sigma Grams	85
<i>Muhammad Marwan Muhammad Fuad</i>	
Improving Classifications for Cardiac Autonomic Neuropathy Using Multi-level Ensemble Classifiers and Feature Selection Based on Random Forest	93
<i>Andrei Kelarev, Andrew Stranieri, John Yearwood, Jemal Abawajy and Herbert Jelinek</i>	

Combining Classifiers in Multimodal Affect Detection	103
<i>Md. Sazzad Hussain, Hamed Monkaresi and Rafael Calvo</i>	
Application of Tree-structured Data Mining for Analysis of Process Logs in XML format	109
<i>Dang Bach Bui, Fedja Hadzic and Michael Hecker</i>	
Associative Classification using a Bio-Inspired Algorithm	119
<i>Omar S. Soliman, Roba Bahgat and Amr Adly</i>	
An Iterative Two-Party Protocol for Scalable Privacy-Preserving Record Linkage	127
<i>Dinusha Vatsalan and Peter Christen</i>	
VICUS - A Noise Addition Technique for Categorical Data	139
<i>Helen Giggins and Ljiljana Brankovic</i>	
Cartesian Genetic Programming for Trading: A Preliminary Investigation	149
<i>Michael Mayo</i>	
A Comparative Study of MRI Data using Various Machine Learning and Pattern Recognition Algorithms to Detect Brain Abnormalities	157
<i>Lavneet Singh and Girija Chetty</i>	
Indirect Weighted Association Rules Mining for Academic Network Collaboration Recommendations	167
<i>Yun Sing Koh and Gillian Dobbie</i>	
Using network evolution theory and singular value decomposition method to improve accuracy of link prediction in social networks	175
<i>Qinxue Meng and Paul J. Kennedy</i>	
Learning Personalized Tag Ontology from User Tagging Information	183
<i>Endang Djuana, Yue Xu and Yuefeng Li</i>	
A Collaborative Filtering Recommendation System Combining Semantics and Bayesian Reasoning ..	191
<i>Jialing Li, Li Li, Xiao Wen and Jianwei Liao</i>	
Evaluating the Performance of Several Data Mining Methods for Predicting Irrigation Water Requirement	199
<i>Mahmood Khan, Md. Zahidul Islam and Mohsin Hafeez</i>	
Anytime Algorithms for Mining Groups with Maximum Coverage	209
<i>Satya Gautam Vadlamudi, Partha Pratim Chakrabarti and Sudeshna Sarkar</i>	
Mining cluster-based patterns for elder self-care behaviour	221
<i>Yu-Shiang Hung, Kuei-Ling B. Chen, Chi-Ta Yang and Guang-Feng Deng</i>	
Data Guided Approach to Generate Multi-dimensional Schema for Targeted Knowledge Discovery ...	229
<i>Muhammad Usman, Russel Pears and A.C.M. Fong</i>	
Author Index	241

Preface

We are delighted to welcome you to the Tenth Australasian Data Mining Conference (AusDM'12) being held this year in Sydney. AusDM started in 2002 and is now the annual flagship meeting for data mining and analytics professionals in Australia. Both scholars and practitioners present the state-of-the-art in the field. Endorsed by the peak professional body, the Institute of Analytics Professionals of Australia, AusDM has developed a unique profile in nurturing this joint community. The conference series has grown in size each year from early workshops held in Canberra (2002, 2003), Cairns (2004), Sydney (2005, 2006), the Gold Coast (2007), Glenelg (2008), Melbourne (2009) and Ballarat (2011). This year's event has been supported by

- Togaware, again hosting the website;
- University of Technology Sydney and the Centre for Quantum Computation and Intelligent Systems for their support with student volunteers;
- University of Western Sydney for support with registrations;
- the Australian Computer Society, for publishing the conference proceedings.

The conference program committee reviewed 55 submissions, out of which 25 submissions were selected for publication and presentation. This was an acceptance rate of 45%. AusDM follows a rigid blind peer-review process and ranking-based paper selection process. All papers were extensively reviewed by at least two referees drawn from the program committee, and the majority of papers were reviewed by three referees. 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, two keynote speakers were invited. Professor Sanjay Chawla from the University of Sydney talked about data being made of normal, anomalous and noisy components, and presented a modern and algorithmic viewpoint of anomaly detection. Professor Longbing Cao from the University of Technology Sydney explored the needs, challenges, opportunities of analysing complex object relations and complex pattern relations where objects are either loosely or tightly coupled with each other.

AusDM'12 was co-located with the 25th Australian Joint Conference on Artificial Intelligence (AI'12), and participants were able to enjoy the keynotes and presentations offered by AI'12.

We would like to thank the organisers of AI'12 for their support in co-locating these two conferences. We would also like to extend our special thanks to the program committee members. The final quality of the 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.

Yanchang Zhao

Department of Immigration and Citizenship, Australia;
and RDataMining.com

Jiuyong Li

University of South Australia
AusDM 2012 Programme Chairs

Peter Christen

The Australian National University

Paul J. Kennedy

University of Technology, Sydney
AusDM 2012 Conference Chairs
December 2012

Conference Organisation

Program Chairs

Yanchang Zhao, Department of Immigration and Citizenship, Australia; and RDataMining.com
Jiuyong Li, University of South Australia

Conference Chairs

Paul Kennedy, University of Technology, Sydney
Peter Christen, Australian National University

Steering Committee Chairs

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

Other Steering Committee Members

Peter Christen, Australian National University
Paul Kennedy, University of Technology, Sydney
Jiuyong Li, University of South Australia
Kok-Leong Ong, Deakin University
John Roddick, Flinders University
Andrew Stranieri, University of Ballarat
Geoff Webb (advisor), Monash University

Program Committee

Adil Bagirov, University of Ballarat, Australia
Rohan Baxter, Australian Taxation Office, Australia
Xuan-Hong Dang, Aarhus University, Denmark
Richard Dazeley, University of Ballarat, Australia
Zari Dzalilov, University of Ballarat, Australia
Vladimir Estivill-Castro, Griffith University, Australia
Ross Gayler, Veda Advantage, Australia
Raj Gopalan, Curtin University of Technology, Australia
Warwick Graco, Australian Taxation Office, Australia
Lifang Gu, Australian Tax Office, Australia
Ping Guo, Beijing Normal University, China
Robert Hilderman, University of Regina, Canada
Yun Sing Koh, University of Auckland, New Zealand
Siddhivinayak Kulkarni, University of Ballarat, Australia
Paul Kwan, University of New England, Australia
Gang Li, Deakin University, Australia
Huizhi Liang, The Australian National University, Australia
Bin Linghu, The University of Science and Technology of China
Jixue Liu, University of South Australia, Australia
Lin Liu, University of South Australia, Australia
Bradley Malin, Vanderbilt University, USA
Musa Mammadov, University of Ballarat, Australia
Arturas Mazeika, Max Planck Institute of Informatics, Germany
Christine O’Keefe, CISRO, Canberra, Australia

Tom Osborn, University of Technology, Sydney, Australia
Robert Pearson, Health Insurance Commission, Australia
Francois Poulet, IRISA, France
Sunam Pradhan, University of Ballarat, Australia
Yin Shan, Department of Human Services, Australia
Zhongzhi Shi, Chinese Academy of Sciences, China
David Taniar, Monash University, Australia
Xiaohui Tao, University of Southern Queensland, Australia
Peter Vamplew, University of Ballarat, Australia
Sitalakshmi Venkatraman, University of Ballarat, Australia
Andrew Wyer, Department of Immigration and Citizenship, Australia
John Yearwood, University of Ballarat, Australia
Ting Yu, University of Sydney, Australia
Ji Zhang, The University of South Queensland, Australia

AusDM Sponsors

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



<http://www.togaware.com>



<http://www.uts.edu.au>



Centre for Quantum Computation and Intelligent Systems

<http://www.qcis.uts.edu.au/>



Bringing knowledge to life

<http://www.uws.edu.au/>

KEYNOTES

Data = Normal + Anomalous + Noise

Sanjay Chawla

School of Information Technologies,
The University of Sydney,
Sydney NSW 2006, Australia Email: sanjay.chawla@sydney.edu.au

Abstract

Our world at the micro, macro and personal level is now highly instrumented. A consequence of this instrumentation is that now it is possible to obtain fine-grained data about almost anything of interest. Once we focus on an application or a domain, it is reasonable to assume that much of the data obtained captures the "normal" behavior of the underlying phenomenon. Historically, "knowledge discovery," if any, has been triggered by the non-normal or anomalous part of the data. In this talk I will present some classic examples of data anomalies and how their discovery has changed our understanding of the world. Then I will present a modern and algorithmic viewpoint of anomaly detection as is currently practiced in the data mining community.

Non-iidness: Coupled Object and Pattern Analysis

Longbing Cao

Advanced Analytics Institute,
University of Technology Sydney
Broadway NSW 2007, Australia
Email: longbing.cao@uts.edu.au

Abstract

Most of existing data mining algorithms are based on the IID assumption, which treats objects independently from each other. In the real world, objects are either loosely or tightly coupled with each other. For instance, a moving vehicle on the street interacts with the cars before and after it, and the ones on its left and right hand sides if any. In social networks, people interact with each other at different levels for varied purposes. Such interactions, or coupling relationships, are ubiquitous, and spread at various levels, between objects, between attributes describing an object, between attribute values within an attribute. It is crucial to cater for such relations in object analysis.

On the other hand, the usual patterns identified by data mining are based on independent objects or items. For instance, often a large number of frequent patterns are mined by the existing algorithms, which are often treated as independent with each other. In fact, due to the object coupling relationships, patterns are associated with each other in structural and/or semantic aspects. Pattern relationship analysis is often ignored.

In this talk, we will explore the needs, challenges, opportunities of analyzing complex object relations and complex pattern relations. On top of a framework for noniid-based coupled object and pattern analysis, several corresponding techniques will be introduced: coupled object analysis to define and quantify the coupling relationships within and between objects and within and between attributes, combined pattern mining to identify a group of patterns coupled by certain relationships. Coupled behavior analysis will be explored to analyse a group of actors behaviors. We will show how such new frameworks outperform the classic iid-based data mining framework in terms of handling complex data, behavior, relation, environment and pattern in clustering, frequent pattern mining, and classification. Several real-life applications will be given, such as the identification of group-based market manipulations in stock markets.

CONTRIBUTED PAPERS

Detecting Topic Labels for Tweets by Matching Features from Pseudo-Relevance Feedback

Jing Zhang Derek Liu Kok-Leong Ong Zhijie Li Ming Li

School of Information Technology, Deakin University
221 Burwood Highway, Burwood, Victoria 3125

Email: {jing.zhang, derek.liu, kok-leong.ong, z.li, ming.li}@deakin.edu.au

Abstract

Detecting a suitable topic label for short texts, e.g., tweets from Twitter, is an important component in many applications including diversity ranking, clustering, information retrieval, and information filtering. To automatically detect topic labels however is a major challenge. The character limit of a short text means the lack of a significant feature space to adequately describe its content in relation to other short texts in a given collection. Therefore, methods like LDA, TF-IDF or similarity measures all fail due to their sensitivity to a small feature space. And when a collection of related short texts are considered, e.g., from a Twitter search, the result set collectively exhibits sparsity *and* high dimensionality – a nightmare for information processing. A solution to this problem is to expand the feature space through a process known as pseudo-relevance feedback. Unfortunately, they disappoint when subjected to real-world conditions. The fundamental problem lie in the level of noise present in both the short texts and the feedback source, which is often the World Wide Web. We propose a novel pseudo-relevance feedback algorithm to accurately identify topic labels for short texts. Our algorithm robustly handles noise in both the short texts and the feedback source through a method called ‘feature matching’. Empirical results confirm the efficacy of our algorithm.

Keywords: Tweets, Twitter, Pseudo-Relevance Feedback, Short Texts, Topic Detection

1 Introduction

The modern Web is no longer just a repository for Web documents. It is now a hybrid of different media and different Web applications. Most recently, a huge amount of user generated content arising from social networking Websites are fuelling a new category of data. They are large in volume but each is terse in its content. We call them short texts. Short texts are increasingly becoming prevalent on the Web. They exist as summaries to a Website in search results, as tweets on Twitter, as status updates on FaceBook, or as comments on YouTube.

The volume of short texts has motivated many applications requiring the use of algorithms in areas such as diversity ranking, clustering, classification, infor-

mation retrieval, and information filtering. These algorithms in turn depend on core components, one of which is to know the topic label of a short text. For example, some diversity ranking algorithms achieve diversity by ensuring different topics of short texts are included. Another example would be in classification, where a rank of topic labels is used to classify short texts into pre-determined categories.

Topic detection in short texts however is a challenging problem. Using the case of tweets for example, the 140 character limit means that there is hardly sufficient features present to adequately describe its content in relation to other tweets in a given collection. When the feature space is very small and the collection in question creates a collective feature space that is very sparse and high in dimension, most techniques like LDA (Blei, Ng & Jordan 2003), TF-IDF (Manning, Raghavan & Schütze 2008), or feature-based similarity measures would all fail under real-world conditions. This has been well-reported in many other literature such as (Bernstein *et al.* 2010) and (Zhang *et al.* 2011).

A way to overcome the limitation of small feature spaces and to deal with a collection that is sparse and highly dimensioned is to expand (or enrich) the original feature space by adding related features from another source. This technique is known as pseudo-relevance feedback, or simply relevance feedback (Lloret 2009). The feedback source, which is where additional related features are found, can be

- a collection of other short texts that has been manually processed;
- a collection of well-structured documents in the same domain as the short texts;
- a public domain collection such as Wikipedia or WordNet;
- or the largest public domain resource, i.e., World Wide Web.

If we consider short texts such as those drawn from Twitter, then the first two feedback sources will not be practically feasible because (i) of the effort required to build the short texts collection or the well-structured documents; and also (ii) the feedback source is likely to become outdated quickly when we consider how fast tweet topics may change. The third feedback source, although more robust towards changes, can be limited in the scope of topics it can cover. The last feedback source, the World Wide Web, is the largest public domain resource and is likely to evolve as rapidly as the topics developing on Twitter. So theoretically, the Web is the ideal candidate.

In exploring our solution, we came across feedback systems that uses the Web as its feedback source. The most recent is the work reported in (Bernstein *et al.* 2010), which is also very close to the problem we are trying to solve. We recreated this system based on the description given and discovered that when the Web is used as the feedback source, the results can disappoint when real-world tweets are used. The problem lie in the noise level of the feedback source, which we will discuss in detail next. Nevertheless, the poor results motivated us to search for a solution that would perform well in real-world situations.

Our quest, based on an understanding of the issues surrounding the Web as a feedback resource, saw the development of a *feature matching* algorithm that would produce an accurate way to determine the topic label of a tweet. The prototype of our implementation is now live for public testing and the evaluation of user results has confirmed its ability to deliver a high level of accuracy based on users of Twitter.

We shall now introduce our *feature matching* algorithm in Section 3 but before that, we discuss in Section 2 why current Web-based feedback systems fail to produce adequate results. We then present our experimental results in Section 4, where we compare the topic labels detected from our algorithm against other Web-based feedback systems. We then end this paper by pointing readers to related works in Section 5 and drawing our conclusions in Section 6.

2 Relevance Feedback in the Real-World

To understand why the state of the art in Web-based pseudo-relevance feedback fail, we discuss an implementation call Eddi (Bernstein *et al.* 2010). Eddi was designed as a tool to organise tweets by their topics. To do so, a relevance feedback process was used to compute a topic label for the tweet. Tweets with similar topic labels are then grouped together.

Eddi's algorithm consists of three main steps: (i) text transformation, (ii) search engine query, and (iii) text feature extraction. The first step aims to transform a tweet into a search query. This involves basic pre-processing such as removing 'RT' (re-tweets), '@username' mentions, URL references, etc. The second step involves taking the transformed tweet and converting it into a search query. In (Bernstein *et al.* 2010), this is done by identifying the noun phrases as it was found that nouns are good topic markers in long documents (Bendersky and Croft 2008, Hulth 2003). To find the nouns, a Part-of-Speech (POS) tagging software is used (Kristina and Christopher 2000). The nouns identified are then used to query a search engine - in their case, Yahoo!. The top ten Web documents associated with the nouns are then retrieved. Each Web document is then computed for its TF-IDF (i.e., *term frequency-inverse document frequency*) and the top TF-IDF (Manning, Raghavan & Schütze 2008) terms are then merged through a voting system, where terms more common among the ten documents are selected over terms with fewer votes. These terms are the topic label(s) associated with the tweet.

Let's look at a tweet that was handled well by Eddi: "*awesome article on some SIGGRAPH user interface work: <http://bit.ly/30MJy>*". As per the algorithmic steps, the transformed output presents us with the search phrase (consisting of noun terms): "article SIGGRAPH user interface work". The first ten Web documents obtained from the search phrase are then downloaded and the TF-IDF of each term across the documents computed. The top TF-IDF

terms obtained in this specific case were *animation, character, 3D, computer, graphics, user, interface* and *SIGGRAPH*. These terms were clearly good candidates as topic labels for the original short text (tweet).

To see why this specific case works, we look at the results from the search engine (our feedback source) as shown in Figure 2. For the SIGGRAPH example, i.e., Figure 2(a), the documents returned are close to plain text which makes them easy to process. Compared to the Web documents we obtained from the next example shown in Figure 2(b), there is a sharp contrast in the level of 'noise' between the two sets of Web documents. For the SIGGRAPH tweet, the query returns the following URLs.

- http://www.interaction-design.org/references/conferences/proceedings_of_the_1st_annual_acm_siggraph_symposium_on_user_interface_software.html
- http://www.interaction-design.org/references/conferences/proceedings_of_the_3rd_annual_acm_siggraph_symposium_on_user_interface_software_and_technology.html
- [http://en.wikipedia.org/wiki/WIMP_\(computing\)](http://en.wikipedia.org/wiki/WIMP_(computing))
- <http://www.siggraph.org/publications/newsletter/v32n3/columns/elvins.html>
- <http://kyungku.net/xs/publication/6442>
- <http://www.ee.columbia.edu/~sfchang/course/svia-F03/papers/siggraph-reject-how.htm>
- <http://mi-lab.org/about/people/michael-haller/>
- http://web.cs.wpi.edu/~matt/courses/cs563/talks/smartin/int_design.html
- <http://plecebo.org/content/fun-ui-innovations-siggraph-09-conference>
- <http://userwww.sfsu.edu/~jkveeder/bio/500.htm>

Now compare this to a tweet about Qantas, Figure 2(b): "*Sale #airfare #fly #Canberra to #Wellington from \$410 with Qantas - <http://t.co/2jsXBRbv>*". which after the POS tagging, we had the search phrase "sale airfare canberra wellington qantas". The ten Web documents we obtained for this case contain JavaScripts, Flash content, advertisements, CSS styling, animated menus, dynamic presentation structures, dynamic forms, and server-side generated content. With so many layers of 'noise', any attempt to get to the actual content relevant to the search query becomes very challenging. We also went further by developing variations of Eddi such as (i) taking advantage of any short URLs present in the tweet to compute the TF-IDF; (ii) using a constrained set of Web documents (BlogSpot) to limit the level of noise; and (iii) using algorithms such as NReadability to extract the content. Unfortunately, the results we obtained from our experiments on all the variations were unsatisfactory. We conclude that when presented with such noisy documents, Eddi fails to provide accurate results. And with most of the Web documents today looking more like those seen in our Qantas example, the ability for Eddi to extend to real-world usage is actually questioned.

3 Feature Matching as Proxy Measure

Having failed from attempts to improve Eddi through various 'de-noising' strategies, we conclude that we have to accept the presence of noise in a feedback source like the Web. We also conclude that it would be difficult to overcome noise. This led us to a different strategy, where we embrace the noise present in Web documents instead. The idea in Eddi is to compute the TF-IDF from Web documents so as to

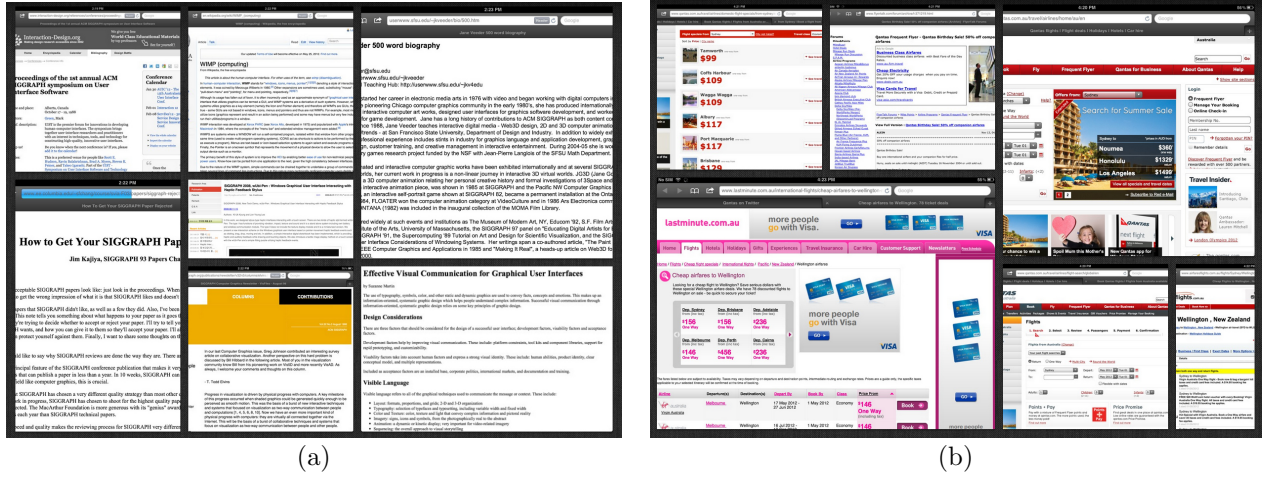


Figure 1: (a) A montage of the various screens for the search phrase “article SIGGRAPH user interface work”. Notice that the Web documents for this particular instance is not “noisy” and many of them are simple text-oriented documents without formatting, layers, advertisements, etc. Consequently, this makes extraction of the actual body of content easy and lowers the error probability significantly to allow the TF-IDF compute to show meaningful topic labels. (b) A montage of the various screens for the search phrase “sale airfare canberra wellington qantas”. Compared to (a), the Web documents here are a lot more complex in their presentation as they incorporate dynamic content such as Flash and JavaScript, CSS styling and interactive menu, advertisements, photos and forms, etc. Extracting the main content from these Web documents so as to compute the TF-IDF of its word terms is not only challenging but it clearly showcases where relevance feedback systems would fail to provide accurate results.

derive the topic labels. As a result, it is very dependent on what terms are in the document. And given the way TF-IDF works for just ten documents, spurious terms can be highly weighted so noise is actually highlighted as topic labels. Furthermore, the results of a TF-IDF compute are single word terms. Topic labels such as “global warming” would appear as two word terms that require an expert to further piece them together. When we consider these limitations, the want for a different solution becomes clear.

3.1 Problem Formulation

Our solution to make Web-based relevance feedback work comes from a simple observation about the relationship between the Web documents, the topic label and the short text, which are all part of the pseudo-relevance feedback process.

Given a tweet t , a human expert could provide a topic label ℓ based on the word terms in t . At the same time, the same word terms from t could be used by the human expert to select a collection of documents $\mathcal{D}_t = \{d_1, d_2, \dots, d_j\}$, such that these documents also share the topic ℓ . In other words, if all the documents in \mathcal{D}_t are selected for ℓ and that ℓ is some function of t , then ℓ can be seen as a query that returns a set of relevant Web documents \mathcal{D}_t . And the query, which is ℓ , is in fact the topic label of t .

The problem in this case is that ℓ is determined by the human expert. For example, the tweet in Figure 2(b) can be labelled by the human expert as $\ell = \text{‘qantas domestic sales’}$. This would make a good topic label for t and a set of relevant documents to expand the feature space can be easily obtained by searching the Web using the terms from ℓ .

Clearly, the human expert cannot possibly be a component of the relevance feedback system. It would appear that without human expertise to determine ℓ , we won’t have a solution. This turns out to be not the case. For a tweet, we often obtain them from a search, a hash tag, or by following another Twitterer. In such situations, we can easily determine the top-level con-

cept \mathcal{C} in relation to the tweet. For example, the tweets in Figure 2 are obtained by searching for ‘sig-graph’ and ‘qantas’ respectively on Twitter. These query terms are therefore our top-level concepts. As soon as we know \mathcal{C} , we can easily derive a set of ℓ -candidates, i.e., $\mathcal{L}(\mathcal{C}) = \{\ell_1, \ell_2, \dots\}$.

In implementation, one way to easily derive the ℓ -candidates from the top-level concept \mathcal{C} is to use the ‘related searches’ often suggested by a search engine. For example when $\mathcal{C} = \text{‘qantas’}$, the Bing search engine returns {‘frequent flyer’, ‘international’, ‘domestic flights’, ‘staff travel’, ‘holidays’, ‘staff credit union’, ‘flights’, ‘frequent flyer points account’} as related search topics. If we drill deeper into ‘international’, we obtain further suggestions which include {‘arrivals’, ‘air fares’, ‘bookings’, ‘baggage allowance’, etc.}. Clearly, each related search suggestion is a candidate for a topic label. So from \mathcal{C} , we can now derive a good set of ℓ -candidates, i.e., $\mathcal{L}(\mathcal{C})$.

At this point, it becomes clear that each $\ell_i \in \mathcal{L}(\mathcal{C})$ allows us to easily obtain a relevant set of documents \mathcal{D}_{ℓ_i} . So for each $\ell_i \in \mathcal{L}(\mathcal{C})$, we now have a tuple $\langle \ell_i, \mathcal{D}_{\ell_i} = \{d_i, d_j, \dots\} \rangle$ or for $\mathcal{L}(\mathcal{C})$, a set of tuples $\{ \langle \ell_x, \mathcal{D}_{\ell_x} \rangle, \langle \ell_y, \mathcal{D}_{\ell_y} \rangle, \dots \}$. To determine the topic label for t obtained via the same concept \mathcal{C} , we perform the usual relevance feedback to obtain the tuple $\langle t_\ell, \mathcal{D}_t = \{d_p, d_q, \dots\} \rangle$, where t_ℓ is the transformed t as per step (i) of a relevance feedback system. Now the solution to our problem of finding a topic label ℓ for t is transformed into finding a tuple in $\{ \langle \ell_x, \mathcal{D}_{\ell_x} \rangle, \langle \ell_y, \mathcal{D}_{\ell_y} \rangle, \dots \}$ where the features in \mathcal{D}_{ℓ} is closest to the features in \mathcal{D}_t . The ℓ of this tuple is the topic label for t as their associated documents (or enriched feature space) are the most similar.

By matching features found in \mathcal{D}_t and \mathcal{D}_{ℓ} , we are no longer looking for specific word terms. Rather, we are looking for a signature in the set of documents to describe a topic label ℓ . Here, when two sets of documents share a similar signature in their features, we can suggest (or equate) ℓ_t as ℓ . In doing so, the solution of finding ℓ for t is solved.

Algorithm 1 FindTopicLabel(t, \mathcal{C})

```

1: build  $\mathcal{L}(\mathcal{C})$  from  $\mathcal{C}$  using ‘related search’
2: obtain  $\mathcal{T}_t = \langle t_\ell, \mathcal{D}_t \rangle$  by relevance feedback
3: obtain  $\mathcal{T}_{\mathcal{L}(\mathcal{C})} = \{\langle \ell_1, \mathcal{D}_{\ell_1} \rangle, \dots \}$  from search engine
4: for each  $i \in \mathcal{T}_{\mathcal{L}(\mathcal{C})}$  do
5:   // calculate each  $\mathcal{S}$  and store result
6:   // in hash table  $M$ .
7:    $M(i) \leftarrow \mathcal{S}(\mathcal{S}'(\mathcal{T}_t, \mathcal{D}_t, i, \mathcal{D}_{\ell_x}))$ 
8: end for
9: return  $i.\ell : M(i) > M(j) \forall j \neq i$ 

```

We can compute the signature in many ways and we present a simple approach in the next section. The strength of the signature approach is that it is a lot more robust against the presence of ‘noise’ in Web documents. In fact, our approach accepts the presence of noise and incorporates them as part of a topic label’s signature.

3.2 Algorithmic Solution

Recall from our earlier discussion, both \mathcal{D}_ℓ and \mathcal{D}_t are a set of documents, i.e., $\{d_x, d_y, \dots\}$. The straightforward approach is to take these documents as the respective signature. After all, the combination of Web documents in \mathcal{D} form a collective set of features that describes the topic label.

In this straightforward approach, we can compute a signature similarity score \mathcal{S} to show how similar the signatures are. This is done in two steps: (i) compute the basic cosine similarity between two documents, each drawn from \mathcal{D}_t and \mathcal{D}_{ℓ_x} respectively, i.e.,

$$\begin{aligned} \mathcal{S}'(\mathcal{D}_t, \mathcal{D}_{\ell_x}) &= \mathcal{D}_t \times \mathcal{D}_{\ell_x} \\ &= \{\text{Sim}(d_i, d_j) : d_i \in \mathcal{D}_t \wedge d_j \in \mathcal{D}_{\ell_x}\} \end{aligned}$$

and then (ii) obtain the average of the cosine similarity scores in \mathcal{S}' , i.e.,

$$\mathcal{S}(\mathcal{S}') = \frac{1}{|\mathcal{S}'|} \sum_i s \in \mathcal{S}'$$

The highest signature similarity score \mathcal{S} for an ℓ -candidate from $\mathcal{L}(\mathcal{C})$ will be selected as the topic label. The algorithm to tie the discussion of our solution together is shown in Algorithm 1.

While the algorithm uses \mathcal{C} to obtain the ℓ -candidates in Step 1, the solution does not really require it. The presence of \mathcal{C} helps cut the search space, i.e., the number of ℓ -candidates to consider and consequently, improves runtime performance. Step 2 of the algorithm would be the usual relevance feedback, where t is first transformed into t_ℓ (by the usual preprocessing and POS tagging), and a search conducted using t_ℓ to find a set of relevant documents \mathcal{D}_t . In Step 3, the relevant documents for each ℓ -candidate from $\mathcal{L}(\mathcal{C})$ are retrieved. Again, a good implementation would have cached the frequently used ℓ -candidates to minimise Web access for performance reasons.

While we didn’t implement caching in our prototype, we did limit the size of each document download to 300KB. This greatly improved performance without having to cache any ℓ -candidate documents, some of which are up to 10MB in our experiments. Empirically, the 300KB performed well without affecting our accuracy. Given that we are only interested in using the documents to form a signature, truncating the download is actually fine.

Steps 4 to 8 simply computes the signature similarity score for each pair of documents in \mathcal{D}_t and \mathcal{D}_{ℓ_x} storing the result in a hash table M . Once this is completed, Step 9 returns the ℓ with the top \mathcal{S} score but since one has access to M , the algorithm can return the top- n topic labels as well.

4 Empirical Results

An important aspect of our solution is the premise that a search query is (or will contain) an implicit topic label. This topic label is developed in a search query as users seek relevant documents by refining their search with additional keywords. Over time, this large amount of user queries and clickthroughs has allowed the search engine to learn related searches and the best documents matching each specific query. The indirect consequence of this is that we can now use ‘related searches’ as a viable source of topic labels based on the solution we presented. It becomes a very powerful way of cutting the search space. At up to two levels deep of related searches, our experimental results show that the topic labels assigned to a tweet will worked very well.

We validated our results as follows. We first obtained a published list of top Twitter queries¹ and hash tags² used in 2010 and 2011 respectively. We then performed a Twitter search using these query terms and hash tags to obtain a collection of tweets for our experiment. In this paper, we reported the results from the tweets we collected over the period of July 2012. For each tweet, we recorded the top three and the bottom three topic labels as determined by our algorithm. We then presented the results to a group of Twitter users to assess whether they agree with the topic label assigned.

Our Twitter users were students in a third year software development class taught by one of the authors. Each student was given twenty tweets, half of the tweets were picked from search terms and the other half from hash tags. For each tweet, the top three and bottom three topic labels are shown. The students were to give a score between 1 to 5 to indicate whether they think the top topic label is the best among the six shown. A score of 5 indicates that they fully agree with the algorithm’s assessment.

There was a total of twenty students who took part in the assessment. After they made their assessment, we assessed the inter-rater agreement for each tweet across the twenty raters using Fleiss’s Kappa measure (Fleiss 1977). The Kappa measure is a statistical method to determine the reliability of agreement between raters. In our experiment, the score of 1 to 5 is treated as a nominal measure rather than an ordinal one. Over the twenty tweets, the Kappa value we obtained was just over 0.6 but less than 0.61 (0.6036 to be exact). This places us somewhere between “moderate agreement” and “substantial agreement” according to (Landis and Koch 1977).

Our personal and possibly subjective assessment however motivated us to look deeper into the results as we anticipated a score that clearly puts us in the “substantial agreement” category. We note that the wider the range of scores, the weaker the final result. When we reduced the scoring system to just ‘yes’, ‘no’ and ‘possibly’, the same twenty tweets achieved a better score of 0.73 putting it clearly in the “substantial

¹<http://blog.sfgate.com/techchron/2010/12/13/>

gulf-oil-spill-world-cup-top-twitter-trends-for-2010/

²<http://tallskinnykiwi.typepad.com/tallskinnykiwi/2011/12/egypt-the-top-twitter-hashtag-for-2011.html>

agreement” category. We did however have one variable: we had a different group of students to score the same twenty tweets. So while Flesch’s Kappa measure provided some statistical validation required for our experiments, we conclude that the best assessment is for the reader to determine the results themselves.

Table 1 shows the tweets we retrieved in July 2012 using the top query terms reported for 2010. The original tweet is shown along with the top/bottom three topic labels (and their \mathcal{S} scores). Table 2 on the other hand shows the tweets we retrieved using the top hash tags reported for 2011. The results are presented in the same way as Table 1. We have given a rather comprehensive list of the results for the readers to make their evaluation. At the same time, we also encourage the reader to download the prototype to test it with their own data. The prototype can be downloaded from <http://www.deakin.edu.au/~leong/getTopic>.

5 Related Works

Topic detection has always been an on-going research question, with reference to the research question from as early as 1996 and discussed with greater interest recently by (Young *et. al.* 2004). Much of the research in topic detection started with conventional text documents, for example, news articles drawn from the Reuters-21578³ or Web pages from the Open Directory project⁴, or in newsgroup. Since then, interests in topic detection moved to short texts such as instant messages and SMS as they became popular. Most of the works however were conducted for a conversational model, i.e., an exchange of emails, SMS or instant messages, e.g., in (Dong *et. al.* 2006, Cselle and *et. al.* 2007, Tian *et. al.* 2010). Soon after, the popularity of blogs moved the research to detecting topics for blog posts, e.g., (Zhang *et. al.* 2011, Xu and Oard 2011). As short texts become increasingly common, e.g., status updates and tweets, the research focus once again shifted with works from (AlSumait *et. al.* 2008, Karandikar 2010, Phuvipadawat and Murata 2010, Cataldi *et.al.* 2010, Zhang and Fan and Chen 2011) being good exemplars.

Among these exemplars, (Cataldi *et.al.* 2010)’s work for example, looks at detecting emerging topics for tweets. Their method begins by modelling tweet content as a feature vector where its word terms are then weighted over time against other tweets drawn from a top-level concept. The idea is that terms with a bigger weight becomes candidates for emerging topics. To confirm a candidate as an emerging topic, user authority and content age are considered. Finally, either a supervised or unsupervised selection algorithm is used to pick word terms that qualify as emerging topics. Therefore, while the objective is to detect a topic label for a tweet, the direction is different. Our goal is to detect a topic for a given tweet. (Cataldi *et.al.* 2010)’s method however requires a constant stream of tweets and requires a window before any emerging topics can be reported.

Most recently in (Zhang and Fan and Chen 2011), the problem of detecting topics from chinese short texts was investigated. The authors approached their research by asking two questions: (i) how to determine the keywords (akin to our topics) in the short text; and (ii) how to expand the keywords to track other short texts that have the same ‘topic’ but used different word terms. Their work interests us because

of their method of finding keywords and then expanding them using hyponymies, i.e., a ‘type of’ relationship between word terms. This may be a way for us to expand our top level concept \mathcal{C} without the need to perform a related search. However, how to relate each expanded keyword to a corpus of documents/short texts isn’t immediately obvious.

6 Conclusions

Making sense of short texts is an important research problem as they are becoming increasingly prevalent and ubiquitous. A crucial component to process short texts is the need to know its class or topic label. However, short texts have little features and collectively, has a sparse feature space that makes processing them using conventional algorithms difficult. We present a method to detect topic labels for short texts such as tweets. Our method does not require priori training but produce results that agree well under expert assessment. More importantly, we present a solution that allows the Web to be used as the relevance feedback source. In doing so, our system is guaranteed to be up to date in learning new topic labels. This is crucial in dealing with evolving topics from the large volume of short texts being generated everyday, such as those seen in Twitter.

References

- Fleiss, J. L. (1971) “Measuring Nominal Scale Agreement Among Many Raters.” *Psychological Bulletin*, Vol. 76, No. 5 pp. 378 – 382.
- Landis, J. R. and Koch, G. G. (1977) “The Measurement of Observer Agreement for Categorical Data.” *Biometrics*. Vol. 33, pp. 159 – 174.
- Zhang, C., Fan, X. and Chen, X. (2011) “Hot Topic Detection on Chinese Short Text.” *Springer Berlin Heidelberg*, Vol. 176, pp. 207 – 212.
- Lloret, E. (2009) “Topic Detection and Segmentation in Automatic Text Summarization.” <http://www.dlsi.ua.es/~elloret/publications/SumTopics.pdf>.
- Cataldi, M., Di Caro, L. and Schifanella, C. (2010) “Emerging topic detection on Twitter based on temporal and social terms evaluation.” *The 10th International Workshop on Multimedia Data Mining*. Washington, D.C., ACM, New York, USA. pp. 1 - 10.
- Tian, Y., Wang, W., Wang, X., Rao, J., Chen, C and Ma, J. (2010) “Topic Detection and Organization of Mobile Text Messages.” *The 19th ACM International Conference on Information and Knowledge Management*. Toronto, ON, Canada. ACM. New York, NY, USA. pp. 1877–1880.
- Dong, H., Hui, S.C. and He, Y. (2006) “Structural Analysis of Chat Messages for Topic Detection.” *Online Information Review*, Vol. 30, pp.496–516.
- Perez-Tellez, F., Pinto, D., Cardiff, J. and Rosso, P. (2010) “Clustering Weblogs on the Basis of a Topic Detection Method.” *The 2nd Mexican conference on Pattern recognition: Advances in pattern recognition*. Puebla, Mexico. Springer-Verlag. Berlin, Heidelberg. pp. 342–351.

³<http://www.daviddlewis.com/resources/testcollections/reuters21578/>

⁴<http://dmoz.org/>

- Moens, M.F. and De Busser, R. (2001) "Generic Topic Segmentation of Document Texts." The 24th Annual International ACM SIGIR conference on Research and Development in Information Retrieval. New Orleans, Louisiana, United States, ACM, New York, USA. pp.418–419.
- Cselle, G., Albrecht, K. and Wattenhofer, R. (2007) "BuzzTrack: Topic Detection and Tracking in Email." The 12th International Conference on Intelligent User Interfaces. Honolulu, Hawaii, USA, ACM, New York, USA. pp. 190–197.
- Phuvipadawat, S. and Murata, T. (2010) "Breaking News Detection and Tracking in Twitter." The 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology. IEEE Computer Society, Washington, DC, USA. pp.120–123.
- AlSumait, L., Barbar, D. and Domeniconi, C. (2008) "On-line LDA: Adaptive Topic Models for Mining Text Streams with Applications to Topic Detection and Tracking." The 8th IEEE International Conference on Data Mining. IEEE Computer Society, Washington, DC, USA. pp. 190–197.
- Xu, T. and Oard, D.W. (2011) "Wikipedia-based Topic Clustering for Microblogs." Wiley Subscription Services, Inc., Vol. 48, pp. 1–10.
- Karandikar, A. (2011) "Clustering short Status Messages : a Topic Model based Approach." published PhD thesis, The University of Maryland.
- Manning, C.D., Raghavan, P. and Schtze, H. (2008) "Introduction to Information Retrieval." Cambridge University Press. New York, NY, USA.
- Blei, D.M., Ng, A.Y. and Jordan, M.I. (2003) "Latent Dirichlet Allocation.", JMLR.org, Vol. 3, pp.993–1022.
- Kristina, T. and Christopher, M. (2000) "Enriching the Knowledge Sources used in a Maximum Entropy Part-of-Speech tagger." The 2000 Joint SIGDAT Conference: Empirical Methods in NLP and Very Large Corpora.
- Bernstein, M.S., Suh, B., Hong, L., Chen, J., Kairam, S. and Chi, E.H. (2010) "Eddi: Interactive Topic-based Browsing of Social Status Streams." The 23rd annual ACM symposium on User Interface Software and Technology. New York, USA. ACM, New York, USA. pp. 303–312.
- Hulth, A. (2010) "Improved Automatic Keyword Extraction Given More Linguistic Knowledge." The 2003 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Stroudsburg, PA, USA. pp. 216–223.
- Joachims, T. (1998). "Text Categorization with Support Vector Machines: Learn-ing with Many Relevant Features." In European Conference on Machine Learning (ECML). Springer, Berlin, Germany, pp. 137–142.
- Young, W. S., Sycara, K. (2004). "Text Clustering for Topic Detection." Technical Report (CMU-RI-TR-04-03), Robotics Institute, Carnegie Mellon University.
- Zhang, J., Xia, Y., Ma, B. and Yao, J. (2011) "Thread Cleaning and Merging for Microblog Topic Detection." IJCNLP, 17 December 2011
- Bendersky, M. and Croft, W.B. (2010) "Discovering key concepts in verbose queries." The 31st annual international ACM SIGIR conference on Research and development in information retrieval. Singapore, Singapore. ACM, New York, USA. pp. 491–498.

Table 1: *Tweets and their assigned topic labels obtained by using top query terms reported for 2010.*

Concept \mathcal{C}	Tweet	Topic Labels (top/bottom 3 tweets)	\mathcal{S} Score
Gulf Oil Spill	bp you've a lot to answer for! do my eyes deceive me? a captain's view of dolphin health in the gulf http://huff.to/lqy3ya via @huffpostgreen	Gulf Coast Oil Spill Timeline	0.0620
		Gulf Oil Spill	0.0601
		BP Gulf Oil Spill	0.0600
		Gulf of Mexico Map	0.0149
		Gulf Oil Logo	0.0195
		Oil Spill Clean Up Products	0.0214
	photographs of animal skeletons inspired by the gulf oil spill #photography http://bit.ly/leyl3d	Animals Affected by Oil Spills	0.0256
		Animals in Oil Spills	0.0230
		Animals After Oil Spill	0.0226
		Gulf of Mexico Map	0.0065
		BP Oil Spill	0.0081
	cdc response to the gulf of mexico oil spill http://tinyurl.com/432odhm	Gulf Coast Oil Spill Information	0.0085
		Oil Spill Response	0.0719
		Gulf of Mexico Oil Spill	0.0665
		Gulf Coast Oil Spill Information	0.0570
		Lucas Oil Company History	0.0152
		Gulf of Mexico Map	0.0178
		Gulf Coast	0.0200
Inception	what is the most resilient parasite? bacteria? a virus? an intestinal worm? an idea. leonardo dicaprio inception 2010	Inception Review Ending	0.0657
		Inception Film Review	0.0628
		Inception Explanation Ending	0.0600
		Limitless Torrent	0.0095
		Origin of Hooky	0.0119
		Inception Torrent Kick-Ass	0.0129
	inception is one of the sickest, deepest movies ever	Inception the Movie	0.1487
		Inception Movie	0.1485
		Inception	0.1390
		Origins of Islam	0.0063
		Origin of Hooky	0.0063
	#meta #inception rt @loudboos: seriously, people rt this? rt @jaketapper: ???	Origins of Words	0.0065
		Inception Review	0.0324
		Inception Wiki	0.0294
		Inception Reviews	0.0276
		Limitless Torrent	0.0083
		Origin of Hooky	0.0087
Haiti Earthquake	powered by action in action in response to the haiti earthquake. see the video - http://bit.ly/yjsoph	Inception the Movie	0.0097
		Haiti Earthquake Relief	0.0677
		Haiti Earthquake Relief Charities	0.0606
		Haiti Earthquake Relief Red Cross	0.0587
		Avg. Weather for Dominican Rep.	0.0102
		Mermaid Found in Haiti Pictures	0.0119
	even before the earthquake , conditions in haiti were quite desperate. just behind our hotel in port- http://pinterest.com/pin/235102043018196777/	Television National d'Haiti	0.0127
		Earthquake in Haiti CNN	0.0778
		CNN News Haiti	0.0680
		The Earthquake That Hit Haiti	0.0638
		Television National d'Haiti	0.0098
		Haiti TV	0.0109
	update: did haarp cause the earthquake in haiti? http://bit.ly/npmjjd	Metropole Haiti	0.0110
		The Earthquake That Hit Haiti	0.0814
		Earthquake in Haiti 2010 Article	0.0745
		Date Haiti Earthquake Hit	0.0731
		Haiti TV	0.0091
		Television National d'Haiti	0.0102
		Haiti Radio	0.0105

Concept \mathcal{C}	Tweet	Topic Labels (top/bottom 3 tweets)	\mathcal{S} Score
Vuvuzela	mind, some love blowing their own trumpet rt @p45c4l linking your twitter account to linkedin is like bringing a vuvuzela to a job interview	South Africa Horn	0.0278
		Horns at World Cup	0.0249
		World Cup Noise	0.0240
		Vuvuzela Hero	0.0035
		YouTube Vuvuzela Alpha Blondi	0.0045
		Vuvuzela Video	0.0057
	cek linkedin rt @15june: rt @p45c4l: linking your twitter account to linkedin is like bringing a vuvuzela to a job interview.	Buy World Cup Vuvuzela	0.0168
		World Cup Vuvuzela	0.0164
		Soccer Horn Vuvuzela	0.0162
		Vuvuzela Hero	0.0027
	#loveprotest time: 10:00am where: uhuru park-freedom corner dress code: kenyan colours,carry a vuvuzela	YouTube Vuvuzela Alpha Blondi	0.0034
		Vuvuzela Video	0.0046
		Horns at World Cup	0.0236
		World Cup Noise	0.0230
		South Africa Horn	0.0217
Apple iPad	google's nexus 7 could force apple's hand on 'ipad mini' -	Vuvuzela Hero	0.0029
		Vuvu Hero	0.0053
		YouTube Vuvuzela Alpha Blondi	0.0058
	microsoft surface vs apple new ipad http://bit.ly/mywxqj	iPad Mini 2012	0.0677
		Mini iPad	0.0654
		New Tablets	0.0649
		Apple	0.0096
		AT&T Wi-Fi	0.0106
		Apple iPhone Support	0.0117
	google unveils \$199 tablet to take on ipad - http://interaksyon.com http://fb.me/1aw2h68p7	New Tablets	0.0676
		HP iPad-like	0.0628
		HP iPad Computer	0.0625
		Apple	0.0108
		AT&T Wi-Fi	0.0116
		Apple iPhone Manual	0.0121
Google Android	google's new youtube app for android 4.0 is rolling out today http://tnw.to/n0dj by @harrisonweber	New Tablets	0.0675
		iPad 2 Price Drop	0.0608
		HP iPad On Sale	0.0602
		AT&T Wi-Fi	0.0096
		Apple	0.0135
		Apple iPhone Support	0.0158
	google's new android 4.1 jelly bean os detailed http://bit.ly/n5p5up	Google Android M Downloads	0.0805
		Google AppBrain	0.0691
		Google Market Download	0.0682
		Transaction Fees	0.0098
		What Does Apps Mean	0.0153
		Google Plus Post	0.0156
	google nexus 7 is official, shows off android 4.1 jelly bean http://cnet.co/oxgw6m	Android Ice Cream Sandwich	0.0909
		Ice Cream Sandwich Operating System	0.0772
		Ice Cream Sandwich Android Release	0.0731
		Transaction Fees	0.0178
		What Does Apps Mean	0.0180
		Google Android M Downloads	0.0192
		Ice Cream Sandwich Tablets	0.0707
		Ice Cream Sandwich Android Tablet	0.0704
		Android Tablets 2011	0.0701
		What Does Apps Mean	0.0156
		Transaction Fees	0.0175
		Live Android Downloads	0.0207

Concept \mathcal{C}	Tweet	Topic Labels (top/bottom 3 tweets)	\mathcal{S} Score
Justin Bieber	one direction will be the biggest boyband in the world by the end of this year. - justin bieber	Selena Gomez and Justin Bieber	0.0422
		Justin Bieber Paternity Suit	0.0404
		J.B. Selena Gomez Pregnant	0.0402
		YouTube Videos	0.0140
		Project Live Love	0.0152
		Countdown to 18th Birthday	0.0153
	official: justin bieber's 'believe' is year's biggest debut, bows at no. 1 - http://bit.ly/mtlgfu	How Old Is Justin Bieber	0.0428
		YouTube J.B. Music Videos	0.0395
		Justin Bieber Lyrics	0.0389
		Project Live Love	0.0089
		Happy Birthday 18th	0.0097
		YouTube Videos	0.0101
	niall horan in justin bieber's boyfriend video. http://pic.twitter.com/edcgvwva	Selena Gomez Justin Bieber Kiss	0.0438
		YouTube J.B. Baby Baby	0.0403
		YouTube J.B. Favorite Girl	0.0403
		Project Live Love	0.0120
		Happy Birthday 18th	0.0131
		Countdown to 18th Birthday	0.0141
Harry Potter & the Deathly Hallows	when a muggle saw me reading the deathly hallows book, he asked me "how does harry potter end?" i simply answered "it doesn't."	H.P. SparkNotes Sorcerer's Stone	0.0541
		Harry Potter Reviews	0.0476
		Hogwarts Professor Names	0.0470
		Dumbledore's Army Font	0.0114
		The Wizard Stone	0.0135
		Harry Potter Fun and Games	0.0146
	harry potter and the deadly hallows, part 1 (four-disc blu-ray deluxe edition): the 4-disc ultimate blu-ray edit... http://amzn.to/obfbrt	Harry Potter Reviews	0.0532
		Harry Potter Actors	0.0481
		Harry Potter Film Cast	0.0440
		Dumbledore's Army Font	0.0093
		Harry Potter Fun and Games	0.0119
		Staff Trivia Questions	0.0122
	rt if you cried throughout most of harry potter and the deathly hallows part 2.	Deathly Hallows Movies	0.1217
		Deathly Hallows Official Site	0.1183
		H.P. and the Deathly Hallows	0.1087
		Staff Trivia Questions	0.0060
		The Wizard Stone	0.0083
		Actor Killed Today	0.0097
Pulpo Paul	-hola, c mo te llamas? -yogi, y t ? -paul. - jajaja!, no mames como el pulpo....	Preguntar Al Pulpo Paul	0.0186
		Spanish Octopus Recipes	0.0092
		Spanish Octopus Tapas	0.0091
		Stoneware Drinking Glass	0.0027
		Bell Co51 Octopus Cup Holder	0.0031
		Al Paul Car Wash	0.0032
	i have doubts about today s spain match but if @virginiecapric (the new pulpo paul) says germany - spain, well,here we go to the final!!	Octopus World Cup Prediction	0.0513
		Paul the Octopus Predictions	0.0505
		Paul the Psychic Octopus	0.0485
		What Is Pulpo	0.0064
		Al Paul Car Wash	0.0075
		Stoneware Drinking Glass	0.0088
	paul the octopus is dead actually so im guessing el pulpo ra l too	Preguntar Al Pulpo Paul	0.0374
		Pulpo Recipe	0.0189
		Pulpo Gallego Recipe	0.0177
		Al Paul Car Wash	0.0023
		Bell Co51 Octopus Cup Holder	0.0035
		Make Your Own Coolie Cup	0.0039

Table 2: *Tweets and their assigned topic labels obtained by using top hash tags reported for 2011.*

Concept \mathcal{C}	Tweet	Topic Labels (top/bottom 3 tweets)	\mathcal{S} Score
egypt	another horrific attack on a woman in cairo http://on.cnn.com/lw2hbg #egypt #tahrir	Clashes Egypt	0.0894
		Egypt Virginity Test	0.0752
		Egypt Soccer Game Deaths	0.0719
		Quiz On Middle East	0.0043
		Soccer Game Cup	0.0086
		Greek Gods	0.0114
	#egypt ex-oil min sameh fahmy + hussein salem get 15 yrs: 'squandering public funds' in #israel gas deal http://tinyurl.com/6wdd8sq	Soccer Game Cup	0.0082
		70 Dead Soccer	0.0072
		Pyramid	0.0071
		Egypt God Horus	0.0010
		Proof of Virginity	0.0011
		Map Africa	0.0011
	christians nervous under new president in egypt. http://bit.ly/lvwza0	Clashes Egypt	0.0745
		Egyptian Soccer Riot	0.0526
		Egypt Soccer Game Deaths	0.0485
		Quiz On Middle East	0.0048
		Soccer Game Cup	0.0087
		Greek Gods	0.0088
tigerblood	charlie sheen calls tmz to address hotel lies about him partying okay, we believe you charlie. http://ow.ly/bsppi #tigerblood	Tiger Blood Quote	0.0569
		Charlie Sheen Drinking Tiger Blood	0.0550
		Charlie Sheen Tiger Blood Interview	0.0528
		Paula Deen Riding a Bunchie	0.0061
		Tiger Blood Snow Cone	0.0066
		Alex Pardee T-Shirts	0.0078
	i know charlie sheen aint cool anymore but i still got #tigerblood and im still #winning	Tiger Blood Intern	0.0258
		I Got Tiger Blood	0.0244
		Charlie Sheen Tiger Blood Video	0.0226
		Tiger Blood Snow Cone	0.0029
		Tiger Blood Snow Cone Syrup	0.0033
		Tiger Pharmacy Steroids	0.0050
	power - kanye west is such a good song omg	Tiger Blood Quote	0.0224
		Charlie Sheen Tiger Blood Interview	0.0222
		Charlie Sheen Tiger Blood Comment	0.0211
		Charlie Sheen Tiger Blood Shirt	0.0033
		Tiger Blood Snow Cone	0.0052
		Paula Deen Riding a Bunchie	0.0057
threewordstoliveby	#threewordstoliveby love your life (:	Great Quotes to Live By	0.0236
		Quotes to Live by Tumblr	0.0226
		Great Words to Live By	0.0213
		Lyrics2liveby	0.0014
		Lyrics 2	0.0038
		Lyrics to Live By	0.0040
	#threewordstoliveby loyalty is everything	Great Quotes to Live By	0.0289
		Best Words to Live By	0.0254
		Shook Ones Part 2 Lyrics	0.0250
		Lyrics2liveby	0.0018
		Lyrics to Live By	0.0038
		Tumblr Lyrics to Live By	0.0050
	#threewordstoliveby faith , love, hope	Great Quotes to Live By	0.0304
		Great Words to Live By	0.0273
		Morning Quotes to Live By	0.0259
		Lyrics2liveby	0.0013
		Lyrics 2	0.0039
		Lyrics to Live By	0.0045

Concept \mathcal{C}	Tweet	Topic Labels (top/bottom 3 tweets)	\mathcal{S} Score
japan	mexico s olympic squad to play friendly v le n on july 5 + will face the england, spain and japan olympic squads prior to london olympics.	World Cup Football Japan	0.0327
		USA Japan Game	0.0304
		Japan US Women Soccer	0.0290
		Soft On Demand Sod	0.0038
		SOD Create	0.0040
		Princess of China Lyrics	0.0052
	the japan night life! all of the lights http://instagr.am/p/maxfdcyda6/	Population of Tokyo	0.0250
		Population of Japan	0.0228
		USA Japan Game	0.0210
		Soft On Demand Sod	0.0044
		China Anne McClain	0.0046
	kim soo hyun to head for japan to promote moon that embraces the sun! http://bit.ly/kfrocr	Princess of China Lyrics	0.0064
		Japan Earthquake Anniversary	0.0325
		2011 Japan Earthquake	0.0312
		Earthquake Japan 2012	0.0308
		Princess of China Lyrics	0.0043
		China Anne McClain	0.0047
superbowl	jets fans this man has been working out. look at those arms. with him and sanchez u heard it here first superbowl http://pic.twitter.com/c4xynzmi	Soft On Demand Sod	0.0056
		Super Bowl Odds	0.0356
		Super Bowl 44 Odds	0.0320
		Super Bowl Scores 2012	0.0308
		CBS Local Chicago	0.0023
		2012 Calendar	0.0069
	the supreme court are those dudes who did "superbowl shuffle", right?	2012 Predictions	0.0136
		Super Bowl 2012 New Orleans	0.0216
		2012 Predictions	0.0213
		Super Bowl 2014	0.0205
		Super Bowl 43	0.0058
		CBS Local Chicago	0.0062
	breaking: cnn reports the indianapolis colts have won the super bowl.	superbowl	0.0073
		Super Bowl 2014	0.0121
		Halftime Show Super Bowl 2012	0.0109
		Where Is Super Bowl 2016	0.0106
		CBS Local Chicago	0.0009
		Super Bowl 44 Logo	0.0018
jan25	martyr: ahmed hashim el-sayyed age 25 died in #alex on 28jan #egypt #jan25	Prince Halftime Show Super Bowl	0.0024
		Egyptian Revolution of 1952	0.0320
		Day of Rage Egypt	0.0310
		Revolution Egyptian	0.0301
		DirecTV Revolution 2012	0.0074
		Egyptian Revolution 2011 Photos	0.0083
	martyr: omar fathi nour al-barbari died in maadi on jan28 by family's received his body ...(more) http://bit.ly/lre59j #egypt #jan25	Lending in Bank	0.0086
		Egyptian Revolution of 1952	0.0435
		Day of Rage Egypt	0.0422
		Revolution Egyptian	0.0408
		DirecTV Revolution 2012	0.0070
		Lending in Bank	0.0083
	martyr: aly elnabawy age 55 died in ismailia by gunshots ..., fisher #egypt #jan25	Egyptian Revolution 2011 Photos	0.0087
		Day of Rage Egypt	0.0681
		Egyptian Revolution of 1952	0.0628
		Revolution Egyptian	0.0621
		Egyptian Revolution 2011 Photos	0.0151
		Lending in Bank	0.0158
		25-Jan	0.0159

Unsupervised Text Segmentation using LDA and MCMC

Kaimin Yu Zhe Li Genliang Guan Zhiyong Wang David Feng

School of Information Technologies
University of Sydney
NSW, 2006, Australia

Email: yu.kaimin,zhli8662,genliang.guan,zhiyong.wang,dagan.feng@sydney.edu.au

Abstract

In this paper, we propose a data driven approach to text segmentation, while most of the existing unsupervised methods determine segmentation boundaries by empirically exploring similarity measurement between adjacent units (e.g. sentences). Firstly, we train a latent Dirichlet allocation (LDA) model with the large scale Wikipedia Corpus to avoid the problem of vocabulary mismatch, which makes our approach domain-independent. Secondly, each segment unit is represented with a distribution of the topics, instead of a set of word tokens. Finally, a text input is modeled as a sequence of segment units and Markov Chain Monte Carlo technique is employed to decide the appropriate boundaries. The major advantage of using MCMC is its ability to detect both strong and weak boundaries. Experimental results demonstrate that our proposed approach achieve promising results on a widely used benchmark dataset when compared with the state-of-the-art methods.

Keywords: Text Segmentation, Topic Model, LDA, Markov Chain Monte Carlo (MCMC), Data Driven

1 Introduction

Text segmentation is to divide a given text data into semantically relevant and coherent segments. It is normally consider as an important prerequisite step for other high level semantic text analysis tasks, such as summarization and information retrieval. For example in the context of information retrieval, web pages often vary in length and content, while some short web page may focus on one topic, web pages that contain lengthy documents are likely to address multiple topics. By dividing a document into topic coherent segments, search engines can index the resulting segments based on the topics which will allow users to quickly access information of interest within a lengthy document.

Various unsupervised and supervised approaches have been proposed for text segmentation. In comparison with supervised segmentation algorithms, unsupervised methods require less domain specific knowledge (e.g. *welcome* and *next* in the transcriptions of TV news programs) and more suitable for domain-independent applications. Most of the existing methods in this category utilize lexical cohesion

among segment units (e.g. sentences) (Choi et al. 2001). These approaches often rely on some heuristic rules (e.g. repetition) to derive lexical cohesion. Recently Misra *et al.* proposed to employ topic modelling techniques for text segmentation (Misra et al. 2009). The well established latent Dirichlet allocation (LDA) model was utilized to learn hidden topics in a generative and unsupervised manner and each document is represented as a distribution of topics. Therefore, lexical cohesion is replaced with similarity measurement in terms of topic distribution in calculating pair-wise path scores. In addition, the segments obtained are associated (or labelled) with topic information.

Rather than calculate cumulative scores of potential paths with topic distributions, we formulate text segmentation with a probabilistic problem which can be solved with the unsupervised Markov Chain Monte Carlo (MCMC) technique. As indicated in (Zhai & Shah 2006), MCMC is able to detect both the strong and weak boundaries (Zhai & Shah 2006).

Due to the small training dataset used by (Misra et al. 2009), they had to deal with the problem of vocabulary mismatch (i.e. the difference between vocabularies of training dataset and test dataset). In this work, we investigate the impact of using a large scale web corpus, Wikipedia Corpus¹. It is expected that more representative topics can be discovered from such a large scale corpus and eventually the problem of vocabulary mismatch can be eliminated. Our experimental results indicate that larger dataset help achieve better segmentation performance.

The rest of paper is organized as follows. The related work is reviewed in Section 2. Sections 3 and 4 describe the proposed unsupervised text segmentation method using LDA and MCMC. In Section 5, we compare our method with several state-of-art text segmentation methods and present the experimental results. Finally, conclusions are given in Section 6.

2 Related Work

Linear text segmentation has attracted a significant attention in the field due to its importance in natural language processing tasks, such as information extraction and text summarization. Early approaches (Passonneau & Litman 1997, Beeferman et al. 1999) often exploit the linguistic information such as cue phrases, syntax or lexical features. They assume certain words or phrases can be used to detect the segment boundaries. For example, in TV new programs, cue phrases like “hello and welcome to” and “good evening I’m” typically appears in the beginning of news stories. Conversely, cue phrases, including “stay

Copyright ©2012, Australian Computer Society, Inc. This paper appeared at the 10th Australasian Data Mining Conference (AusDM 2012), Sydney, Australia, December 2012. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 134, Yanchang Zhao, Jiuyong Li, Paul Kennedy, and Peter Christen, Ed. Reproduction for academic, not-for-profit purposes permitted provided this text is included.

¹http://en.wikipedia.org/wiki/Wikipedia:Database_download

with us”, “when we come back” and “weather forecast is next”, often indicates the end of a segment. While cue phrases may convey the document structures, they are normally specific for a type of data and cannot be generalized to other application domains. For each new application, a new set of cue phrases are required to be identified which can be very time consuming and cost prohibitive (Misra et al. 2009, 2010).

The most dominant direction in text segmentation is based on the lexical cohesions (Hearst 1997, Choi 2000, Utiyama & Isahara 2001, Fragkou et al. 2004, Malioutov & Barzilay 2006). These approaches are built around the fact that related or similar words tend to be repeated in topically coherent segments and a change in the vocabulary often indicates segment boundaries. Such approaches normally do not require supervised training, hence they can be applied to any text form any domain. TextTiling (Hearst 1997) is one of the most influential approach in this category. It works by first dividing a document into blocks of fixed number of words which is usually 3-5 sentence long, and the similarity of adjacent blocks is measured based on cosine similarity. The resulting sequence of similarity values is then graphed and smoothed. The local maxima of the word similarity curve indicates that the adjacent blocks cohere well, whereas the local minima is the point of low lexical cohesion and being regarded as a potential segment boundary candidate. However, the numerical value of the similarity is prone to local extrema which has shown to be unreliable (Choi et al. 2001). Choi (2000) replaced the numerical similarity values with its rank in the local region, and used divisive clustering for segmentation in their C99 algorithm. Other techniques have also been used for segmentation. Fragkou et al. (2004) proposed a dynamic programming algorithm to perform the text segmentation by global minimizing the segmentation cost. In order to address the poor segmentation performance caused by smooth topic transitions (weak boundaries), Malioutov & Barzilay (2006) represent a text document as a weighted undirected graph and formalized the text segmentation task as graph partition solved using normalized cut. Kazantseva & S. (2011) proposed to utilize Affinity Propagation clustering algorithm to locate the segment boundaries and segment centers.

Recently topic models are used to compute the similarity. By adopting topic models, the similarity are measured not only based on the exact word repetitions, but also the relations of related words. Choi et al. (2001) applied Latent Semantic Analysis (LSA) (Landauer et al. 1998) in the C99 (Choi 2000) algorithm to measure the sentence similarities where a sentence is represented by the sum of the LSA feature vectors. Their experimental results show that the LSA based similarity measures can significantly outperform the cosine metric used in the original C99 algorithm (Choi et al. 2001). Latent Dirichlet Allocation (LDA) topic models are also exploited by a group of other researchers (Sun et al. 2008, Misra et al. 2009, Riedl & Biemann 2012). Misra et al. (2009, 2011) used the topics discovered by LDA to compute the log-likelihood of each possible segment. The log-likelihood was then used as a score in the dynamic programming algorithm to recover the segmentation from the path that yields the highest log-likelihood (Misra et al. 2009). Sun et al. (2008) used kernel function to measure how much two segments share the same latent topic and dynamic programming for segments selection. Riedl & Biemann (2012) proposed the TopicTiling algorithm which uses topics obtained

by LDA model in a similar fashion as TextTiling uses words.

In our work, LDA is used to compute the pairwise sentence similarity as it is shown to be very effective (Sun et al. 2008, Misra et al. 2009, Riedl & Biemann 2012). Unlike previous approaches, we use the data-driven Markov Chain Monte Carlo (MCMC) techniques to discover the segment boundaries. The major advantage is that MCMC is able to detect both the strong and weak boundaries.

3 Topic modeling with LDA

LDA is a probabilistic generative model to explore the topics of a set of documents. It assumes that each document can be represented by a distribution of the topics and each topic has its underlying multinomial distribution over the vocabulary (Blei et al. 2003). Note that LDA ignores the word orders which means that the words in a document are interchangeable. For example, “topic modeling with LDA” and “LDA with topic modeling” are viewed as completely equivalent by the LDA model.

Given a set of topics $t_i, i = 1, \dots, N_T$ and a vocabulary $W = \{w_i | i = 1, \dots, N_W\}$, LDA assumes a document d can be produced as follows. First, a distribution β_t over the vocabulary is drawn from a Dirichlet distribution for each topic t . Second, a topic distribution θ_d for d is randomly drawn from a Dirichlet distribution. Finally, each word w_i in the document d is generated by selecting a topic according to the topic distribution θ_d and then randomly choosing a word from the chose topic based on the word distribution for the topic β_t . Formally, the probability of the i^{th} word is as follows (Misra et al. 2009):

$$\begin{aligned} P(w_i | \theta_d, \beta) &= \sum_{t=1}^{N_T} P(t_i = t | \theta_d) P(w_i | t_i, \beta) \\ &= \sum_{t=1}^{N_T} \theta_{dt} \beta_{tw} \end{aligned} \quad (1)$$

where θ_{dt} is the probability of using the topic t in the document d and β_{tw} is the probability of using the word w in the topic t .

The topic distribution θ for each document d and the word distribution β for each topic t are the parameters that need to be inferred from a corpus. Gibbs sampling is used to estimated these two model parameters as follows (Griffiths & Steyvers 2004):

$$\theta_{dt} = \frac{K_{dt} + \alpha}{\sum_{k=1}^{N_T} K_{dk} + N_T \alpha} \quad (2)$$

$$\beta_{tw} = \frac{J_{tw} + \lambda}{\sum_{k=1}^{N_W} J_{tk} + N_W \lambda} \quad (3)$$

where K_{dt} is the total number of words in the document d that are assigned to topic t , J_{tw} is the number of times a word w is assigned to a topic t , α and λ are Dirichlet priors.

After obtaining the word distribution β_{tw} for each latent topic t , the topic distribution of an unknown document can be estimated iteratively as (Misra et al. 2008):

$$\theta_{dt}^{n+1} = \frac{1}{L_d} \sum_{w=1}^{N_W} \frac{C_{dw} \theta_{dt}^{(n)} \beta_{tw}}{\sum_{t'=1}^{N_T} \theta_{dt'}^{(n)} \beta_{t'w}} \quad (4)$$

where $\theta_{dt}^{(n)}$ is the value of θ_{dt} at the n th iteration, l_d is the number of words in the document d that are presented in the training vocabulary W , and $C_{d\omega}$ is the count of word ω in d . It should be noted that the words in d but not in W are ignored in this process. Given the topic distribution θ_d for a document d and the word distribution β for all the discovered latent topics, the likelihood of document d can be calculated as:

$$P(C_d | \theta_d, \beta) = \prod_{\omega=1}^{N_W} \left(\sum_{t=1}^{N_T} \theta_{dt} \beta_{t\omega} \right). \quad (5)$$

In this paper, the LDA model is trained using a large scale web corpus, Wikepeida Corpus. It is expected that discovered topics can be more general with boarder applications. We then apply the learned LDA model to a test document for measuring the pairwise sentence similarities. Specifically, we compute the topic distribution for each sentence and measures their Euclidean distance as follows:

$$D(s_i, s_{i+1}) = \left[\left(p(t_1 | s_i) - p(t_1 | s_{i+1}) \right)^2 + \dots + \left(p(t_n | s_i) - p(t_n | s_{i+1}) \right)^2 \right]^{\frac{1}{2}} \quad (6)$$

where $\{t_i | i = 1 \dots n\}$ is the latent topics obtained by the trained LDA model. Once the pairwise sentence similarity matrix is built, the linear text segmentation problem can be solved by the Markov Chain Monte Carlo technique as discussed in Section 4.

4 Boundary detection with MCMC

Linear text segmentation is a process of partitioning a given document into meaningful segments, such that each segment is coherent about a specific topic and consecutive segments are about different topics. In this paper, we consider sentence as the smallest unit that forms a document, hence a segment consists of one or more sentences. Let k denotes the potential number of segments in a document and θ_k denotes their corresponding boundary locations, the general Metropolis-Hasting-Green algorithm (Green 1995) is employed to estimate these two parameter as follows, where $x = k, \theta_k$ and $\pi(x)$ denotes the posterior probabilities of x :

- 1) The parameter x_0 is initialized.
- 2) The followings are conducted in each iteration i .
 - 3) Generate Th_α from $Uni[0, 1]$.
 - 4) Create a new parameter x'_{i-1} based on x_{i-1} with a diffusion or jump.
 - 5) Calculate the radio $\alpha(x_{i-1}, x'_{i-1})$ as:

$$\alpha(x_{i-1}, x'_{i-1}) = \min\left\{1, \frac{\pi(x'_{i-1})q(x'_{i-1}, x_{i-1})}{\pi(x_{i-1})q(x_{i-1}, x'_{i-1})}\right\} \quad (7)$$

- 6) Update $x_i = x'_{i-1}$, if $\alpha > Th_\alpha$. Otherwise, set $x_i = x_{i-1}$

In Equation 7, $q(x, x')$ is the transition probability from state x to x' . Such probability between two states is dependent on the updates types and should be reversible. As in (Zhai & Shah 2006), there are two types of updates that are diffusion and jump. The diffusion update simulates the shifting of boundaries between two adjacent text segments, hence the dimension of the parameter θ_k does not change. The

jump update simulates a pair of reversed actions: split and jump. Split divides a text segment into two parts which increase the dimension of θ_k by 1, while merge combines two adjacent text segments into one thus reducing the dimension of θ_k by 1. The details will be discussed in the following.

4.1 Diffusion

Diffusion is the process of updating the location of the boundary between two adjacent text segments. It uniformly randomly selects a segment boundary and draws a new boundary from a 1D normal distribution with the mean at its original position. Assume t' denotes the new location of the boundary and t denotes its original position, the probability of drawing t' from t can then be calculated as (Zhai & Shah 2006):

$$p(t') = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(t' - t)^2}{2\sigma^2}\right) I(t') \quad (8)$$

where σ is the standard deviation of the movement, and $I(t')$ is a indicator function which is 1 only if the new boundary is within the correct range of the updated segment. Then the forward transition probability for the shift update becomes $q(x, x') = 1/(k-1)p(t')$, and the backward transition probability is $q(x', x) = (1/(k-1))(1 - p(t'))$, where k is the number of segments.

4.2 Jump

The jump update consists of two reversed actions: split and merge. Split divides a original segment $S_m = \{s_m^1, \dots, s_m^n\}$ into two new segments $S'_m = \{s_m^1, \dots, s_m^{t-1}\}$ and $S'_{m+1} = \{s_m^t, \dots, s_m^n\}$, where s_m^t is the new boundary. The data-driven technique (Zhai & Shah 2006) is used to propose the new boundary. We assume uniform probability for selecting scene S_m , the new boundary location t is selected to maximize the likelihood of the new segments as follows:

$$t = \arg \max (\mathbb{L}(S'_m | f'_m) + \mathbb{L}(S'_{m+1} | f'_{m+1})) \quad (9)$$

where $\mathbb{L}(S'_m | f'_m)$ and $\mathbb{L}(S'_{m+1} | f'_{m+1})$ are the likelihood of the two new segments S'_m and S'_{m+1} , f is the features used to measure the sentence similarity. The transition probability for split can then be calculated as:

$$q(x, x') = \frac{1}{k} \mathbb{L}(S'_m | f'_m) \mathbb{L}(S'_{m+1} | f'_{m+1}) \quad (10)$$

Merge is the reversed update of split which combines two adjacent segments into one. As in (Zhai & Shah 2006), we assume uniform probability for selecting segment S_m and combine it with S_{m+1} to form a new segment S'_m . The transition probability can then be easily obtained as follows:

$$q(x, x') = \frac{1}{k-1} \mathbb{L}(S'_m | f'_m). \quad (11)$$

4.3 Posterior Probability

The posterior probability of the two parameters k and θ_k is:

$$p(k, \theta_k | y) \propto \mathbb{L}(y | k) p(\theta_k | k) p(k) \quad (12)$$

```

1  =====
2  Payne dismounted in Madison Place and handed the reins to Herold .
3  There was a fog , which increased the darkness of the night .
4  Two gas lamps were no more than a misleading glow .
5  He might have been anywhere or nowhere .
6  The pretence was that he was delivering a prescription from Dr. Verdi .
7  =====
8  Note : Directions are written for those who have had previous experience .
9  Instructions for preparing clay , drying , glazing and firing are not given .
10 Equipment : Basic pottery studio equipment .
11 Wooden butter molds and cookie presses .
12 =====

```

Figure 1: Two sample segments from the Choi’s “3-5” dataset

where y is the feature selected for computing the sentence similarities, $\mathbb{L}(y | k)$ is the overall data likelihood given θ_k , $p(\theta_k | k)$ is the conditional probability for the boundary locations θ_k given k , and $p(k)$ is the prior probability for the number of segments.

As discussed before, different text segments are about different topics. Hence we can assume that each segment is independent from other, and the overall data likelihood can be calculated as (Zhai & Shah 2006):

$$\mathbb{L}(y | \theta_k) = \left(\prod_{m=1}^L \mathbb{L}(y_m | f_m) \right)^{\frac{1}{L}}. \quad (13)$$

$\mathbb{L}(y_m | f_m)$ is the individual likelihood of data y_m in segment S_m and it is computed as the average of the pairwise similarity value of the sentences within S_m :

$$\mathbb{L}(y_m | f_m) = \text{avg}(\mathbb{M}(a : b, a : b)) \quad (14)$$

where \mathbb{M} is the pairwise sentence similarity matrix obtained using the LDA model, a and b are the first and last sentence in S_m respectively.

The conditional probability for the boundary locations θ_k given k is defined in terms of the combinations as (Zhai & Shah 2006):

$$p(\theta_k | k) = \frac{(k-1)!(T-k)!}{(T-1)!} \quad (15)$$

where T is the total number of sentences in the given document.

As in (Zhai & Shah 2006), we assume the number of segments is drawn from a Poisson distribution as it models the number of incidents happening in a unit time interval. Hence, the model prior is calculated as:

$$p(k) = e^{-\lambda} \frac{\lambda^k}{k!} I(k) \quad (16)$$

where $I(k)$ is an indicator function which equals to 1 if $1 \leq k \leq k_{max}$ and k_{max} is parameter that can be tuned based on the categories and length of the documents of interest.

5 Experiments

5.1 Experimental Settings

Our experiments were carried out with the widely used Choi’s dataset (Choi et al. 2001). The Choi’s dataset used in our experiments consists of 300 documents. Each document consists of ten text segments,

where each segment is comprised of the first “ n ” sentences selected from an article in the Brown corpus. The successive segments within a document are corresponding to different topics. The Choi’s dataset is divided into three subsets (namely “3-5”, “6-8” and “9-11”) based on the lengths of text segments “ n ”. For example, the Choi’s “3-5” dataset contains the segment with the length of 3 to 5 sentences. Figure 1 shows two successive segments from the “3-5” dataset. In order to investigate the impact of the training data size on segmentation performance, we also created 3 different datasets (namely A, B, and C) for training the LDA model by sampling the Wikipedia Corpus every 100, 50, and 10 entries, respectively.

Following previous research (Griffiths & Steyvers 2004), we set the Dirichlet priors (α and β) of the LDA model to (1 and 0.01), the number of topic to 200 (after a number of trials from 10 to 500), and the number of iterations to 600 (after a number of trials from 100 to 2000). For MCMC technique, the shifting distance variance is set to 3, the number of independent Markov chain to 200, and the iteration for each chain to 1000.

The evaluation protocol is the standard P_k (probabilistic error metric) which is the probability that two randomly drawn sentences which are K sentences apart are classified incorrectly. The higher value of P_k indicates lower accuracy in text segmentation. Compared to the conventional precision and recall measures, P_k penalizes near misses less than pure false positive and false negative, hence more accurately reflecting the segmentation performance.

5.2 Results and Discussions

5.2.1 Impact the of the size of the training dataset

The impact of the length of the text segments and the size of the training corpus on the segmentation performance is studied. As show in Figure 2, the segmentation performance consistently increases when the segment size increases from “3-5” to “9-11”, which suggests that longer segments allow a more reliable estimation of the topic distribution by the LDA model. Moreover, it is observed that the larger the training corpus, the better segmentation performance can be obtained. As discussed in Section 3, during the estimation process, LDA drops the words which do not appear in the training process. Hence if there is a significant vocabulary mismatch between the training and testing data, potentially a large amount of words in the testing document will be dropped which can result in a great reduction of information thus affecting the segmentation performance. To demonstrate this, we trained our *LDA + MCMC* model us-

ing three Wikipedia corpus of different sizes, namely A (smallest), B (middle), and C (largest). The result is shown in the last three rows in Table. 1. As can be seen, the model $LDA + MCMC(C)$ trained using the largest corpus C obtains the best performance due to the fact that the vocabulary size increases proportional to the size of the training corpus, thus it has a better chance to cover the testing vocabularies.

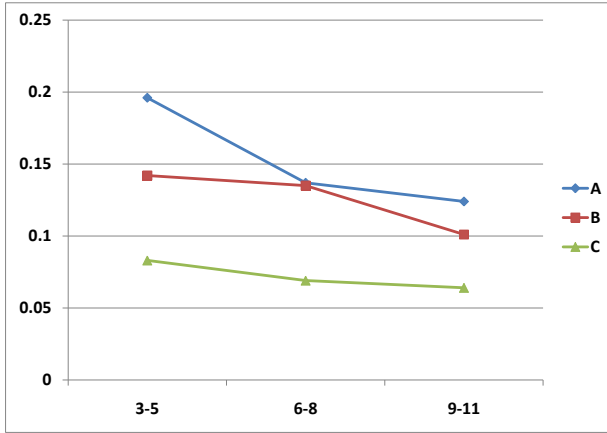


Figure 2: The impact of the length of the segment and the size of the training dataset on segmentation performance. The experiment is performed on the three Choi’s subset, namely “3-5”, “6-8” and “9-11”. The lower the result, the better the performance.

5.2.2 Comparison with the state of the art

Our approach is also compared with the other state of the art methods. As shown in Table 1 where methods are sorted in chronicle order, our proposed approach achieves promising results benchmarked with the Choi’s dataset. Specifically, our approach performs better than Unadapted LDA approach (Misra et al. 2009), which indicates the contribution from the MCMC technique. Though LDA (Adapted) approach achieves better result than our method, part of the Choi’s dataset is required for training the LDA model to avoid the problem of vocabulary mismatch. Compared with the JSeg approach (Nguyen et al. 2011) which utilizes non-systematic relation in lexical cohesion, our approach also demonstrates better segmentation accuracy. JSegT approach further improves the segmentation performance when topic based similarity is combined with lexical distance (with empirically set combination weight). Interestingly, we are not able to achieve the similar gain when taking such combination into our MCMC based approach. It is worthwhile to investigate the fusion of different similarity measurements in the MCMC framework.

Methods	3-5	6-8	9-11	Avg
JTextTile	0.473	0.513	0.533	0.506
C99	0.115	0.104	0.112	0.110
TextSeg	0.090	0.070	0.050	0.070
MinCutSeg	0.340	0.241	0.174	0.252
LDA (Unadapted)	0.230	0.158	0.144	0.177
LDA (Adapted)	0.022	0.023	0.041	0.029
JSeg	0.091	0.107	0.121	0.106
JSegT	0.020	0.030	0.046	0.032
LDA+MCMC	0.083	0.069	0.064	0.072

Table 1: Comparison of segmentation performance on the Choi’s dataset

6 Conclusions

We present an approach to text segmentation by combining the LDA model and the MCMC technique. Both methods are unsupervised and data driven, which makes our approach domain-independent. Our approach also achieve promising results on the benchmark dataset, when compared with the state-of-the-art methods. In the future, we will investigate the close integration of LDA and MCMC and further evaluate the proposed approach with topic models obtained from different datasets.

References

- Beeferman, D., Berger, A. & Lafferty, J. (1999), ‘Statistical models for text segmentation’, *Machine learning* **34**(1), 177–210.
- Blei, D., Ng, A. & Jordan, M. (2003), ‘Latent Dirichlet Allocation’, *Journal of Machine Learning Research* **3**, 993–1022.
- Choi, F. (2000), Advances in domain independent linear text segmentation, in ‘Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference’, Morgan Kaufmann Publishers Inc., pp. 26–33.
- Choi, F., Wiemer-Hastings, P. & Moore, J. (2001), Latent semantic analysis for text segmentation, in ‘Proceedings of EMNLP’, pp. 109–117.
- Fragkou, P., Petridis, V. & Kehagias, A. (2004), ‘A dynamic programming algorithm for linear text segmentation’, *Journal of Intelligent Information Systems* **23**(2), 179–197.
- Green, P. (1995), ‘Reversible jump markov chain monte carlo computation and bayesian model determination’, *Biometrika* **82**(4), 711.
- Griffiths, T. & Steyvers, M. (2004), ‘Finding scientific topics’, *Proceedings of the National Academy of Sciences of the United States of America* **101**(Suppl 1), 5228.
- Hearst, M. (1997), ‘TextTilling: Segmenting texts into multi-paragraph subtopic passages’, *Computational Linguistics* **23**(1), 33–64.
- Kazantseva, A. & S., S. (2011), Linear Text Segmentation Using Affinity Propagation, in ‘Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing’.
- Landauer, T., Foltz, P. & Laham, D. (1998), ‘An Introduction to Latent Semantic Analysis’, *Discourse Processes* **25**(2-3), 259–284.
- Malioutov, I. & Barzilay, R. (2006), Minimum cut model for spoken lecture segmentation, in ‘Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics’, Association for Computational Linguistics, pp. 25–32.
- Misra, H., Cappé, O. & Yvon, F. (2008), ‘Using LDA to detect semantically incoherent documents’, *Proc. of CoNLL 2008* pp. 41–48.
- Misra, H., Hopfgartner, F., Goyal, A., Punitha, P. & Jose, J. (2010), ‘Tv news story segmentation based on semantic coherence and content similarity’, *Advances in Multimedia Modeling* pp. 347–357.

- Misra, H., Yvon, F., Cappé, O. & Jose, J. (2011), 'Text segmentation: A topic modeling perspective', *Information Processing & Management* **47**(4), 528–544.
- Misra, H., Yvon, F., Jose, J. & Cappe, O. (2009), Text segmentation via topic modeling: an analytical study, in 'Proceeding of the 18th ACM conference on Information and knowledge management', ACM, pp. 1553–1556.
- Nguyen, V., Nguyen, L. & Shimazu, A. (2011), 'Improving text segmentation with non-systematic semantic relation', *Computational Linguistics and Intelligent Text Processing* pp. 304–315.
- Passonneau, R. & Litman, D. (1997), 'Discourse segmentation by human and automated means', *Computational Linguistics* **23**(1), 103–139.
- Riedl, M. & Biemann, C. (2012), TopicTiling: A Text Segmentation Algorithm based on LDA, in 'Student Research Workshop of the 50th Meeting of the Association for Computational Linguistics'.
- Sun, Q., Li, R., Luo, D. & Wu, X. (2008), Text segmentation with lda-based fisher kernel, in 'Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers', Association for Computational Linguistics, pp. 269–272.
- Utiyama, M. & Isahara, H. (2001), A statistical model for domain-independent text segmentation, in 'Proceedings of the 39th Annual Meeting on Association for Computational Linguistics', Association for Computational Linguistics, pp. 499–506.
- Zhai, Y. & Shah, M. (2006), 'Video scene segmentation using Markov Chain Monte Carlo', *IEEE Transactions on Multimedia* **8**(4), 686–697.

CRUDAW: A Novel Fuzzy Technique for Clustering Records Following User Defined Attribute Weights

Md Anisur Rahman and Md Zahidul Islam

Centre for Research in Complex Systems, School of Computing and Mathematics,
Charles Sturt University, Panorama Avenue, Bathurst, NSW 2795,
Australia.

{arahman, zislam}@csu.edu.au

Abstract

We present a novel fuzzy clustering technique called CRUDAW that allows a data miner to assign weights on the attributes of a data set based on their importance (to the data miner) for clustering. The technique uses a novel approach to select initial seeds deterministically (not randomly) using the density of the records of a data set. CRUDAW also selects the initial fuzzy membership degrees deterministically. Moreover, it uses a novel approach for measuring distance considering the user defined weights of the attributes. While measuring the distance between the values of a categorical attribute the technique takes the similarity of the values into consideration instead of considering the distance to be either 0 or 1. Complete algorithm for CRUDAW is presented in the paper. We experimentally compare our technique with a few existing techniques – namely SABC, GFCM, and KL-FCM-GM based on various evaluation criteria called Silhouette coefficient, F-measure, purity and entropy. We also use t-test, confidence interval test and time complexity in evaluating the performance of our technique. Four data sets available from UCI machine learning repository are used in the experiments. Our experimental results indicate that CRUDAW performs significantly better than the existing techniques in producing high quality clusters.

Keywords: Clustering, Fuzzy Clustering, Hard Clustering, Cluster Evaluation, Data Mining.

1 Introduction

Clustering is a process of grouping similar records in a cluster and dissimilar records in different clusters. The records within a cluster are more similar to each other than the records in different clusters (Han and Kamber 2006, Tan et al. 2005). Therefore, clustering extracts hidden patterns (from large data sets) that can help in decision making processes. It has a wide range of applications including social network analysis, DNA analysis, software engineering, crime detection, medical imaging, market segmentation, and search result grouping (Zhao and Zhang 2011, Haung and Pan 2006, Songa and Nicolae 2008, Lung et al. 2003, Grubescic and Murray 2001, Tsai and Chiu 2004, Zamir and Etzioni 1999, Masulli and Schenone 1998). Hence, it is important to

produce good quality clusters for supporting decision making processes.

There is always room for further improvements in the existing clustering techniques. For example, a group of techniques select initial seeds randomly (Saha et al. 2010, Hasan et al. 2009, Redmond and Heneghan 2006). Due to the random selection of initial seeds, they end up producing different sets of clusters in different runs resulting in an uncertainty of cluster quality in a run. Good quality of initial seeds is crucial for good quality clusters (Rahman and Islam 2011). Some other clustering techniques require various user inputs including number of clusters which can often be very difficult for a user/data miner to provide (Lee and Pedrycz 2009, Chatzis 2011, Ahmad and Dey 2007a, Saha et al. 2010).

Moreover, most of the existing clustering techniques consider that all attributes of a data set are equally important for clustering. That is, the weights (significance levels) of all attributes of a data set are considered to be equal. In all clustering steps including measuring distance, between a record and a seed, all attributes are used with the same significance/importance level, say 1. The clustering techniques do not allow a data miner/user to assign different significance levels such as 1, 0.8, 0.2 and 0 to different attributes as appropriate/desired. A data miner can either ignore (i.e. assign significance level equal to 0) or consider (i.e. assign significance level equal to 1) an attribute while clustering the records. It would be very useful if a clustering technique could provide a data miner with the flexibility to assign different significance levels (anything between 1.0 and 0.0) to different attributes. This would help a data miner to explore various sets of clusters using different weight arrangements for the attributes of a data set.

In many existing fuzzy clustering techniques initial fuzzy membership degrees of the records are assigned randomly (Lee and Pedrycz 2009, Alata et al. 2008, Bezdek 1981, Hathaway and Bezdek 1988). Hence, a data miner may get different clustering results in different runs of a fuzzy clustering technique resulting in similar problems to the case where initial seeds are selected randomly.

Some techniques are suitable either for data sets having only numerical attributes or for data sets having only categorical attributes (Bai et al. 2011, Li et al. 2008, Guha et al. 1988, Zhang et al. 1996), while in reality data sets often have both numerical and categorical attributes. Although there are techniques that can handle both numerical and categorical attributes (Huang 1997, Ji et al. 2012), some of them do not consider any similarity

between categorical values in the sense that if two categorical values (of an attribute belonging to two records) are different then the distance between the two records in terms of the attribute is considered to be 1 (regardless of the similarity of the values), and otherwise 0.

In this study, we present a novel fuzzy clustering technique called **Clustering Records Following User Defined Attribute Weights (CRUDAW)**. The key contributions of our proposed technique are as follows.

In CRUDAW, we use high quality initial seeds obtained through a deterministic process based on the density of the records of a data set. Besides, the number of clusters is automatically defined through the clustering process without requiring a user input on this. Moreover, it allows a user to assign different significance levels (ranging from 0.0 to 1.0) to different attributes and cluster the records accordingly. If the significance of an attribute is advised to be 0 the technique totally ignores the attribute while clustering the records, whereas if the significance of an attribute is something between (0.0, 1.0] CRUDAW considers the influence of the attribute according to its weight. For example, while calculating the distance between two records (as part of the steps of clustering) the technique considers the weights of the attributes, where the influence of an attribute is greater when the weight of the attribute is higher.

Note that CRUDAW offers more options than just the traditional two options; i.e. either consider an attribute or ignore the attribute while clustering records. A data miner may want to cluster people mainly based on career related information, but also may want to give some importance to the demographic information. In that case he/she may want to assign high weights (such as 0.9 and 0.7) on the career related attributes and low weights (such as 0.1 and 0.4) on demography related attributes, and zero weights on all other attributes. If a user chooses to cluster records considering say three attributes with weights 1.0, 0.7, and 0.2, respectively then he/she gets a clustering result that is likely to be different to the clustering result he/she would get if he/she had chosen even the same three attributes with different weights say 0.4, 0.9 and 0.6.

Another interesting property of our technique is that it calculates the distance between two categorical values based on their similarity, instead of considering the distance either 1 (if the values are different) or 0. The distance between two categorical values can therefore be anything between 0.0 and 1.0. Hence, our technique is suitable for data sets having only numerical, only categorical or both numerical and categorical attributes. Additionally, we determine the initial fuzzy membership degree of the records from the initial seeds that are selected deterministically, and thereby avoid the randomness of initial membership degree.

We compare the cluster quality of our proposed technique with a few other top quality exiting techniques called SABC, GFCM, and KL-FCM-GM (Ahmad and Dey 2007a, Lee and Pedrycz 2009, Chatzis 2011). We use four publicly available data sets that are obtained from UCI machine learning repository (UCI 2012). Several commonly used criteria namely Silhouette

coefficient, F-measure, entropy, and purity (Chuang 2004, Tan et al. 2005, Kashef and Kamel 2009) are used to evaluate the technique. The experimental results clearly indicate that the quality of clusters produced by CRUDAW is better than the quality of clusters produced by the top class existing techniques.

The structure of the paper is as follows. In Section 2, we discuss some existing clustering techniques. Our novel clustering technique is presented in Section 3. We present experimental results in Section 4 and give concluding remarks in Section 5.

2 Literature Review

In this study, we consider a data set as a two dimensional table (see Table 1) with a number of columns (attributes) and rows (records). Attributes of a data set can be categorical and numerical. In our example data set there are ten records, six categorical attributes (Marital-Status, Qualification, Occupation, Professional-Training, Country-of-Origin, and First-Language), and one numerical attribute (Age). We can group the attributes of the data set into three categories namely demographic, career, and background as shown in Table 1. The domain values for categorical attribute Marital-Status are {Single, Married}. Similarly, the domain values of all other categorical attributes can be learnt from Table 1.

Clustering is a data mining task that groups similar records in a cluster and dissimilar records in different clusters. Similarity of records are typically measured based on their distances. For the purpose of clustering, the distance between two numerical attribute values can be measured based on Euclidian distance since numerical values exhibit a natural ordering among them. For a categorical attribute, the distance between two categorical attribute values are typically considered to be either zero or one. However, it may not be sensible to consider the distance between two categorical attribute values either zero or one. The distance between two categorical values can depend on their similarity (Islam and Brankovic 2011, Rahman and Islam 2011). The similarity between two categorical values are generally measured based on their co-appearance (connection) with the domain values of other categorical attributes among the records of a data set (Giggins 2009, Ganti et al. 1999).

To calculate similarity, a data set is first converted into a graph by considering all categorical attribute values of a data set as vertices of the graph (Giggins 2009). Co-appearances of two attribute values are used for drawing the edges between the vertices representing the values. Let, $S_{p,q}$ be the similarity for categorical attribute values p and q , v be the total number of vertices, a_{pt} be the number of edges between vertices p and t (where t represents the domain value of another categorical attribute), a_{tq} be the number of edges between vertices t and q , and $d(p)$ and $d(q)$ be the degrees of vertices for p and q , respectively. The similarity between two categorical attribute values (p and q) belonging to an attribute can be calculated with respect to another value t belonging to another attribute as follows.

$$S_{p,q} = \frac{\sum_{t=1}^p \sqrt{a_{pt} \times a_{tq}}}{\sqrt{d(p) \times d(q)}} \quad (1)$$

If a data set has both categorical and numerical attributes, we suggest that the numerical attribute values can be first categorized and then the similarity between two categorical attribute values can be calculated based on both categorical and numerical (categorized) attribute values. Similarity of categorical attribute values can be useful in clustering records of a data set having categorical attributes.

For a data set having numerical attributes, K-Means is one of the most widely used clustering techniques. A user first needs to define the desired number of clusters. K-Means then selects as many seeds as the user defined number of clusters where each seed, which is a record, is chosen randomly (Han and Kamber 2006, Tan et al. 2005). Distances between a record and all the seeds are calculated. The record is assigned to the seed with which it has the minimum distance. Each record is assigned to only one seed. Records assigned to the same seed are considered to be a cluster.

distributes the records of a data set among a user defined number of clusters. It then determines the center (seed) of each cluster in a way so that instead of having a single value of a categorical attribute, the seed contains all categorical values of an attribute proportionate to their frequencies within the records belonging to the cluster.

The distances between a record and all seeds are then calculated. In order to compute the distance between a record and a seed SABC calculates the distance between them for each attribute; both categorical and numerical. Distance between two categorical values is calculated with respect to their co-appearance with values of another attribute (Ahmad and Dey, 2007b). Another interesting property of SABC is that it automatically (not user defined) computes the significance of each numerical attribute which is then used the distance calculation function.

A record is then assigned to the seed with which it has the minimum distance among all seeds. After the allocation of all records to their nearest seeds SABC moves to the next iteration where it calculates a new set of seeds as before based on the new arrangement of the records. The process of record re-allocation and seed

Record	Demographic		Career			Background	
	Age	Marital-Status	Qualification	Occupation	Professional-Training	Country-of-Origin	First-Language
R ₁	65	Married	PhD	Academic	No	Australia	English
R ₂	30	Single	Master	Engineer	No	Bangladesh	Non-English
R ₃	45	Married	Master	Engineer	No	India	Non-English
R ₄	30	Single	Bachelor	Physician	Yes	Australia	English
R ₅	55	Married	PhD	Academic	No	Australia	English
R ₆	35	Single	Bachelor	Physician	Yes	India	Non-English
R ₇	60	Married	PhD	Academic	No	Bangladesh	Non-English
R ₈	45	Single	Bachelor	Physician	Yes	Australia	English
R ₉	35	Single	Master	Engineer	Yes	India	Non-English
R ₁₀	42	Married	Master	Engineer	No	Australia	English

Table 1: An Example Data Set

While calculating the distance between a record and a seed, typically Euclidian distance between two numerical values belonging to an attribute is used. However, for measuring the distance between two categorical values 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 (Huang 1997, Ji et al. 2012).

K-Means then re-calculates the seeds based on the records belonging to each cluster. Generally a seed is calculated by taking the average of a numerical attribute, and the mode value of a categorical attribute among all records belonging to a cluster. After new seeds are selected all records are again reorganized in such a way 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 condition is satisfied. Typically, a user defined number of iteration and/or a minimum difference between the seeds are considered as termination conditions.

Another existing technique (Ahmad and Dey 2007a) uses fuzzy seeds while performing clustering through a modified version of K-Means. We call the technique as SABC throughout the paper. SABC first randomly

selection continues until a maximum number of iteration is complete or the clusters stabilise.

All clustering techniques discussed so far are hard clustering where a record belongs to only one cluster. There is another type of clustering called fuzzy clustering where a record has some attachment/relationship with all clusters, instead of just with one cluster. Fuzzy C-Means (FCM) is one of the most commonly used fuzzy clustering techniques, which explores such fuzziness nature of the records (Bezdek 1981, Huang and Ng 1999). For each record, FCM assigns a fuzzy membership degree for the record and a cluster in order to represent the level of attachment between the record and the cluster.

Some early FCM techniques can handle a data set having only numerical attributes. However, there are FCM techniques such as General Fuzzy C-Means (GFCM) that can handle data sets having both numerical and categorical attributes (Lee and Pedrycz 2009). General Fuzzy C-Means (GFCM) uses the following clustering steps.

1. Takes a user defined number of clusters.
2. Randomly assigns a fuzzy membership degree for a record and a cluster; for all records and all

clusters. Let, μ_{ij} be the fuzzy membership degree of the i th record with the j th cluster. GFCM chooses initial fuzzy membership degrees in such a way so that $\sum_{j=1}^k \mu_{ij}=1; \forall i$, where k is the total number of clusters.

3. Using the fuzzy membership degrees of the records, cluster centers are re-calculated as we explain below in the description of the steps.
4. A new set of fuzzy membership degrees is calculated (as explained below) for every record considering the new cluster centers as calculated in Step 3.
5. Repeat Step 3 and Step 4 until a termination condition is met.

We now explain the process of cluster center calculation (Step 3) and fuzzy membership degree calculation (Step 4) in the following paragraphs.

Each attribute value of a center is calculated using the values of the attribute for all records of the data set. The seed value of a numerical attribute is calculated as the same way Fuzzy C-Means does. For a categorical attribute, GFCM uses a fuzzy seed value where the seed contains each value of the domain of the attribute according to a confidence degree (Kim et al. 2004). The confidence degree of an attribute value a_1 is the sum of the membership degrees (with the cluster) of all records having a_1 .

Once the cluster centres are calculated – based on the distance between a record and a centre, a new set of membership degrees is calculated for the center and the records (Step 4). The fuzzy membership degree of a record and a cluster center (seed) is inversely proportional to the distance between the record and the seed. For numerical attributes, GFCM calculates normal Euclidean distance. However, for categorical attributes it calculates distance based on frequency of attribute values.

GFCM repeats Step 3 and Step 4 until a termination condition is met i.e. either a user defined number of iteration is completed or the objective function is minimized.

Another fuzzy clustering technique called Kull-back-Leibler FCM (KL-FCM-GM) can handle a data set having both categorical and numerical attributes (Chatzis 2011). It is an extension of Gath-Geva algorithm, which is a well-known fuzzy clustering technique that works on a data set having only numerical attributes (Gath and Geva 1989).

It randomly chooses the initial fuzzy membership degree for a record in such a way so that $\sum_{j=1}^k \mu_{ij}=1$, where k is the user defined number of clusters and μ_{ij} is the fuzzy membership degree of i th record with j th cluster. It then calculates the weight of each cluster. The weight of each cluster is the summation of the fuzzy membership degrees of the records with that cluster divided by the total number of records of a data set.

Using the weights of the clusters and the distances between a record and the seeds, KL-FCM-GM then re-calculates the fuzzy membership degrees of each record. The membership degree of a record with a cluster is higher than the fuzzy membership degree of the record

with another cluster, if the weight of the former cluster is higher than the weight of later cluster, and the distance between the record and the seed of the former cluster is less than the distance between the record and the seed of the later cluster.

KL-FCM-GM next repeats the steps to re-calculate the weights of the clusters and fuzzy membership degrees of the records. The clustering process continues until a user defined number of iterations is reached or the values of the objective function converge.

3 Our Clustering Technique

We present a novel clustering technique called “Clustering Records Following User Defined Attribute Weights” (CRUDAW). In this section we first discuss the basic concepts used in our clustering technique. We then introduce different components used in various steps of the clustering technique.

3.1 Basic Concepts

Most of the existing clustering techniques only allow a user (data miner) either to consider or totally ignore an attribute while clustering the records. All attributes that are considered have equal influence in clustering the records. That is, most of the existing techniques do not allow a user to assign different significance levels (weights) to different attributes. All attributes that are considered for clustering have weight equal to 1 and all attributes that are ignored have weight equal to 0. Nevertheless, it can often be crucial for a user to consider a few attributes with high weights, and some other attributes with low weights while ignoring the remaining attributes for clustering. For example, different users may want to cluster records of our example data set (see Table 1) for different purposes such as grouping people according to their similarity based on career and finding groups of similar people based on their background. Therefore, a user may want to cluster the records from different perspectives including career and background related view point.

A user (data miner) wanting to explore the clusters from a career point of view may want to assign high weights on career related attributes, low weights on demographic attributes and zero weights (completely ignore) on background related attributes. An example of a weight distribution can be 0.3, 0.2, 0.8, 0.8, 0.6, 0.0 and 0.0 for the attributes Age, Marital-Status, Qualification, Occupation, Professional-Training, Country of Origin, and First Language, respectively. Note that the attributes Qualification, Occupation, and Professional-Training are considered to be career related attributes and therefore, given high weights 0.8, 0.8 and 0.6. Based on the weight distribution a possible clustering result can be three clusters having {R2, R3, R9, R10}, {R4, R6, R8} and {R1, R5, R7} records, respectively (see Table 1).

However, another user may want to cluster the records mainly based on background information and therefore, assign high weights on background related attributes. In that case, an example of possible weight assignment can be 0.2, 0.3, 0.0, 0.0, 0.0, 0.6, and 0.9 weights on the

attributes, respectively. A possible clustering outcome can be a set of three clusters having {R1, R5, R7, R10}, {R2, R3, R6, R9} and {R4, R8} records, respectively. The records are clustered differently for the two users. For the first user a record (say R10) is clustered with a set of records (R2, R3, and R9) whereas the same record R10 is clustered together with a different set of records (R1, R5, and R7) for the second user, due to different weight assignments on the attributes.

Similarly, the clusters can be very different even for the first user if he/she assigns a different weight pattern for the attributes of his interest. For example, the first data miner could also use a different weight pattern 0.2, 0.3, 0.7, 0.8, 0.9, 0.0 and 0.0 for the attributes, respectively. Note that the first user still assigns high weights on career related attributes.

Therefore, following the underlying approach of some previous studies (Rahman and Islam 2011, Islam and Brankovic 2011, Islam 2008, Islam and Brankovic 2005) we propose a novel clustering technique allowing a user to assign different weights on different attributes. We find in the literature another clustering technique called SABC (Ahmad and Dey 2007a) that automatically (not user defined) calculates the weights of the attributes. However, the technique does not allow a user to assign weights according to the requirements of a user.

Moreover, in an initial experiment we do not find the calculated weights (by SABC) to be matching with the actual weights that are calculated according to an entropy analysis as follows. We carry out an experiment on the Breast Cancer data set available from UCI Machine Learning repository (UCI 2012). The data set has a natural class attribute (also called label of a record) to indicate the diagnosis for each patient. We then calculate the entropy, gain and gain ratio (Quinlan 1993, Quinlan 1996, Islam 2012) of each attribute in order to explore their significance levels with respect to the natural class attribute. We also calculate the significance (weight) of an attribute according to SABC. We find that the attributes having high weights according to the entropy analysis (i.e. low entropy, high gain and high gain ratio) do not necessarily have high weights according to SABC calculation.

Attribute Name	Significance (SABC)	Attribute Name	Entropy
inv-nodes	0.414614011	deg-malig	0.783292426
age	0.323475993	inv-nodes	0.789404198
menopause	0.232572616	tumor-size	0.810368462
node-caps	0.225719134	node-caps	0.815943182
tumor-size	0.215500036	irradiat	0.837121642
irradiat	0.186524756	age	0.851096449
deg-malig	0.157275391	menopause	0.860274546
breast-quad	0.14948599	breast-quad	0.863183885
breast	0.095167189	breast	0.870592539

Figure 1: Comparative study on significance of the attributes according to SABC and conventional entropy calculation using Breast Cancer data set

We understand that typically a data set used for clustering does not have a natural class attribute. However, one of the main purposes of clustering is to assign such a label (class value) to an unlabeled record. Therefore, following a reverse engineering approach the

use of entropy analysis for finding possible significance levels is used.

In Figure 1, the first two columns show the attributes and their significance values, respectively as calculated by SABC. The third and fourth columns show the attributes and their entropy values, respectively. According to entropy calculation “deg-malig” is the most important attribute (having the least entropy), whereas according to significance calculation the attribute is only the 7th most important attribute. Similarly, according to the significance calculation “age” and “menopause” are the 2nd and 3rd most important attributes, respectively while they are only the 6th and 7th important attributes in terms of entropy (Figure 1).

Unlike many existing techniques (Ahmad and Dey 2007a, Lee and Pedrycz 2009, Chatzis 2011), our proposed technique uses a deterministic process (instead of a random process) in order to identify high quality initial seeds for clustering. High quality initial seeds are very important for a high quality clustering as evidenced in a previous study (Rahman and Islam 2011). However, unlike the previous technique, in this study we take an approach to identify high quality initial seeds with low time complexity.

Moreover, unlike many existing techniques our proposed technique does not require a user to give the number of clusters as an input. We argue that the estimating the number of clusters in advance can be a difficult job for a user. However, the proposed technique instead takes the radius for a seed as an input. Note that the radius is only used to find the initial seeds through a deterministic process. It is not used as the radius of the final clusters.

We identify the initial seeds and thereby initial fuzzy membership degrees in a deterministic way. Moreover, our proposed technique uses a novel approach for distance calculation, between two records, where the attributes having higher significance have higher influence in the distance calculation for two records. The proposed technique also calculates the similarity (anything between 0 and 1) among the values of a categorical attribute and uses the similarity in order to calculate their distance, unlike many existing techniques (Huang 1997, Ji et al. 2012) that consider the distance to be either 1 (if the values are different) or 0 (otherwise). Thus, our technique can handle better the data sets having numerical and/or categorical attributes.

3.2 CRUDAW: A Novel Fuzzy C-Means Clustering Technique

We now first introduce the components of CRUDAW and then use them to introduce the technique in details as follows.

SiDCAV: Similarity based Distances for Categorical Attribute Values

We use the similarity of two values C_1 and C_2 (belonging to a categorical attribute) to calculate their distance as follows.

$$distance(C_1, C_2) = 1 - similarity(C_1, C_2) \quad (2)$$

The similarity of C_1 and C_2 can be calculated using an existing technique (Giggins 2009) that has been discussed in detail in Section 2 on Background Study. The similarity of any two categorical values can vary between 0 and 1.

NoNAV: Normalized Numerical Attribute Values

In order to maintain the consistency between a categorical and a numerical attribute in influencing the distance between two records, we normalize a numerical attribute so that its domain ranges between 0.0 and 1.0. Therefore, after normalization the distance between two values belonging to a numerical attribute can vary between 0.0 and 1.0, similar to a categorical attribute. The normalization is obtained as follows.

$$N(v) = \frac{(v - \min)}{(\max - \min)} \quad (3)$$

where $N(v)$ is the normalized numerical value, v is the original value of a numerical attribute, and \min and \max are the minimum and maximum values of the domain of the attribute.

WeDiF: Weighted Distance Function

We calculate the distance between the i th and the j th record, R_i and R_j using a novel weighted distance function as follows.

$$\text{dist}(R_i, R_j) = \frac{\sum_{a=1}^{m_1} w_a |R_{i,a} - R_{j,a}| + \sum_{a=m_1+1}^m w_a * \text{distance}(R_{i,a}, R_{j,a})}{\sum_{a=1}^m w_a} \quad (4)$$

Here, $R_{i,a}$ and $R_{j,a}$ are the a th attribute values belonging to the i th and j th record, w_a is a user defined weight (significance level) for attribute a , n is the number of numerical attributes (say, first m_1 attributes are numerical), m is the total number of attributes (both numerical and categorical), and $\text{dist}(R_{i,a}, R_{j,a})$ is the similarity based distance (see SiDCAV) between records R_i and R_j for categorical attribute values of the a th attribute. According to the weighted distance function (WeDiF), the distance between two records $\text{dist}(R_i, R_j)$ can vary between zero and one. Besides, $|R_{i,a} - R_{j,a}|$ is the difference between the normalized values of the a th numerical attribute of records R_i and R_j . The novel weighted function was first introduced in a previous study (Rahman and Islam 2011), but it is used for the first time in the clustering techniques and experiments of this study.

ISS: Initial Seed Selection

Unlike many existing techniques (Ahmad and Dey 2007a, Lee and Pedrycz 2009, Chatzis 2011), our proposed technique detects initial seeds using a deterministic process based on the density of the records in order to ensure a high quality of the initial seeds. We first calculate the number of records (density) within a user defined radius r of each record of a data set. That is, if there are N number of records in the data set within r distance (calculated using Equation 4 considering the user defined weight distribution of the attributes) of a record

R_i then the density of R_i is N . We choose the record having the highest density as the first seed of the data set, provided the density of the first seed is greater than or equal to a user defined threshold T . We then remove all the records (including the first seed itself) that are within the r distance of the first seed while calculating the density of the remaining records of the data set. The record currently having the highest density is then picked as the second seed of the data set, if the density of the second seed is greater than or equal to T . We continue the process of seed selection while we find a seed having density greater than or equal to T .

Algorithm: Initial Seed Selection

Method 1: InitialSeed ()

Input: A dataset D , a user defined radius r , a user defined minimum number of records T , user defined attribute-weight-distribution W for all attributes.

Output: A set of Initial seeds S

```

/* Set initially the "set of initial seeds" to null */
Set S ← ∅
/* density of each record will be stored in the density vector q. Set initially the density vector to null */
Set q ← ∅
/* index of the record having the maximum density will be stored in max_density_rec variable. Set initially the max_density to null */
Set max_density_rec ← ∅
/* the set of records within r distance of the max_density_rec will be stored in D_r. */
Set D_r ← ∅
/* the loop will continue while the remaining records of a data set is greater than or equal to T. */
WHILE |D| ≥ T DO
    q ← Density(D, r, W) /* call Density (D, r, W) */
    /* the record having maximum density is returned by Index_max(q) */
    max_density_rec ← Index_max(q)
    /* if the maximum density max(q) is greater than or equal to T then the Index_max(q) record is considered to be an initial seed. */
    IF max(q) ≥ T
        S ← S ∪ max_density_rec
        /* Find_records (max_density_rec, r, W) returns the set of records that are within r distance of max_density_rec record */
        D_r ← Find_records (max_density_rec, r, W)
        D ← D - D_r
    ENDIF
    ELSE
        Break;
    END ELSE
ENDWHILE
Return S.

```

Figure 2: Algorithm for Initial Seed Selection

In our novel fuzzy clustering approach (CRUDAW) we then calculate initial fuzzy membership degree of each record of the data set from the initial seeds. A similar approach for initial seed selection was also taken by an existing technique (Andreopoulos 2006, Andreopoulos et al. 2007) that clusters records of a data set having

categorical attributes. Our algorithm for initial seed selection is shown in the Figure 2 and Figure 3.

Algorithm: Initial Seed Selection
Method 2: Density()
Input: A data set D , a user defined radius r , user defined attribute-weight-distribution W
Output: A density vector q

```

Set distance  $d \leftarrow 0$ ,  $q \leftarrow \emptyset$ 
FOR all records  $R_i \in D$  DO
    /*  $c$  counts number of neighbor records of  $R_i$  within its  $r$  distances */
    Set  $c \leftarrow 0$ 
    FOR all records  $R_j \in D$  DO
         $d \leftarrow \text{distance}(R_i, R_j, W)$  /* call weighted distance function (WeDiF) */
        IF  $d \leq r$ 
             $c++$ ;
        END IF
    END FOR
     $q \leftarrow q \cup c$ 
ENDFOR
Return  $q$ ;
    
```

Figure 3: Algorithm for Density calculation

FuMeD: Fuzzy Membership Degree

For our proposed Fuzzy clustering technique CRUDAW, we calculate the membership degree $\mu_{i,j}$ ($0 \leq \mu_{i,j} \leq 1$) of the i th record R_i with the j th cluster seed S_j , $\forall i, j$ using the algorithm shown in Figure 4 following an existing membership degree calculation approach (Tang et al. 2010).

Algorithm: Fuzzy membership degree
Method: FuMeD()
Input: A dataset D , the set of seeds S , user defined attribute-weight-distribution W , a user defined fuzzy coefficient β .
Output: Fuzzy membership degree μ having size $|D| \times |S|$

```

Set  $\mu \leftarrow \emptyset$ 
FOR all records  $R_i \in D$  DO
    FOR all seeds  $S_j \in S$  DO
         $\mu_{i,j} = \frac{1}{\sum_{l=1}^{|S|} \left( \frac{\text{dist}(R_i, S_j)}{\text{dist}(R_i, S_l)} \right)^{\frac{2}{\beta-1}}}$  /* the  $\text{dist}(R_i, S_j)$  is calculated by using WeDiF using  $W$  */
    END FOR
     $\mu \leftarrow \mu \cup \mu_{i,j}$ 
END FOR
Return  $\mu$ ;
    
```

Figure 4: Algorithm for Fuzzy membership degree

However, note that the distance between record R_i and seed S_j i.e. $\text{dist}(R_i, S_j)$ is calculated using our novel function for distance measure called WeDiF. A seed is

considered to be structurally similar to a record in the sense that a seed has as many attributes as the number of attributes of a record.

SeCaF: Seed Calculation for Fuzzy Technique

Following traditional fuzzy C-means algorithms (Tang et al. 2010, Lee and Pedrycz 2009), we calculate the seed value $S_{j,a}$ of the j th cluster for the a th numerical attribute as follows.

$$S_{j,a} = \frac{\sum_{i=1}^n \mu_{i,j}^\beta R_{i,a}}{\sum_{i=1}^n \mu_{i,j}^\beta} \quad (5)$$

Here, n is the total number of records in a data set. Note that we use normalized records while calculating the seed values for a numerical attribute.

Algorithm: Seed calculation for CRUDAW
Method: SeCaF()
Input: A dataset D having altogether $|A|$ number of attributes and $|D|$ records, a set of fuzzy membership degrees μ , a user defined fuzzy coefficient β , number of clusters $|S|$.
Output: A set of seeds S having size $|S| \times |A|$

```

Set  $S \leftarrow \{S_1, S_2, \dots, S_{|S|}\}$  /*  $S_j$  is the  $j$ th seed. Initially  $S_j$  is a null set. */
FOR all  $S_j \in S$  DO
    FOR all attributes  $A_m \in A$  DO
        IF  $A_m$  is categorical
            /* the summation of fuzzy membership degree for each value  $v$  of the attribute  $A_m$  will be stored in  $M$  */
            Set  $M \leftarrow 0$ 
            FOR all domain values  $v \in A_m$ 
                IF  $M \leq \sum_{i=1}^{|D|} \mu_{i,j}^\beta |R_{i,m} = v|$  DO
                     $M \leftarrow \sum_{i=1}^{|D|} \mu_{i,j}^\beta |R_{i,m} = v|$ 
                     $S_{j,m} \leftarrow v$  /*  $S_{j,m}$  is the  $m$ th attribute value of the  $j$ th seed */
                END IF
            END FOR
        ELSE /* if attribute  $A_m$  is numerical */
             $S_{j,m} \leftarrow \frac{\sum_{i=1}^{|D|} \mu_{i,j}^\beta R_{i,m}}{\sum_{i=1}^{|D|} \mu_{i,j}^\beta}$  /*  $\mu_{i,j}$  is the fuzzy membership degree of  $i$ th record with  $j$ th cluster and  $R_{i,m}$  is the  $m$ th attribute value of  $i$ th record */
        END ELSE
         $S_j \leftarrow S_j \cup S_{j,m}$ 
    END FOR
END FOR
Return  $S$ ;
    
```

Figure 5: Algorithm for Seed calculation in CRUDAW

However, following the approach taken by another existing fuzzy clustering technique (Kim et al. 2004), we calculate the seed value of a categorical attribute b as follows. Let, the domain values of a categorical attribute b are $\{b_1, b_2, \dots, b_r\}$, where r is the domain size for the attribute. The seed value of the attribute for the j th cluster, $S_{j,b} = b_p$ when the Equation 6 is satisfied.

$$\sum_{i=1}^n R_{i,b=b_p} \mu_{i,j}^\beta \geq \sum_{i=1}^n R_{i,b=b_q} \mu_{i,j}^\beta; \forall q \neq p \quad (6)$$

Here, $1 \leq p \leq r$ and $1 \leq q \leq r$. That is, if the summation of the membership degrees (with the j th cluster) of all records having the value b_p (for the categorical attribute b) is greater than the summation of the membership degrees of all records having any other value (for all other values) then the seed value for the attribute b is equal to b_p for the j th cluster. The algorithm for seed calculation is shown in Figure 5.

TCFCM: Termination Conditions for Fuzzy Clustering Method

For CRUDAW, we use a weighted fuzzy objective function J_λ for the λ th iteration as follows.

$$J_\lambda = \sum_{i=1}^n \sum_{j=1}^{|S|} \mu_{i,j}^\beta * \text{dist}(R_i, S_j) \quad (7)$$

If $|J_\lambda - J_{\lambda-1}| \leq \epsilon$ or a user defined number of iterations η is completed then the clustering iteration terminates; otherwise it continues. $J_{\lambda-1}$ and J_λ are the objective function values in two consecutive iterations. Note that unlike the existing techniques we calculate $\text{dist}(R_i, S_j)$ using our WeDiF function.

FCFMD: Final Clustering based on Fuzzy Membership Degrees

Algorithm: CRUDAW Algorithm

Input: A dataset D , a user defined radius r , a user defined minimum number of records T , user defined attribute-weight-distribution W , a user defined fuzzy coefficient β , a user defined objective function threshold ϵ , a user defined maximum number of iteration η

Output: A set of rigid clusters C , and a set of membership degree μ

Method:

```

Step 1: Normalize the data set D using NoNAV function
        D ← Normalize (D) /* call NoNAV function*/
Step 2: Initial Seed Selection
        S ← InitialSeed(D, r, T, W) /* call ISS function*/
Set  $J_{cur} \leftarrow 0, J_{prev} \leftarrow 0$ 
FOR ( $\lambda = 1$  to  $\eta$ ) DO /*  $\lambda$  counts the number of iteration*/
    Step 3: Fuzzy membership degree
         $\mu \leftarrow \text{FuMeD}(D, S, W, \beta)$  /* call FMD function*/
    Step 4: Seed calculation
        S ← SeCaF(D,  $\mu, \beta, |S|$ ) /* call SCF function*/
    Step 5: Termination conditions for CRUDAW
         $J_{cur} \leftarrow \text{TCFCM}(D, S, W, \mu, \beta)$  /*call TCFCM function*/
        IF  $\lambda > 1$  && ( $|J_{cur} - J_{prev}| \leq \epsilon$ ) DO
            Break; /* terminate clustering as it meets the termination condition*/
        END IF
         $J_{prev} \leftarrow J_{cur}$ 
END FOR
Step 6: Produce the final clusters based on fuzzy membership degree
        C ← FCFMD ( $\mu, D, S$ )
Return C,  $\mu$ .

```

Figure 6: CRUDAW Algorithm

CRUDAW finally produces two outputs; the first output is a set of fuzzy membership degrees of each record with all cluster centers (seeds) and the second output is the rigid clustering where each record is assigned to the cluster for which it has the highest membership degree. This way a record is associated with only one cluster. It also returns the rigid clustering since a user often may need it for a number of purposes. We now present the algorithm (Figure 6) and block diagram (Figure 7) for CRUDAW integrating various components introduced above.

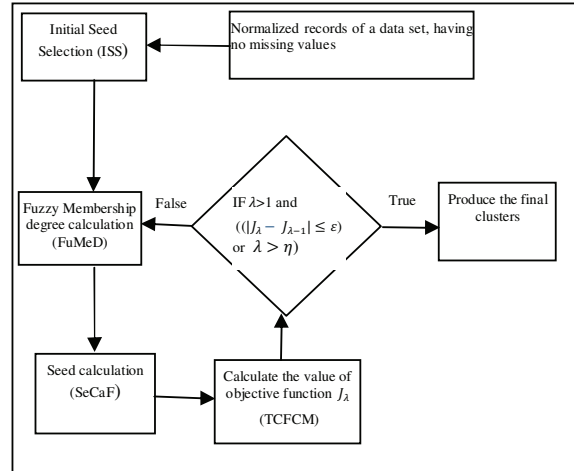


Figure 7: Block Diagram of CRUDAW

4 Experimental Results

We implement our technique CRUDAW and a few existing techniques namely SABC (Ahmad and Dey 2007a), GFCM (Lee and Pedrycz 2009), and KL-FCM-GM (Chatzis 2011). We use a few evaluation criteria, specifically Silhouette Coefficient, F-measure (with $\delta = 1$), Entropy, and Purity (Tan et al. 2005, Chuang 2004, Kashef and Kamel 2009, Rahman and Islam 2011) to compare the performance of the technique. We also use t-test (Johnson and Bhattacharyya 1985, Moore 1995) and confidence interval test (Johnson and Bhattacharyya 1985, Moore 1995, Triola 2001) to estimate the statistical significance of the performance of the technique.

Data set	Records (size)	Categorical attributes (cat)	Numerical attributes (num)	Missing values	Classification Accuracy	Class size
Mushroom	8124	22	0	yes	95%	2
Credit Approval	690	9	6	yes	80%	2
Pima Indian Diabetes	768	0	8	no	72%	2
Contraceptive Method Choice (CMC)	1473	7	2	no	52.4%	3

Table 2: Information on data sets at a glance

We use four natural data sets namely Mushroom, Credit Approval, Pima Indian Diabetes, and Contraceptive Method Choice (CMC) – all of them are available from UCI machine learning repository (UCI

2012). Brief information on the data sets is presented in Table 2.

The Mushroom data set and Credit Approval data set have some missing values. We first remove all records having any missing values. After removing the records having missing values Mushroom, Credit Approval data sets have 5644 and 653 records, respectively. We also remove the class attributes from the data sets before we apply clustering techniques on them. The class attributes are used again for the cluster evaluation based on F-measure, Purity and Entropy.

In all experiments on CRUDAW, we use fuzzy coefficient $\beta=2.2$, fuzzy termination condition $\varepsilon = 0.005$, and for initial seed selection $T=1\%$ of the records of a data set. However, in all experiments on GFCM we use fuzzy coefficient $\beta = 1.3$ and fuzzy termination condition $\varepsilon = 0.0001$ following the recommendation of the original paper (Lee and Pedrycz 2009) in order to achieve the best result from the technique. Similarly, for the experiments on KL-FCM-GM we use degree of fuzziness $=1.5$ and fuzzy termination condition $\varepsilon = 0.005$ as recommended for obtaining the best result from the technique (Chatzis 2011). The maximum number of iterations for CRUDAW, SABC, GFCM and KL-FCM-GM are considered to be 50.

For CRUDAW, a user can assign weights on the attributes according to his/her requirement. A user assigns higher weights on the attributes that he/she considers to be more important for clustering the records, as discussed in Section 3.1. The evaluation criteria (used in this study) such as F-measure, Purity and Entropy focus on the ability of a clustering technique to group the records in homogeneous collections where in each collection all records have the same class value. Therefore, in order to match the focus of the evaluation criteria we (in this experiment) consider that a user assigns high weights on the attributes that are strongly related to the class attribute – i.e. high weights on the attributes having low entropy with respect to the class values (Quinlan 1993, Quinlan 1996, Islam 2012).

We argue that if CRUDAW can achieve good F-measure, Purity and Entropy values under such weight distribution then they should also achieve good clustering results (according to the purposes of the users) when the users assign a different weight distribution following their purposes.

Based on entropy values of the attributes of a data set we divide them into three categories namely the best attributes (BA) consisting of the attributes having low entropies, medium attributes (MA), and the worst attributes (WA). If the number of attributes of a data set is divisible by three then each category contains one third of the total number of attributes. Otherwise, the best and the worst categories have the same number of attributes while the medium category contains more attributes. In order to simulate different user attitudes we use different weight patterns to assign high weights on the best attributes (BA) and a combination of attributes from the best and medium categories (BM).

We now explain the weight patterns for the best attributes (BA), and the best and medium attributes (BM)

using an example on CMC data set (see Table 3). Using the entropy of the attributes, we rank the attributes where attributes having low entropy (i.e. good attributes) are ranked low. Various weight patterns such as BA1, BA2, and BM1 are shown in Table 3.

Weights on best attributes (BA)										
Attribute	A1	A2	A3	A4	A5	A6	A7	A8	A9	Notations
Rank	2	1	3	8	7	9	5	4	6	
Best attributes (BA)	0.2	0.2	0.2	0	0	0	0	0	0	BA1
	0.2	0.4	0.2	0	0	0	0	0	0	BA2
	0.4	0.6	0.2	0	0	0	0	0	0	BA3
	0.6	0.8	0.4	0	0	0	0	0	0	BA4
	0.8	1	0.6	0	0	0	0	0	0	BA5
Weights on best and medium attributes (BM)										
Best and Medium attributes (BM)	0.2	0.2	0	0	0	0	0	0.2	0	BM1
	0.2	0.4	0	0	0	0	0	0.4	0	BM2
	0.4	0.6	0	0	0	0	0	0.6	0	BM3
	0.6	0.8	0	0	0	0	0	0.8	0	BM4
	0.8	1.0	0	0	0	0	0	1	0	BM5

Table 3: Weights pattern for CMC data set

In the experiments of CRUDAW, we use three different r values for each data set in order to test the technique on different numbers of seeds or clusters. For comparing CRUDAW with the existing techniques we produce the same numbers of clusters for all techniques.

However, SABC, GFCM, and KL-FCM-GM do not produce initial seeds deterministically and therefore, produce different sets of clusters in different runs. That is if we run SABC twice to produce say 7 clusters we may get different sets of 7 clusters. Hence, we run each of these existing techniques ten times for each number of clusters. We then calculate the average Silhouette coefficients, F-measures, Entropy, and Purity for the ten runs.

Silhouette Coefficient						
Weights	r	k	CRUDAW	SABC (avg: 10 exp)	GFCM (avg: 10 exp)	KL-FCM-GM (avg: 10 exp)
BA1	0.05	6	0.5338[4]	0.5092[3]	0.3889[2]	0.0717[1]
	0.02	20	0.7777[4]	0.2702[3]	0.142[2]	0.0649[1]
	0.01	26	0.8536[4]	0.2144[3]	0.0877[2]	0.0425[1]
BA2	0.05	7	0.7609[4]	0.5028[3]	0.3831[2]	0.0983[1]
	0.02	17	0.6992[4]	0.2678[3]	0.1416[2]	0.078[1]
	0.01	26	0.8504[4]	0.214[3]	0.0877[2]	0.0413[1]
BA3	0.05	8	0.8409[4]	0.387[3]	0.2988[2]	0.2043[1]
	0.02	14	0.5612[4]	0.3241[3]	0.1756[2]	0.068[1]
	0.01	26	0.5612[4]	0.2165[3]	0.0868[2]	0.0404[1]
BA4	0.05	8	0.8771[4]	0.4031[3]	0.2975[2]	0.2068[1]
	0.02	12	0.6126[4]	0.4717[3]	0.1661[2]	0.0802[1]
	0.01	22	0.7243[4]	0.2819[3]	0.1109[2]	0.0579[1]
BA5	0.05	8	0.8872[4]	0.4036[3]	0.2964[2]	0.2101[1]
	0.02	11	0.7439[4]	0.4297[3]	0.2322[2]	0.0938[1]
	0.01	16	0.6335[4]	0.43[3]	0.1485[2]	0.0854[1]
Total Score			60	45	30	15

Table 4: Silhouette Coefficient based on best attributes (BA) of Mushroom data set

In Table 4, we present silhouette coefficients of CRUDAW for the weight patterns on the best attributes (BA) of Mushroom data set. We also present the average silhouette coefficients of SABC, GFCM, and KL-FCM-GM from *ten runs* of each technique for each number of clusters i.e. for each k value in Table 4. For weight pattern BA1 and $r = 0.05$, we get six initial seeds and therefore, six clusters ($k=6$) for CRUDAW. The

Silhouette coefficient for the six clusters of CRUDAW is 0.5338. For same number of clusters the average (of all ten runs) Silhouette coefficients of SABC, GFCM and KL-FCM-GM are 0.5092; and 0.3889 and 0.0717, respectively. Similarly, we also estimate the F-measure, Entropy and Purity (Table 5, Table 6 and Table 7).

F-measure						
Weights	r	k	CRUDAW	SABC (avg: 10 exp)	GFCM (avg: 10 exp)	KL-FCM-GM (avg: 10 exp)
BA1	0.05	6	0.9025[4]	0.8682[2]	0.83[3]	0.5252[1]
	0.02	20	0.9971[4]	0.7929[2]	0.8322[3]	0.6875[1]
	0.01	26	0.9971[4]	0.744[2]	0.8341[3]	0.7363[1]
BA2	0.05	7	0.9781[4]	0.8453[3]	0.8336[2]	0.5822[1]
	0.02	17	0.9971[4]	0.7728[2]	0.832[3]	0.6467[1]
	0.01	26	0.9971[4]	0.744[2]	0.8341[3]	0.7363[1]
BA3	0.05	8	0.9971[4]	0.843[3]	0.8356[2]	0.8252[1]
	0.02	14	0.9865[4]	0.8392[3]	0.8335[2]	0.6383[1]
	0.01	26	0.9908[4]	0.744[2]	0.8341[3]	0.7363[1]
BA4	0.05	8	0.9971[4]	0.843[3]	0.8356[2]	0.8252[1]
	0.02	12	0.9971[4]	0.8733[3]	0.8318[2]	0.6489[1]
	0.01	22	0.99[4]	0.7865[2]	0.8322[3]	0.6556[1]
BA5	0.05	8	0.9971[4]	0.843[2]	0.8356[3]	0.8252[1]
	0.02	11	0.9971[4]	0.8783[3]	0.8335[2]	0.6026[1]
	0.01	16	0.9971[4]	0.9036[3]	0.8365[2]	0.6934[1]
Total Score			60	37	38	15

Table 5: F-measure based on best attributes (BA) of Mushroom data set

Entropy						
Weights	r	k	CRUDAW	SABC (avg: 10 exp)	GFCM (avg: 10 exp)	KL-FCM-GM (avg: 10 exp)
BA1	0.05	6	0.3344[4]	0.3821[3]	0.535[2]	0.8536[1]
	0.02	20	0.0183[4]	0.3607[3]	0.5235[2]	0.6867[1]
	0.01	26	0.0183[4]	0.416[3]	0.5001[2]	0.637[1]
BA2	0.05	7	0.132[4]	0.3886[3]	0.5138[2]	0.7987[1]
	0.02	17	0.0197[4]	0.4124[3]	0.5124[2]	0.7075[1]
	0.01	26	0.0183[4]	0.416[3]	0.5001[2]	0.637[1]
BA3	0.05	8	0.024[4]	0.3653[3]	0.4935[2]	0.559[1]
	0.02	14	0.0526[4]	0.3098[3]	0.5173[2]	0.7415[1]
	0.01	26	0.0437[4]	0.416[3]	0.5001[2]	0.637[1]
BA4	0.05	8	0.0247[4]	0.3653[3]	0.4935[2]	0.559[1]
	0.02	12	0.0218[4]	0.2622[3]	0.5256[2]	0.7079[1]
	0.01	22	0.0437[4]	0.3756[3]	0.5012[2]	0.7219[1]
BA5	0.05	8	0.0247[4]	0.3653[3]	0.4935[2]	0.559[1]
	0.02	11	0.0218[4]	0.2793[3]	0.5201[2]	0.7699[1]
	0.01	16	0.0197[4]	0.2032[3]	0.5121[2]	0.6679[1]
Total Score			60	45	30	15

Table 6: Entropy based on best attributes (BA) of Mushroom data set

We now compare the techniques through their scores (as shown within the brackets/parentheses) based on a scoring rule where we assign 4, 3, 2, and 1 point for the techniques having the best, the 2nd best, the 3rd best, and the worst result, respectively. For each evaluation criteria, the Total Scores of CRUDAW are significantly better than the scores of any other techniques for the Mushroom data set. See Table 4, Table 5, Table 6 and Table 7 for more information. Note that all distances are calculated using Equation 4 following the weights assigned in a weight pattern.

Finally, in Table 8 and Table 9 we show the total scores of the techniques for BA and BM weight patterns for all evaluation criteria. The total score (as shown in the last row of Table 8 and Table 9) of each technique is also presented in Figure 8, which shows a clear domination of CRUDAW over all other techniques for all evaluation criteria. Similarly, from Figure 9 to Figure 11, we present

total scores of the techniques for Credit Approval (CA), Pima Indian Diabetes (PID), and Contraceptive Method Choice (CMC) data sets.

Purity						
Weights	r	k	CRUDAW	SABC	GFCM	KL-FCM-GM
BA1	0.05	6	0.8986[4]	0.8862[3]	0.8532[2]	0.6723[1]
	0.02	20	0.9971[4]	0.8647[3]	0.8557[2]	0.7746[1]
	0.01	26	0.9971[4]	0.8389[3]	0.8573[2]	0.8026[1]
BA2	0.05	7	0.978[4]	0.8775[3]	0.8565[2]	0.7086[1]
	0.02	17	0.9971[4]	0.8464[2]	0.8551[3]	0.7555[1]
	0.01	26	0.9971[4]	0.8389[2]	0.8573[3]	0.8026[1]
BA3	0.05	8	0.9971[4]	0.8763[3]	0.858[2]	0.8493[1]
	0.02	14	0.9865[4]	0.8881[3]	0.855[2]	0.746[1]
	0.01	26	0.9907[4]	0.8389[2]	0.8573[3]	0.8026[1]
BA4	0.05	8	0.9971[4]	0.8763[2]	0.858[3]	0.8493[1]
	0.02	12	0.9971[4]	0.9051[3]	0.8551[2]	0.7581[1]
	0.01	22	0.9907[4]	0.8587[3]	0.8553[2]	0.7556[1]
BA5	0.05	8	0.9971[4]	0.8763[3]	0.858[2]	0.8493[1]
	0.02	11	0.9971[4]	0.906[3]	0.8561[2]	0.7318[1]
	0.01	16	0.9971[4]	0.9288[3]	0.8585[2]	0.7812[1]
Total Score			60	45	30	15

Table 7: Purity based on best attributes (BA) of Mushroom data

Dataset: Mush Room ; records=5644; cat=21								
Weight	Score based on Silhouette coefficient				Score based on F-measure			
	CRUDAW	SABC	GFCM	KL-FCM-GM	CRUDAW	SABC	GFCM	KL-FCM-GM
BA	60	45	30	15	60	37	38	15
BM	59	46	30	15	60	40	35	15
Total	119	91	60	30	120	77	73	30

Table 8: Score comparison based on Mushroom data set

Dataset: Mush Room ; records=5644; cat=21								
Weight	Score based on Entropy				Score based on Purity			
	CRUDAW	SABC	GFCM	KL-FCM-GM	CRUDAW	SABC	GFCM	KL-FCM-GM
BA	60	45	30	15	60	41	34	15
BM	59	46	30	15	60	42	33	15
Total	119	91	60	30	120	83	67	30

Table 9: Score comparison based on Mushroom data set

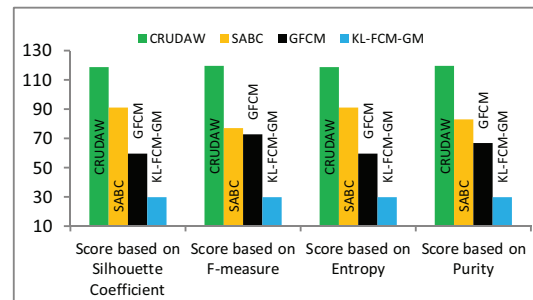


Figure 8: Score comparison based on Mushroom data set

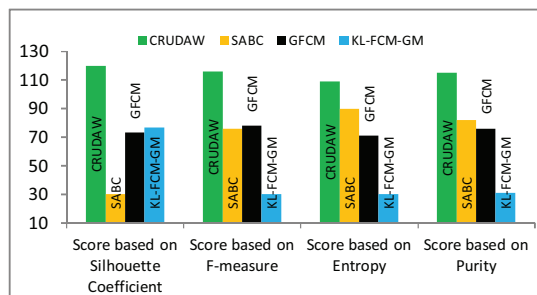


Figure 9: Score comparison based on Credit Approval data set

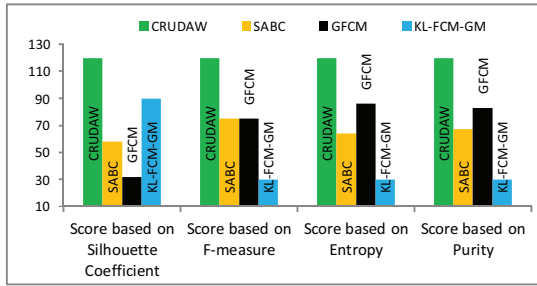


Figure 10: Score comparison based on Pima Indian Diabetes data set

Overall, CRUDAW performs clearly better than the other techniques. In Mushroom, Credit Approval, Pima Indian Diabetes, and Contraceptive Method (CMC) our technique score higher than all other techniques for all evaluation criteria as presented in the figures below (Figure 12 to Figure 15)

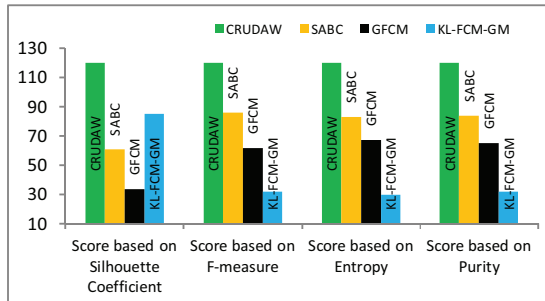


Figure 11: Score comparison based on CMC data set

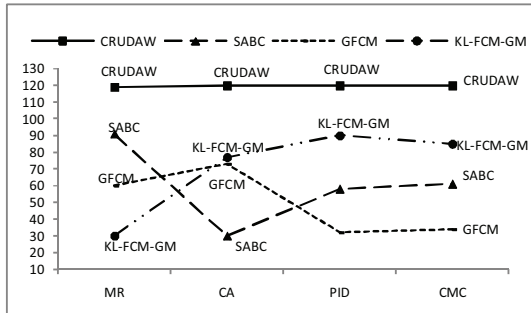


Figure 12: Score comparison for Silhouette Coefficient on all datasets

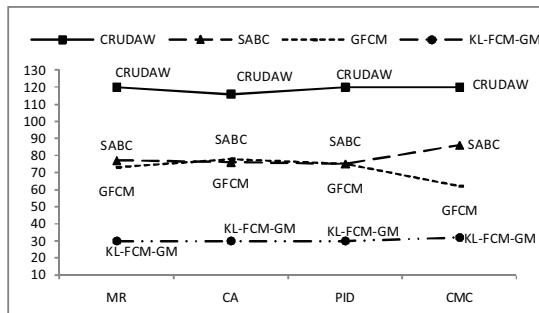


Figure 13: Score comparison for F-measure on all datasets

We now use statistical t-test (Johnson and Bhattacharyya 1985, Moore 1995) in order to explore whether the results of various evaluation criteria for our technique are significantly higher than the results for the existing techniques. In the t-tests, we considered $p = 0.05$ (i.e. 95% significance level) and degrees of freedom (df) = 58. For $p = 0.05$ and $df = 58$ the t-ref value is 1.644 which we call “t-ref” (reference t-value).

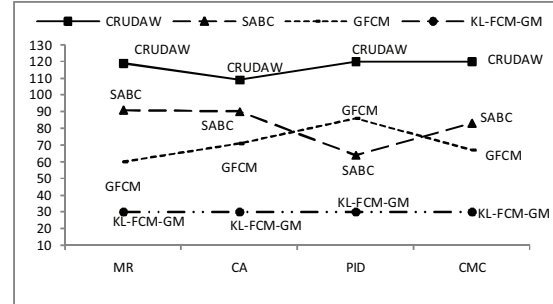


Figure 14: Score comparison for Entropy on all datasets

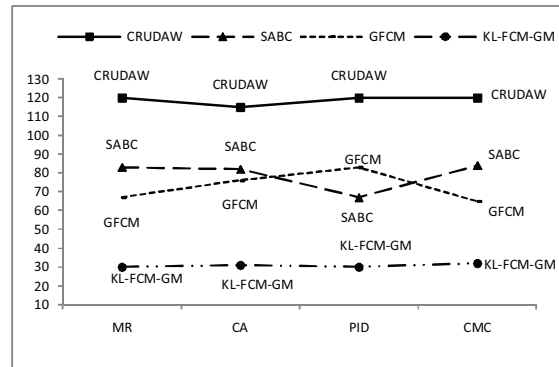


Figure 15: Score comparison for Purity on all datasets

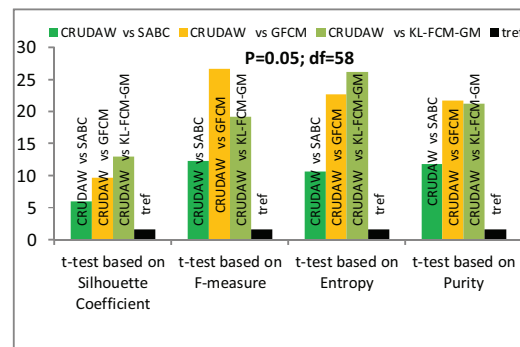


Figure 16: t-test for CRUDAW on Mushroom data set

In the figures (Figure 16 to Figure 18), we present t-test results of CRUDAW compared with other techniques on the Mushroom, Credit Approval, and Pima Indian Diabetes data sets. In Figure 16, the first bar from the left side (“CRUDAW vs SABC”) is taller than the t-ref bar meaning that the actual Silhouette coefficient values (not the score) of CRUDAW are significantly better than the Silhouette coefficient values of SABC technique at 95%

significance level. The t-test results for CMC data set is presented in tabular form (see Table 10). We experience difficulties in presenting them in graphical form due to huge differences between the values.

We also carry out the Confidence Interval analysis (Johnson and Bhattacharyya 1985, Moore 1995, Triola 2001) at 90% confidence level for all data sets. The confidence intervals for actual silhouette coefficient for Mushroom (MR), Credit Approval (CA), Pima Indian Diabetes (PID), and Contraceptive Method Choice (CMC) are presented in Figure 19. Similarly, Figure 20 presents the confidence intervals for F-measure for the data sets

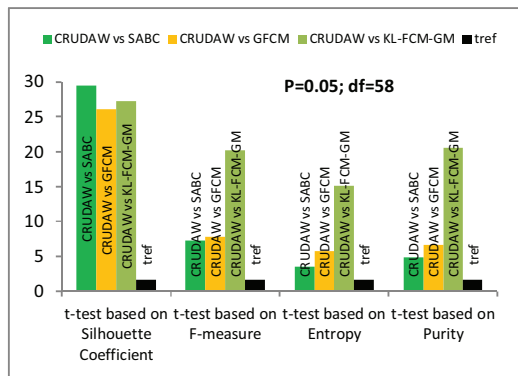


Figure 17: t-test for CRUDAW on Credit Approval data set

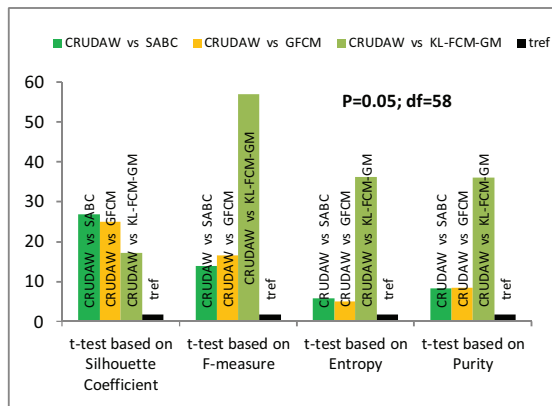


Figure 18: t-test for CRUDAW on Pima Indian Diabetes data set

Data set : CMC ; records:1473; c=7; n=2											
P=0.05; tref=1.644; df=58; T1 = CRUDAW ; KF = KL-FCM-GM											
t-value based on Silhouette coefficient			t-value based on F-measure			t-value based on Entropy			t-value based on Purity		
T1	T1	T1	T1	T1	T1	T1	T1	T1	T1	T1	T1
vs	vs	vs	vs	vs	vs	vs	vs	vs	vs	vs	vs
SABC	GFCM	KF	SABC	GFCM	KF	SABC	GFCM	KF	SABC	GFCM	KF
22.63	40.90	42.35	14.33	40.44	90.18	6.62	8.21	16.04	8.51	12.86	13.74

Table 10: t-test on CMC data set

According to Figure 19 and Figure 20, the average values of Silhouette coefficient (the most natural evaluation criterion) and F-measure (a combination of precision and recall) for CRUDAW are clearly better than other techniques for all data sets. Moreover, there is no

overlap of the confidence intervals of CRUDAW with the intervals of other techniques. This is the case for other two evaluation criteria as well. We use results of 30 experiments for confidence interval calculation; 15 from BA categories and 15 from BM categories. However, each of the 30 results is the average value of 10 runs as explained before (see Table 3 to Table 7).

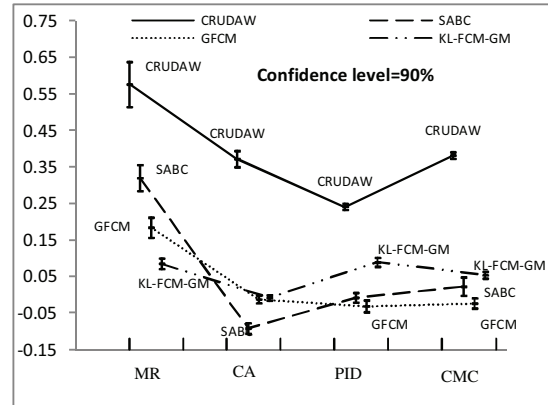


Figure 19: Confidence Interval based on Silhouette coefficient

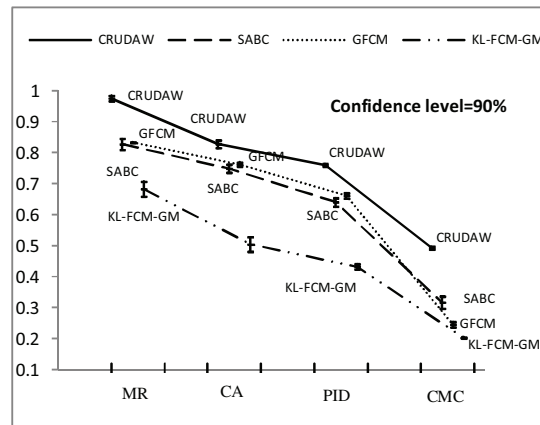


Figure 20: Confidence Interval based on F-measure

Overall average computational time (seconds) of the techniques for all data				
Data sets	CRUDAW	KL-FCM-GM	SABC	GFCM
Credit Approval	2.05	127.8125	2.235	0.1465
Pima Indian Diabetes	2.0995	38.806	1.169	1.152
Contraceptive Method Choice	4.7635	350.874	2.136	0.119
Mushroom	213.32	59.53	45.812	1.6775

Table 11: Overall average computational time (in seconds) of the techniques for all data set

We also calculate the overall time required for clustering by the techniques (Table 11). For experiments on time complexity analysis we use a shared computer system the configuration of which is 4x8 core Intel E7-8837 Xeon processors, 256 GB of RAM, and 23 TB of disk storage. Generally KL-FCM-GM technique requires the maximum amount of time in our experiments. Perhaps due to random selection of initial seeds SABC and GFCM require less computation time than

CRUDAW at the cost of relatively inferior quality of clusters. Finally Table 12 presents a comparison between the time complexity of CRUDAW and an existing technique called Seed-Detective.

Data set	Seed-Detective Execution Time (sec.)	CRUDAW Time (sec.)
CMC	35.01	4.76
CA	17.62	2.05

Table 12: Overall average computational time (in seconds)

5 Conclusion

In this study we present a novel clustering technique called CRUDAW. Our proposed technique (CRUDAW) allows a data miner to assign weights on the attributes of a data set based on their importance (to the data miner) for clustering. The technique uses a novel approach to select initial seeds deterministically using the density of the records of a data set. CRUDAW selects the initial fuzzy membership degrees deterministically. CRUDAW also uses a novel approach for measuring distance considering the user defined weights of the attributes. Moreover, while measuring the distance between the values of a categorical attribute the technique takes the similarity of the values into consideration. We also present complete algorithms for the technique.

We experimentally compare our technique with a few existing techniques namely SABC, GFCM, KL-FCM-GM based on various evaluation criteria called Silhouette coefficient, F-measure, purity and entropy. The experimental results strongly indicate the supremacy of our novel technique over the existing techniques. For all data sets used in this study, our technique scores higher than all other techniques for all evaluation criteria.

We carry out statistical t-tests to ensure the significance of the better result of our technique. We then also perform confidence interval tests at 90% confidence level. Both tests confirm the statistical significance of the superior results achieved by CRUDAW.

We also record the time complexity (during execution) of the technique. CRUDAW performs better than KL-FCM-GM and Seed-Detective. However, SABC and GFCM require less computation time than CRUDAW, perhaps due to their random seed selection approach, at the cost of relatively inferior quality of clusters. Hence, for non-time critical applications requiring good quality clusters, we believe CRUDAW is more suitable than the existing techniques tested in this study.

Our future research goals include a further improvement of the technique, reduction of time complexity, and automatic generation of attribute weights as a suggestion for a user.

6 References

- Ahmad, A. and Dey, L. (2007a): A k-mean clustering algorithm for mixed numeric and categorical data. *Data & Knowledge Engineering*, 63(2), 503-527. doi: 10.1016/j.datak.2007.03.016
- Ahmad, A. and Dey, L. (2007b): A method to compute distance between two categorical values of same attribute in unsupervised learning for categorical data set. *Pattern Recognition Letters*, 28(1), 110-118. doi: 10.1016/j.patrec.2006.06.006
- Alata, M., Molhim, M. and Ramini, A. (2008): Optimizing of Fuzzy C-Means Clustering Algorithm Using GA *World Academy of Science, Engineering and Technology* (Vol. 39, pp. 224-229).
- Andreopoulos, B., An, A. and Wang, X. (2007): Hierarchical Density-Based Clustering of Categorical Data and a Simplification. *Proc. 11th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2007)*, Nanjing, China.
- Andreopoulos, W. (2006): Clustering Algorithms for Categorical Data. Ph.D. thesis. York University, Toronto, Ontario.
- Bai, L., Liang, J. and Dang, C. (2011): An initialization method to simultaneously find initial cluster centers and the number of clusters for clustering categorical data. *Knowledge-Based Systems*, 24(6), 785-795. doi: 10.1016/j.knsys.2011.02.015
- Bezdek, J. J. (1981): *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum.
- Chatzis, S. P. (2011): A fuzzy c-means-type algorithm for clustering of data with mixed numeric and categorical attributes employing a probabilistic dissimilarity functional. *Expert Systems with Applications*, 38(7), 8684-8689. doi: 10.1016/j.eswa.2011.01.074
- Chuang, K.-T. and Chen, M.-S. (2004): Clustering Categorical Data by Utilizing the Correlated-Force Ensemble. *Proc. 4th SIAM International Conference on Data Mining (SDM 04)*, Lake Buena Vista, Florida.
- Ganti, V., Gehrke, J. and Ramakrishnan, R. (1999): CACTUS-Clustering Categorical Data Using Summaries. *Proc. Fifth ACM SIGKDD international conference on Knowledge discovery and data mining* San Diego, CA, USA.
- Gath, I. and Geva, A. B. (1989): Unsupervised optimal fuzzy clustering. *Proc. IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7), 773-781.
- Giggins, H. P. (2009): Security of Genetic Databases. Ph.D. thesis. School of Electrical Engineering and Computer Science, The University of Newcastle, Australia.
- Grubestic, 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, USA.
- Guha, S., Rastogi, R. and Shim, K. (1998): CURE: an efficient clustering algorithm for large databases. *Proc. ACM SIGMOD international conference on Management of data (SIGMOD'98)*, New York, NY, USA
- Han, J. Kamber, M. (2006): *Data Mining Concepts and Techniques* (2nd ed.). San Francisco: Morgan Kaufmann.
- Hasan, M. A., Chaoji, V., Salem, S. and Zaki, M. J. (2009): Robust partitional clustering by outlier and density insensitive seeding. *Pattern Recognition*

- Letters*, 30(11), 994-1002. doi: 10.1016/j.patrec.2009.04.013
- Hathaway, R. J. and Bezdek, J. C. (1988): Recent Convergence Results for the Fuzzy c-Means Clustering Algorithms. *Journal of Classification*, 5(2), 237-247. doi: DOI: 10.1007/BF01897166
- Huang, D. and Pan, W. (2006): Incorporating biological knowledge into distance-based clustering analysis of microarray gene expression data. *Journal of Bioinformatics* 22(10), 1259-1268. doi: doi>10.1093/bioinformatics/btl065
- Huang, Z. (1997): Clustering large data sets with mixed numeric and categorical values. *Proc. First Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Singapore.
- Huang, Z. and Ng, M. K. (1999): A Fuzzy k-Modes Algorithm for Clustering Categorical Data. *IEEE Transactions on Fuzzy Systems*, 36(2), 1615-1620. doi: 10.1016/j.eswa.2007.11.045
- Islam, M. Z. (2012): EXPLORE: A Novel Decision Tree Classification Algorithm, *Data Security and Security Data*, LNCS, Vol. 6121, L.M. MacKinnon (Ed.), Springer, Berlin/Heidelberg, ISBN 978-3-642-25703-2, pg. 55-71.
- 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*. Vol. 24, Issue 8, ISBN 0950-7051, DOI: 10.1016/j.knosys.2011.05.011.
- 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, Australia.
- Islam, M. Z. (2008): Privacy Preservation in Data Mining through Noise Addition. Ph.D. Thesis. School of Electrical Engineering and Computer Science, The University of Newcastle, Australia.
- Ji, J., Pang, W., Zhou, C., Han, X. and Wang, Z. (2012): A fuzzy k-prototype clustering algorithm for mixed numeric and categorical data. *Knowledge-Based Systems*, 30(0), 129-135. doi: 10.1016/j.knosys.2012.01.006
- Johnson, R. and Bhattacharyya, G. (1985): *Statistics Principles and Methods*, Revised Printing, John Wiley and Sons.
- Kashef, R. and Kamel, M. S. (2009): Enhanced bisecting-means clustering using intermediate cooperation. *Pattern Recognition*, 42(11), 2557-2569. doi: 10.1016/j.patcog.2009.03.011
- Kim, D.-W., Lee, K. H. and Lee, D. (2004): Fuzzy clustering of categorical data using fuzzy centroids. *Pattern Recognition Letters*, 25(11), 1263-1271. doi: 10.1016/j.patrec.2004.04.004
- Lee, M. and Pedrycz, W. (2009): The fuzzy C-means algorithm with fuzzy P-mode prototypes for clustering objects having mixed features. *Fuzzy Sets and Systems*, 160(24), 3590-3600. doi: 10.1016/j.fss.2009.06.015
- Li, M. J., Ng, M. K., Cheung, Y.-m. and Huang, J. Z. (2008): Agglomerative Fuzzy K-Means Clustering Algorithm with Selection of Number of Clusters. *IEEE Transactions on Knowledge and Data Engineering*, 20(11), 1519-1534. doi: 10.1109/TKDE.2008.88
- Lung, C.-H., Zaman, M. and Nandi, A. (2004): Applications of clustering techniques to software partitioning, recovery and restructuring. *Journal of Systems and Software*, 73(2), 227-244. doi: 10.1016/s0164-1212(03)00234-6
- Masulli, F. and Schenone, A. (1999): A fuzzy clustering based segmentation system as support to diagnosis in medical imaging. *Artificial Intelligence in Medicine*, 16(2), 129-147. doi: 10.1016/s0933-3657(98)00069-4
- Moore, D. S. (1995): *The Basic Practice of Statistics*. W. H. Freeman and Company, New York.
- Quinlan, J. R. (1993): *C4.5: programs for machine learning*, San Francisco, CA, USA Morgan Kaufmann Publishers Inc.
- Quinlan, J. R. (1996): Improved use of continuous attributes in C4.5. *Journal of Artificial Intelligence Research*, 4(1), 77-90.
- Rahman, M. A. and Islam, M. Z. (2011): Seed-Detective: A Novel Clustering Technique Using High Quality Seed for K-Means on Categorical and Numerical Attributes. *Proc. Ninth Australasian Data Mining Conference: AusDM 2011*, vol. 121. University of Ballarat, Australia.
- Redmond, S. J. and Heneghan, C. (2007): A method for initialising the K-means clustering algorithm using kd-trees. *Pattern Recognition Letters*, 28(8), 965-973. doi: 10.1016/j.patrec.2007.01.001
- Saha, I., Maulik, U. and Plewczynski, D. (2011): A new multi-objective technique for differential fuzzy clustering. *Applied Soft Computing*, 11(2), 2765-2776. doi: 10.1016/j.asoc.2010.11.007
- Song, J. and Nicolae, D. L. (2009): A sequential clustering algorithm with applications to gene expression data. *Journal of the Korean Statistical Society*, 38(2), 175-184. doi: 10.1016/j.jkss.2008.09.006
- Tan, P.-N., Steinbach, M. and Kumar, V. (2005): *Introduction to Data Mining* (1st ed.): Pearson Addison Wesley.
- Tang, C., Wang, S. and Xu, W. (2010): New fuzzy c-means clustering model based on the data weighted approach. *Data & Knowledge Engineering*, 69(9), 881-900. doi: 10.1016/j.datak.2010.05.001
- Triola, M. F. (2001): *Elementary Statistics*, 8th ed. Addison Wesley Longman, Inc.
- Tsai, C. Y. and Chiu, C. C. (2004): A purchase-based market segmentation methodology. *Expert Systems with Applications*, 27(2), 265-276. doi: 10.1016/j.eswa.2004.02.005
- UCI (2012): UCI Machine Learning Repository. <http://archive.ics.uci.edu/ml/>. Accessed 11 July 2012
- Zamir, O. and Etzioni, O. (1999): Grouper: a dynamic clustering interface to Web search results. *Computer Networks: The International Journal of Computer and*

- Telecommunications Networking*, 31(11-16), 1361 - 1374 doi: doi>10.1016/S1389-1286(99)00054-7
- Zhang, T., Ramakrishnan, R. and Livny, M. (1996): BIRCH: an efficient data clustering method for very large databases. *Proc. ACM SIGMOD international conference on Management of data* New York, NY, USA
- Zhao, P. and Zhang, C.-Q. (2011): A new clustering method and its application in social networks. *Pattern Recognition Letters*, 32(15), 2109-2118. doi: 10.1016/j.patrec.2011.06.008

GOTOGene: A Method for Determining the Functional Similarity among Gene Products

Kamal Taha

Department of Electrical and Computer Engineering
Khalifa University of Science, Technology, and Research
Box 127788, Abu Dhabi, UAE

E-mail: kamal.taha@kustar.ac.ae

Abstract

We present in this paper novel techniques that determine the semantic relationships among genes and gene products. We implemented these techniques in a middleware system called GOTOGene, which resides between user application and Gene Ontology database. Given a set S of genes, GOTOGene would return another set S' of genes, where *each* gene in S' is semantically related to *each* gene in S . The framework of GOTOGene refines the concept of Lowest Common Ancestor by defining the concept of Semantically Relevant Lowest Common Ancestor using the concept of existence dependency. We evaluated GOTOGene experimentally and compared it with three other methods. Results showed marked improvement.

Keywords: middleware; Gene Ontology; semantic similarity

1 Introduction

Life science ontologies are used in different types of applications. One of these applications is the annotation of biological objects such as gene products and proteins. Biological objects are annotated with ontology concepts to semantically describe their properties. The Gene Ontology (GO) (Gene Ontology 2011) has emerged as one of the most important ontology concept and the most widely used bio-ontology. Many genomic databases use GO annotations, which assign genes to term nodes to describe these genes. GO ontology is structured as a Directed Acyclic Graphs (DAG). In this graph, GO terms are represented by nodes and the different hierarchical relations between the terms (*mostly “is-a” and “part-of” relations*) are represented by edges. The “is-a” relation represents the fact that a given child term is a subtype of a parent term, and the “part-of” relation represents part-whole relationships. The lower in the DAG a term is located, the more specific it is. When a gene product is

annotated using GO, the DAG displays the term node(s) describing this gene product in such a way that reflects how this gene product is related to other gene products. Thus, annotation of a gene with a GO term is an indicative that this gene is closely related to all other genes annotated with the same GO term and the genes annotated with ancestors and descendants of this GO term.

Biologists often need to determine the semantic similarities and relationships between genes. Semantic similarity measures in GO is widely used to identify the relationships between genes and gene products. That is because genes whose GO terms are semantically related tend to have common properties. The correlation between protein/gene expression and GO semantic similarities have been demonstrated in several studies such as (Sevilla et al. 2005, Wang et al. 2007). Functional similarity describes the similarity between genes/gene products based on the similarity between the GO terms annotating these genes/gene products. The similarity between two genes is the maximal semantic similarity of two GO terms, where one of the terms annotates one of the genes and the other annotates the other gene. That is, determining the relationships between GO terms enables the quantification of the semantic similarity of the gene products annotated with these terms. Thus, functional similarity between genes can be determined using a semantic similarity measure, since GO terms are organized in DAG.

The semantic relationships between a set of genes corresponds to that between the GO terms describing these genes, if each gene is annotated by only one GO term. But most genes have several annotation GO terms. Therefore, we need a strategy and mechanism to determine the relationships between *all the occurrences* of genes under consideration. In this paper, we propose a middleware system called GOTOGene that determines the semantic relationships between gene products and considers all the annotation terms of each gene product.

Given a set S of genes, GOTOGene would return a another set S' of genes, where *each* gene in S' is semantically related to *each* gene in S . Towards this, GOTOGene would identify the GO terms that have the closest semantic relationships with *all* GO terms annotating the genes in set S . It would first determine the most *significant* Lowest Common Ancestor (LCA) term of the terms annotating the genes in S . Towards this, the framework of GOTOGene refines the concept of LCA by defining the concept of Relevant Lowest Common

Ancestor (RLCA) and the concept of Semantically Relevant Lowest Common Ancestor (SRLCA). We observe that the terms annotating a certain set of genes have *existence dependency relationships* with the SRLCA t of these terms. That is, their existence in the GO graph is dependent on the existence of t . We developed this observation into formal sets of rules and techniques that compute the semantic relationships among GO terms.

2 Relates WORK

Pesquita (2008, 2009) defines a semantic similarity measure as a function that returns a numerical value reflecting the closeness in meaning between two ontology terms (or two sets of terms) annotating two biological entities (Pesquita et al. 2009). The authors distinguish between the comparison of two ontology terms and two sets of ontology terms. GOAT (Bada et al. 2004) proposes the mining of the *Gene Ontology Annotation* (GOA) of a database for co-occurrence of GO terms in order to acquire associations between the terms. Using this method, 600,000 associations were identified, excluding unreliable associations as well as the hierarchical relations that are explicitly represented in GO.

Node-based measures are the most cited semantic similarity measures. This approach exploits the information content (IC) of two terms being compared and of their Lowest Common Ancestor (LCA) (Coute et al. 2003, Lee et al. 2004, Lin 1998, Resnik 1999). The information content of a term is based on its frequency or probability of occurring in a corpus. Resnik (1999) uses the negative logarithm of the probability of a term to quantify its information content. Thus, a term with a high probability of occurring has a low IC. Very specific terms that are less frequent have a high IC. Resnik's similarity measure consists of determining the IC of all common ancestors of two terms and selecting the one with maximal value, since it is the most specific common ancestor of the two terms. That is, if two terms have an ancestor with high information content, they are considered to be semantically related. Since the maximum of this IC value can be greater than one, Lin (1998) introduced a normalization term into Resnik's measure. Schlicker (2006) improved Lin's measure by using a correction factor based on the probability of occurrence of the LCA. A general ancestor of terms should not have high contribution to the similarity of the terms (Schlicker et al. 2006). GOSim (Frohlich 2007) extended Resnik's similarity concept by considering *all* terms having the highest information content, based on the notion of disjunctive common ancestors. Lord (2003) computes the *information content* for each GO term as a measure of the degree of its specificity. A term that describes many genes (i.e., frequently used) is not specific and vice versa. Therefore, (Lord 2003) uses the negative logarithm of the frequency of each term to quantify its *information content*.

However, node-based measures have limitations such as: (1) they do not take into account the *distance* separating term nodes from their LCA (Frohlich 2007), (2) they use IC as the major factor for determining the

semantic similarity of term nodes, *which is inappropriate for some types/scenarios of biological ontologies*, (3) some of them rely only on the *number* of common ancestor nodes, while overlook their semantic contributions to the two nodes under consideration, and (4) many of these methods overlook the information contained in the structure of the ontology and concentrate only on the information content of a node. We take (Benabderrahmane et al. 2010, Frohlich 2007, Wang et al. 2007) as sample of current semantic similarity measures and overview them below.

Similarity method proposed by (Wang et al. 2007): The semantic similarity between terms A and B , $S_{GO}(A, B)$, is defined as:

$$S_{GO}(A, B) = \frac{\sum_{t \in I_A \cap I_B} (S_A(t) + S_B(t))}{SV(A) + SV(B)}$$

$S_A(t)$ is the contribution of term t to the semantics of A :

$$\begin{cases} S_A(A) = 1 \\ S_A(t) = \max \{w_e * S_A(t') \mid t' \in \text{childrenof}(t)\} \text{ if } t \neq A \end{cases}$$

w_e is the semantic contribution factor for edge e linking term t with its child term t' ($0 < w_e < 1$).

IntelliGO (Benabderrahmane et al. 2010): Given two terms t_i and t_j represented by their vectors \vec{e}_i and \vec{e}_j respectively, the dot product between the base vectors is:

$$\vec{e}_i * \vec{e}_j = \frac{2 * \text{Depth}(LCA)}{\text{MinSPL}(t_i, t_j) + 2 * \text{Depth}(LCA)}$$

$\text{Depth}(LCA)$ is function associating the LCA with its maximal depth. $\text{MinSPL}(t_i, t_j)$ is the minimal shortest path length between t_i and t_j .

GOSim (Frohlich 2007): It extended Resnik's similarity concept by considering *all* terms having the highest information content, based on the notion of disjunctive common ancestors:

$$\text{Sim}(t, t') = \text{IC}_{\text{share}}(t, t') = \frac{1}{|\text{DisjCommAnc}|} \sum_{t \in \text{DisjCommAnc}} \text{IC}(t)$$

3 Outline of the Approach

In the framework of GOTOGene, the structure of GO is described in terms of a graph, which we call GO Graph. In this graph, GO terms are nodes and the relationships between the terms are edges. For example, Fig. 1 presents a fragment of a GO Graph showing the ontological relationships of 29 GO terms. GOTOGene accepts Keyword-based queries with the form Q (" g_1 ", " g_2 ", ..., " g_n "), where g_i denotes a gene (or a gene product) keyword.

User selects an input (*the query, which is composed of genes that are annotated to GO*). GOTOGene would then map these genes to a set of GO terms. Let S_T denote these GO terms. GOTOGene would then determine the *Relevant Lowest Common Ancestors* (RLCA) of the set S_T . It would then find the *Semantically Relevant Lowest Common Ancestors* (SRLCA) of the set S_T . Let S_1 be a set of GO terms annotating a gene g_1 and let S_2 be another set of GO terms annotating a gene g_2 . For each term T_i from set S_1 , the concept of RLCA determines the most relevant term T_j from set S_2 to T_i and then identifies their

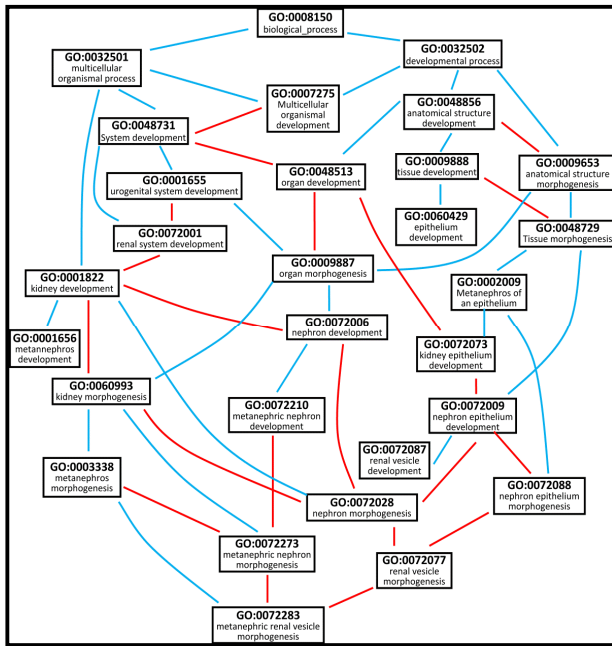


Fig. 1: A fragment of GO Graph showing the ontological relationships of 29 GO terms. Blue edges denote “is-a” relations and red edges denote “part-of” relations.

RLCA. If more than one RLCA have been identifies, the concept of SRLCA would identify the most significant one using the concept of *existence dependency*. That is, the SRLCA is a LCA, on which the *existence* of T_j and T_i depends in GO graph. GOTOGene would then convert back to genes based on annotations and retrieved back to the user. The genes annotated to the SRLCA are the most semantically related to the user’s input genes.

Notation 3.1, Keyword Context (KC): A KC is a GO term that is annotated to a query gene product. For example, consider Fig. 1 and the query Q (“JAG1”). The term organ morphogenesis (GO:0009887) is a KC because the gene “JAG1” is annotated to it.

Let S_{KC} be a set of KCs annotating user’s input genes (i.e., query). To construct the answer for this query, GOTOGene needs to identify the SRLCA of the set S_{KC} based on the concept of existence dependency. Towards this, GOTOGene will need to check all “part-of” relations in GO graph, because: “part of has a specific meaning in GO and a part of relation would only be added between A and B if B is necessarily part of A: wherever B exists, it is as part of A, and the presence of the B implies the presence of A” (Gene Ontology 2011). “part-of relation embodies some aspects of existence dependency. A part-of relation with existence dependent parts can simply be replaced by existence dependency: In case of existence dependent components, the existence dependency relation is identical to the part of relation” (Snoeck and Dedene 1998). Fig. 2 is an overview of our approach.

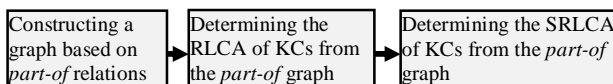


Fig 2: The sequential processing steps for answering a query

4 Constructing part-of Graph

Since not all “part-of” relations are explicitly expressed in a GO Graph (*some can be inferred from the graph*), GOTOGene converts the GO Graph into a graph called Part-Of Graph (POG), which contains only the explicit and inferred “part-of” relations. The LCA of KCs will be determined from the POG and not from the GO Graph. A POG is a GO Graph after: (1) removing all its relations except for the “part-of” ones, and (2) adding the inferred “part-of” relations. The terms A and B are connected by a “part-of” relation in the POG, if the GO Graph either states this relation expressly or it can be inferred from the graph using the following two *inference rules* described in (Gene Ontology 2011): (1) if A “is-a” B and B is “part-of” C , A is “part-of” C , and (2) if A is “part-of” B and B “is-a” C , A is “part-of” C . Fig. 3 shows a fragment of a POG derived from the GO Graph in Fig. 1. For example, since in Fig. 1: (1) the term multicellular organismal process (GO:0032501) “is-a” the term biological process (GO:0008150), (2) the term multicellular organismal development (GO:0007275) “is-a” the term multicellular organismal process (GO:0032501), and (3) the term system development (GO:0048731) is “part-of” the term multicellular organismal development (GO:0007275), then in Fig. 3 the term system development (GO:0048731) is “part-of” the term biological process (GO:0008150). In Fig. 3, each term node shows the genes that the term annotates.

We observe that the specificity of a term (*with regard to its “is-a” relations*) influences its semantic relationships with other terms. This specificity differentiates “general” functions that are close to the root from specific detailed ones. Therefore, we determine the specificity of each term. “is-a” is a simple type-subtype relation between two GO terms (Gene Ontology 2012). Consider that: (1) A' “is-a” A , (2) A “is-a” C , (3) B' “is-a” B , and (4) B “is-a” C . Both of the terms A and B inherit the characteristics and properties of their supertype C . Therefore, intuitively, A and B have the *same* specificity. Since A' and B' inherit from the characteristics and properties of terms that have the same specificity (*the terms A and B*), A' and B' have the *same* specificity also. Thus, the specificity of a term node is the number of “is-a” relations that connect it with the root term node (its “is-a” distance to the root). For example, recall Fig. 1. The root term biological process (GO:0008150) has its own specificity. Since both of the terms multicellular organismal process (GO:0032501) and developmental process (GO:0032502) inherit the same characteristics from their supertype GO:0008150, they both have the same specificity¹. As another example, the terms kidney development (GO:0001822), system development (GO:0048731), multicellular organismal development (GO:0007275), anatomical structure morphogenesis (GO:0009653), and anatomical structure development (GO:0048856) have the same specificity¹. If a term has multiple inheritances, only its shortest distance to the root is

¹ Alternatively, we can determine that these terms have the same specificity, because they have the same distance to the root based on their is-a relations.

considered. In the POG in Fig. 3, each set of terms that have the same specificity are colored with the same color for easy reference. For example, the terms kidney development (GO:0001822), system development (GO:0048731), and anatomical structure morphogenesis (GO:0009653) are colored with the same color as an indicative that they have the same specificity.

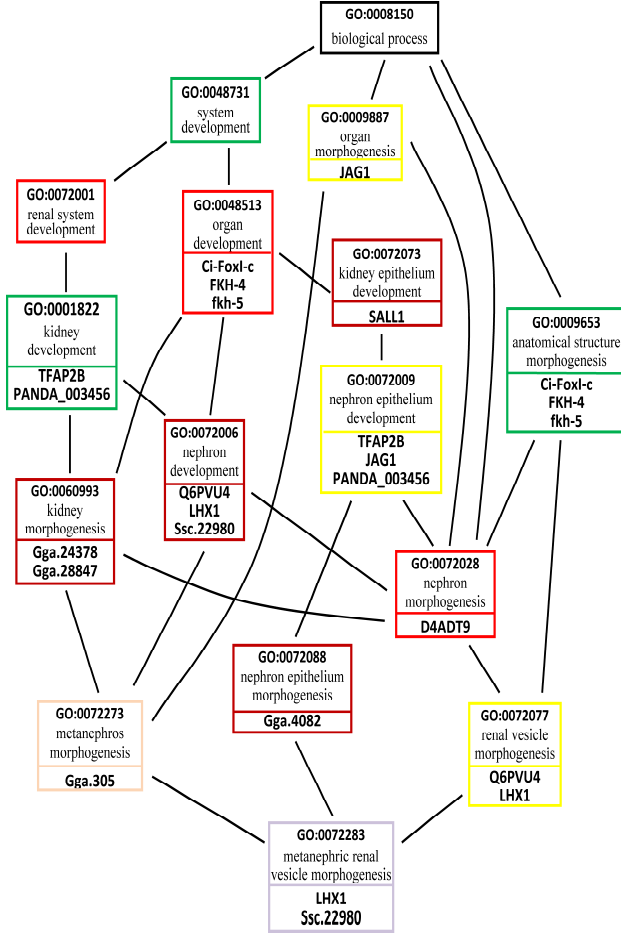


Fig. 3: POG constructed from the GO Graph in Fig. 1 after coloring each set of terms that have the same specificity with the same color. Each term node includes the genes annotated to the term.

5 Determining RLCA

In this section, we describe how GOTOGene determines the most *significant* Lowest Common Ancestor of terms annotating input genes (i.e., keywords of a query). We first formalize the notion of LCA in Definition 5.1.

Definition 5.1, Lowest Common Ancestor (LCA): Let: (1) N be the set of GO terms in a GO Graph, (2) $T_i, T_j, T_x \in N$, (3) T_x is the LCA of T_i and T_j (denoted as $LCA(T_i, T_j)$), and (4) $descendant-or-self(T_i, T_x)$ denotes that T_i is a descendant of T_x or is equal to T_x . If:

- $descendant-or-self(T_i, T_x) = \text{true}$, and
- $descendant-or-self(T_j, T_x) = \text{true}$, and
- $\forall T' \in N$, if $descendant-or-self(T_i, T') = \text{true}$ and $descendant-or-self(T_j, T') = \text{true}$, then $descendant-or-self(T_x, T') = \text{true}$.

We now introduce a notion for two or more terms that annotate at least one same gene product.

Notation 5.1, $ANTN_{T_x} = ANTN_{T_y}$: Denotes GO terms T_x and T_y annotate at least one same-gene. That is, there is at least one gene annotated to both T_x and T_y . For example, consider Fig. 3. The gene “LUX1” is annotated to GO terms GO:0072006, GO:0072077, and GO:0072283. Therefore, $ANTN_{GO:0072006} = ANTN_{GO:0072077} = ANTN_{GO:0072283}$

We now refine the concept of LCA and introduce the concept of Relevant Lowest Common Ancestor (RLCA). Let g_1, g_2, \dots, g_n be a set of input genes selected by the user (i.e., keywords of a query). Let S be the set of terms annotating the input g_1, g_2, \dots, g_n . A RLCA is a LCA of a subset $S' \subseteq S$ where the terms of S' are *meaningfully* related to each other and contain at least one occurrence of each of the genes. We present below two scenarios that describe what we mean by *meaningfully* related terms.

Scenario 1: Consider the situation where two GO term nodes have no hierarchical relationship with each other. Suppose that the LCA of T_1 and T_2 is T_x as shown in Fig. 4. We can regard both T_1 and T_2 as *meaningfully* related to each other by *belonging* to T_x . The RLCA of T_1 and T_2 is T_x . Given two sets of GO terms, where the terms in each set fall under the same annotation cluster, Definition 4.3 describes how to determine the RLCA of each pair from the two sets.

Definition 5.2, RLCA of two nodes: Let the set of GO terms in a GO Graph be N . Given $A, B \subseteq N$, where A is comprised of nodes having the same annotation A , and B is comprised of nodes having the same annotation B , the RLCA Set $C \subseteq N$ of A and B satisfies the following conditions:

- $\forall c_k \in C, \exists a_i \in A, b_j \in B$, such that $c_k = LCA(a_i, b_j)$. c_k is denoted as $RLCA(a_i, b_j)$.
- $\forall a_i \in A, b_j \in B$, if $d_{ij} = LCA(a_i, b_j)$ and $d_{ij} \notin C$, then $\exists c_k \in C$, $descendant(c_k, d_{ij}) = \text{true}$.

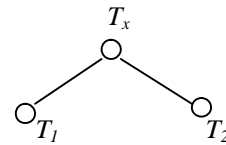


Fig. 4: Relationships between nodes located in adjacent hierarchical levels

Scenario 2: As demonstrated by Fig. 5, let there be two terms T'_2 and T_2 , where $ANTN_{T'_2} = ANTN_{T_2}$. Let the LCA of T_1 and T'_2 be T' . If T_x is an ancestor of T' , we should then conclude that nodes T_1 and T_2 are *not* meaningfully related to each other, because: (1) T_1 is more related to T'_2 than to T_2 (recall that

$ANTN_{T_2} = ANTN_{T_2'}$, and (2) the LCA of T_1 and T_2 is an ancestor of the LCA of T_1 and T_2' . Term T' is the RLCA of T_1 and T_2' . We formalize this concept in Definition 5.3.

Definition 5.3 : Let $ANTN_A = ANTN_B \neq ANTN_C$. Node C is relevantly related to node A and not to node B if the LCA of nodes A and C is a descendant of the LCA of nodes C and B . The LCA of nodes C and A is a RLCA.

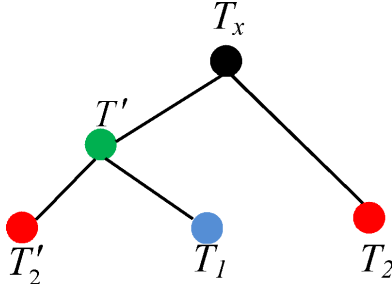


Fig. 5: Hierarchical relationships among term nodes of different specificities. A node's color represents its specificity

Example 1: Consider Fig. 3 and the query $Q(\text{"LHX1", "Gga.4082"})$. As shown in Fig. 3: (1) the KCs annotating the gene "LHX1" are *nephron development* (GO:0072006), *renal vesicle morphogenesis* (GO:0072077), and *metanephric renal vesicle morphogenesis* (GO:0072283), and (2) the KC annotating the gene "Gga.4082" is *nephron epithelium morphogenesis* (GO:0072088). The term GO:0072088 is more related to GO:0072077 than to GO:0072006, because: (1) $ANTN_{GO:0072077} = ANTN_{GO:0072006}$, and (2) the LCA of GO:0072006 and GO:0072088 (which is GO:0048513) is an ancestor of the LCA of GO:0072088 and GO:0072077 (which is GO:0072009). Therefore, GO:0072009 is the RLCA of GO:0072088 and GO:0072077 and the genes it annotates (i.e., the genes "TFAP2B", "JAG1", and "PANDA_003456") are related to both of the input gene keywords "Gga.4082" and "LHX1". Fig. 6 shows the subtree rooted at the RLCA node GO:0072009.

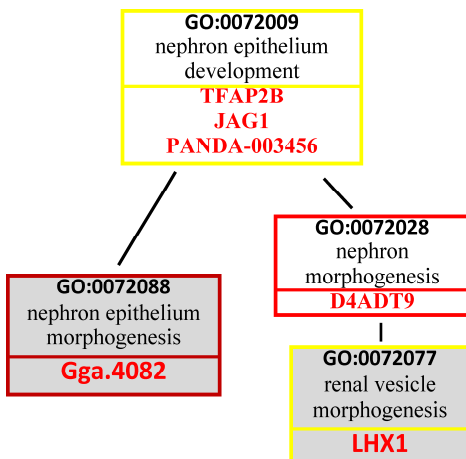


Fig. 6: GO:0072009 is a RLCA of GO:0072088 and GO:0072077

Example 2: Consider Fig. 3 and the keyword query $Q_2(\text{"JAG1", "LHX1"})$. The GO terms annotating the gene "JAG1" are *organ morphogenesis* (GO:0009887) and *nephron epithelium development* (GO:0072009). Recall example 1 for the GO terms annotating the gene "LHX1". The term GO:0072006 is more related to GO:0072009 than to GO:0009887, because: (1) $ANTN_{GO:0072009} = ANTN_{GO:0009887}$, and (2) the LCA of GO:0072006 and GO:0009887 (which is GO:0008150) is an ancestor of the LCA of GO:0072006 and GO:0072009 (which is GO:0048513). Therefore, GO:0048513 is the RLCA of GO:0072006 and GO:0072009 and the genes it annotates (i.e., the genes *Ci-Foxl-c*, *FKH-4*, and *fkh-5*) are related to both of the input genes "JAG1" and "LHX1". Fig. 7 shows the subtree rooted at the RLCA GO:0048513.

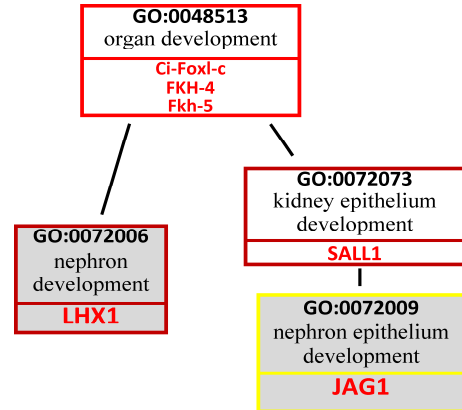


Fig. 7: GO:0048513 is a RLCA of GO:0072006 and GO:0072009

We constructed an algorithm called *GetRLCA* (see Fig. 8) that determines the RLCA for an input set of genes. Function *getAnnotations* (see line 1) returns the terms annotating the input genes and stores the results in set S . Line 2 stores in set S' the duplicate annotations in set S . In line 6, function *GetLCA* (see Fig. 9) returns the LCA of an input set of annotations, and function *getAncestors* returns all ancestors of the annotations. In lines 6 and 7, if the LCA of nodes n_i and $s' - s''$ is an ancestor of the LCA of nodes n_j and $s' - s''$, line 7 will return the later LCA as the RLCA.

```

GetRLCA ( $g_1, g_2, \dots, g_n$ ) {
1.  $S \leftarrow \text{getAnnotations}(g_1, g_2, \dots, g_n)$ 
2.  $S' \leftarrow S - \text{distinct}(S)$ 
3. for each term node  $n \in S'$ 
4.    $S'' \leftarrow \text{distinct}(S) - \text{annotation}(n)$ 
5.   for each term nodes  $n_i, n_j \in S''$ 
6.     if  $\text{GetLCA}(n_i, (S' - S'')) \in \text{getAncestors}(\text{GetLCA}(n_j, (S' - S'')))$ 
7.        $\text{RLCA} \leftarrow \text{GetLCA}(n_j, (S' - S''))$ 
}
    
```

Fig. 8: Algorithm *GetRLCA*

```

GetLCA ( $T_1, T_2, \dots, T_n$ ) {
1.  $S \leftarrow \text{getAncestors}(T_1) \cap \text{getAncestors}(T_2) \dots \cap \text{getAncestors}(T_n)$ 
2. if ( $S \neq \text{null}$ )
3.    $\text{LCA}(T_1, T_2, \dots, T_n) \leftarrow \text{getDescendantAll}(S)$ 
}
    
```

Fig. 9: Subroutine *GetLCA*

6 Determining SRLCA

From the set of RLCA, GotoGene needs to determine the ones that are *semantically related* to the KCs. We call each one the Semantic Relevant Lowest Common Ancestor (SRLCA) of the KCs. A SRLCA should be semantically related to *each* of the KCs, otherwise the answer subtree is considered invalid.

We use the notation SR_{KC} to denote the set of terms that are *semantically related* to the KC. The notation $T \in SR_{KC}$ denotes that term T is semantically related to the KC. In order for a $RLCA \in SR_{KC}$ (i.e., to be SRLCA), we observe that: (1) the RLCA should have a different specificity than the KC, and (2) the path from the RLCA to the KC in the GO Graph should not include two or more terms with the same specificity. For example, recall the query in example 1. As described in the example, the term GO:0072009 is a RLCA of the KCs GO:0072088 and GO:0072077. However, $GO:0072009 \notin SR_{GO:0072077}$, because its specificity is the same as the specificity of GO:0072077. Therefore, GO:0072009 is not a SRLCA for GO:0072077. Based on these observations, we introduce proposition 6.1 below.

Proposition 6.1: SRLCA: *In order for a RLCA to be a SRLCA: (1) its specificity should be different than the specificities of the KCs, and (2) the path in the GO Graph from the RLCA to each of the KCs should not include two or more terms with the same specificity.*

Notation 6.1, $SPEC_x$: $SPEC_x$ denotes the specificity of GO term x .

We prove observation/proposition 6.1 heuristically as follows. First, we prove: *if a $RLCA \in SR_{KC}$, then $SPEC_{RLCA} \neq SPEC_{KC}$* . That is, in order for a RLCA to be SRLCA, its specificity should be different than the specificity of the KC. We are going to validate this observation by checking whether it conforms to the structural characteristics of *existence dependency*. The concept of existence dependency was first proposed for Entity-Relationship modeling (Elmasri, and Navathe 2011). An object x is *existence-dependent* on an object y if the existence of x is dependent on the existence of y (Widjaya et al. 2003). The existence dependency concept and the SR_{KC} concept have correspondences: both denote that *an object(s) has a strong association with another object*. SR_{KC} is a set of GO terms, whose *existence* in a POG is *dependent* on the existence of the KC (or conversely, the *existence of the KC in the graph is dependent on the existence of the set of terms*). Snoeck et al. (1998) argue that the existence dependency relation is a partial ordering of *object types* (i.e., *specificities*). The authors transform an OO schema into a graph consisting of the *object types* found in the schema and their relations. The object types in the graph are related only through associations that express existence dependency. The authors demonstrated through the graph that *an object type is never existence-dependent on itself*. That is, if the two objects O_i and O_j belong to the same type, O_i

cannot be dependent on O_j and vice versa. This finding is in agreement with our proposed rule, when we view: (1) a GO term in a GO Graph as an object, and (2) a GO term's specificity as an object's type. Thus, if a RLCA has the same specificity as the KC, the RLCA can never be existence-dependent on the KC (and vice versa); therefore, this RLCA is NOT SRLCA and the genes annotated to it may not be semantically related to the genes annotated to the KCs.

Second, we prove: *If a RLCA is semantically related to the KC, then $SPEC_{T_x} \neq SPEC_{T_y}$ where T_x and T_y are term nodes located between the RLCA and the KC in the POG*. We can verify this rule as follows. In order for $RLCA \in SR_{KC}$, all term nodes located between the RLCA and the KC in the POG have to be related to the KC. Let: (1) term $T_y \in SR_{KC}$, (2) T_y be a descendant of the KC, and (3) term T_x be a descendant of T_y . In order for T_x to be semantically related to the KC, intuitively T_x has to be semantically related T_y , because T_y relates (connects) T_x with the KC. If T_x and T_y have the same specificity, then $T_x \notin SR_{T_y}$ (according to the first rule). Therefore, in order for T_x to be semantically related to the KC, $SPEC_{T_x} \neq SPEC_{T_y}$.

Example 3: Recall example 1. By applying proposition 6.1, GO:0072009 is NOT a SRLCA for GO:0072088 and GO:0072077, because GO:0072009 and GO:0072077 have the same specificity. Therefore, the genes annotated to GO:0072009 (i.e., the genes *TFAP2B*, *JAG1*, and *PANDA_003456*) may not be semantically related to the input gene keywords "Gga.4082" and "LHX1".

Example 4: Recall example 2. By applying proposition 6.1, the RLCA of GO:0072088 and GO:0072077 (i.e., the term GO:0048513) is a SRLCA. Therefore, the genes annotated to GO:0048513 (i.e., the genes *Ci-Foxl-c*, *FKH-4*, and *fkh-5*) are semantically related to both of the input gene keywords "JAG1" and "LHX".

7 Experimental Results

We experimentally evaluated the quality of GotoGene and compared it with (Benabderrahmane et al. 2010, Frohlich 2007, Wang et al. 2007) (recall their descriptions in section 2). We implemented GotoGene in Java, run on Intel(R) Core(TM)2 Dup CPU processor, with a CPU of 2.1 GHz and 3 GB of RAM, under Windows Vista. The implementation of (Frohlich 2007) was released as part of GOSim package (Cran 2012), which we used for the evaluation of (Frohlich 2007). We implemented the methods of (Benabderrahmane et al. 2010, Frohlich 2007, Wang et al. 2007) from scratch.

7.1 Benchmarking Datasets

Pathways are sets of genes shown to have high functional similarity and can be used to validate similarity measures (Guo 2006, Nagar and Al-Mubaid 2008, Wang 2004). A fully describe a pathway represent the dynamics and dependencies among a set of gene/gene products. Therefore, we used in our experiments pathways as a

reference for evaluating and comparing the similarity (and semantic relationships) measures of GOtoGene and (Benabderrahmane et al. 2010, Frohlich 2007, Wang et al. 2007). Given a set S of genes, a system/method should return a set S' of other genes that are semantically related to S . In order for sets S and S' to be related, S and S' should be part of a *same pathway*.

We used for the evaluation two different benchmarks: KEGG and Pfam benchmarks. We selected 15 groups of highly related Pfam entries (see Table 1) from the Sanger Pfam database. We selected a set of 15 human and 15 yeast diverse KEGG pathways (see Tables 2 and 3) containing between 10 and 30 genes, which were retrieved using the *DBGET* database. For each group, we retrieved the corresponding human and yeast gene identifiers from the Uniprot database. Assuming that genes belonging to the same KEGG pathway are often related to a similar biological process, the similarity values calculated for this dataset should be related to the BP GO aspect. And, assuming that genes which share common domains in a Pfam clan often have a similar molecular function, the similarity values calculated for this second dataset should be related to the MF GO aspect.

7.2 Evaluating Recall and Precision

We measured the *recall* (or *true positive rate*) and *precision* of GOtoGene and of (Benabderrahmane et al. 2010, Frohlich 2007, Wang et al. 2007). Recall (or *true positive rate*) is the *fraction* of correct genes determined by a similarity measure relevant to all genes determined by the measure. Precision is the *fraction* of correct genes determined by a similarity measure relevant to all genes in the pathway. Let: (1) G_P be all genes in the pathway and n be the number of these genes, and (2) G_M be the top n genes determined by a similarity measure, which are semantically related to the input gene keywords. $\text{recall} = (|G_M \cap G_P| / |G_M|)$. $\text{Precision} = (|G_M \cap G_P| / |G_P|)$. We computed the recall and precision for the four methods as follows. We first measured the semantic similarity/relationship between each term in the GO Graph and the set of terms annotating the input gene keywords using each of the four methods. We then clustered the genes based on the obtained similarity values. G_M are the top n genes in each cluster with the highest similarity values. As for GOtoGene, G_M are n genes annotated by the GO terms located in the paths from a SRLCA to the terms annotating the input gene keywords.

Fig. 10 shows the recall and precision results obtained with the KEGG pathways. Fig. 11 shows the recall and precision results obtained with the pfam pathways. For each KEGG and pfam pathway (x-axis), the recall and precision values are represented as histograms (y-axis). As the figures show, recall and precision values vary based on: (1) pathways, and (2) the accuracy of each of the four methods to capture the semantic similarities and relationships among gene annotations within pathways.

Table 1. The 15 Pfam Human Pathways and the 15 Pfam Yeast Pathways used in the experiments

Pfam Accession	Pfam ID	Number of genes (human)	Number of genes (yeast)
CL0406	vWA-like	11	6
CL0344	4Fe-4S	7	4
CL0461	Metallothionein	18	11
CL0020	TPR	13	6
CL0418	GIY-YIG	8	19
CL0417	BIR-like	10	6
CL0233	SufE_NifU	9	10
CL0167	Zn_Beta_Ribbon	7	5
CL0099	ALDH-like	18	11
CL0042	Flavoprotein	10	7
CL0040	tRNA_synt_II	12	2
CL0179	ATP-grasp	7	6
CL0417	BIR-like	11	9
CL0445	SNARE-fusion	8	6
CL0444	YNI	9	5
Total number of genes		158	113

Table 2. The 15 KEGG Human Pathways used in the experiments.

Pathway	Name	# of genes
sce00562	Inositol phosphate metabolism	15
sce00920	Sulfur metabolism	15
sce00600	Sphingolipid metabolism	13
sce00410	beta-Alanine metabolism	12
sce00514	Saccharomyces cerevisiae	13
sce00670	One carbon pool by folate	15
sce00903	Limonene and pinene degradation	20
sce03022	Basal transcription factors	32
sce04130	SNARE interactions in vesicular transport	23
sce03450	Non-homologous end-joining	10
sce04070	Phosphatidylinositol signaling system	15
sce04140	Regulation of autophagy	17
sce04111	Saccharomyces cerevisiae	25
sce04011	MAPK signaling pathway	57
sce03010	Ribosome	12
Total number of genes		294

Table 3. The 15 KEGG Yeast Pathways used in the experiments.

Pathway	Name	# of genes
hsa00040	Pentose and glucuronate interconversions	34
hsa00920	Sulfur metabolism	14
hsa00140	Steroid hormone biosynthesis	26
hsa00290	Valine, leucine and isoleucine biosynthesis	5
hsa00563	Glycosylphosphatidylinositol	25
hsa00670	One carbon pool by folate	19
hsa00232	Caffeine metabolism	7
hsa03022	Basal transcription factors	23
hsa04130	SNARE interactions in vesicular transport	36
hsa03450	Non-homologous end-joining	13
hsa03430	Mismatch repair	23
Hsa00085	Fatty acid biosynthesis	12
hsa04950	Maturity onset diabetes of the young	25
hsa04803	Homo sapiens	16
hsa00120	Primary bile acid biosynthesis	14
Total number of genes		292

In summary, the recall and precision values for the two benchmarking datasets showed that **GotoGene** outperforms the other three methods. The results reveal the robustness of the **GotoGene**'s method and its ability to reflect the semantic relationships among gene annotations.

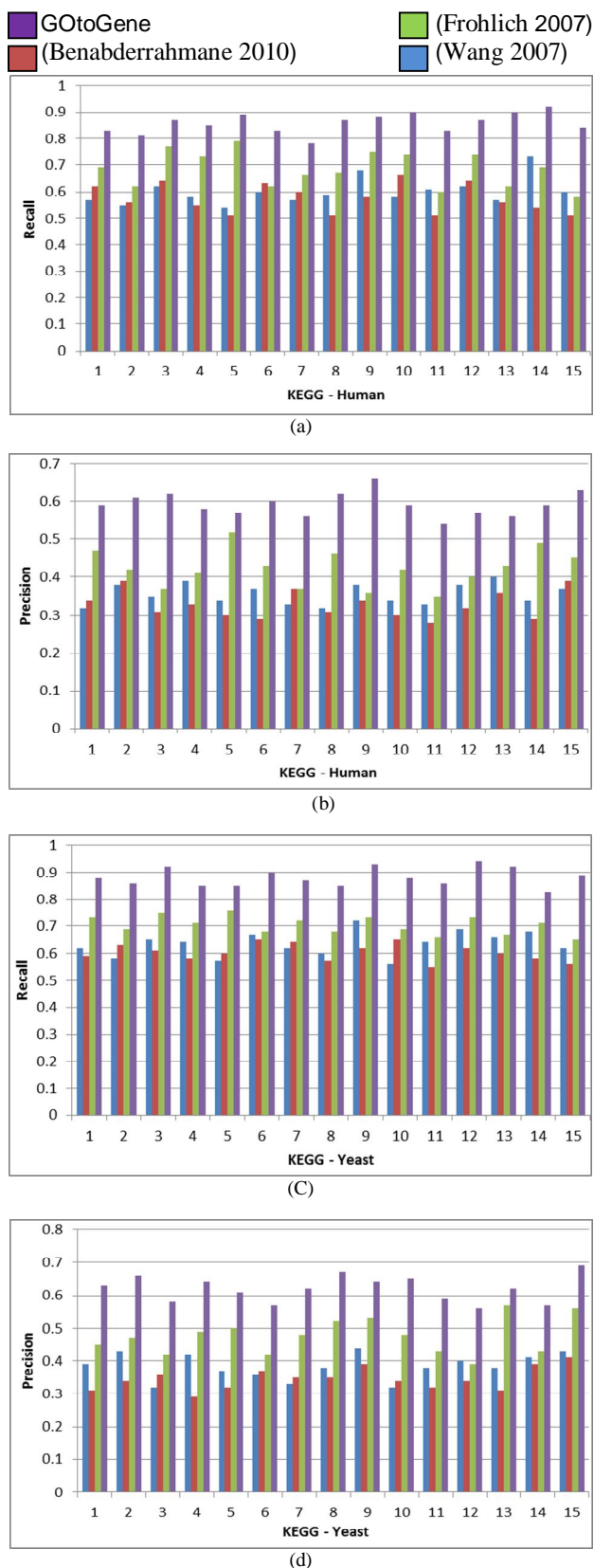


Fig. 10: Recall and precision using KEGG benchmark

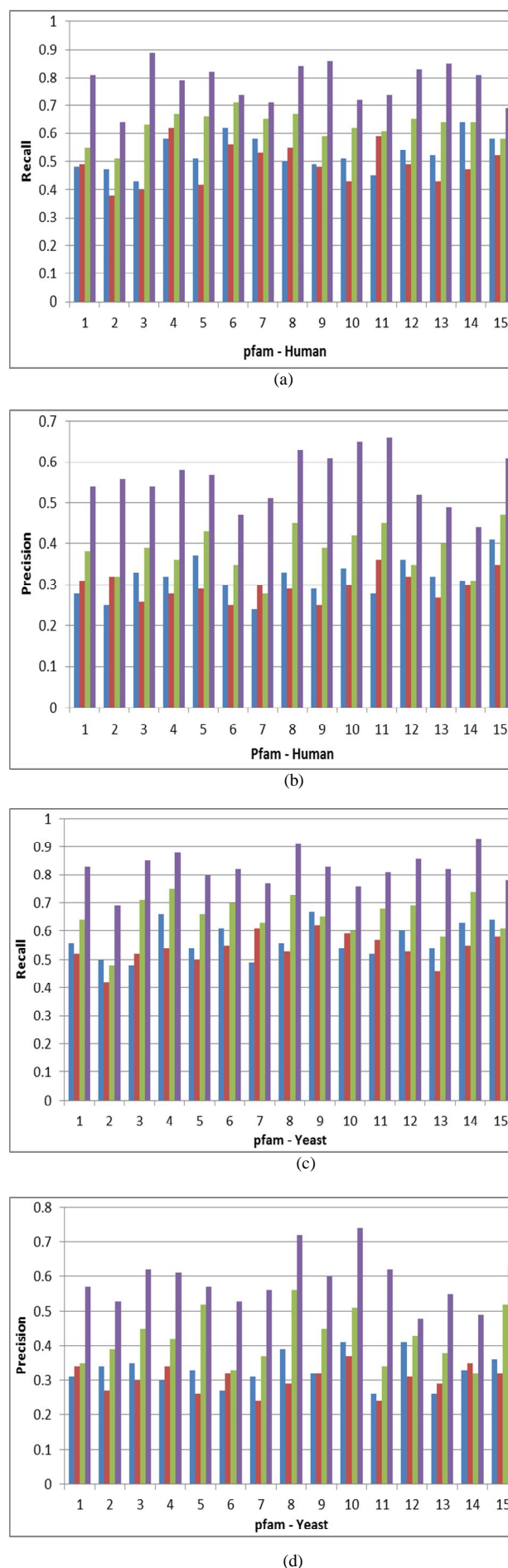


Fig. 11: Recall and precision using Pfam benchmark

8 Conclusion

In this paper, we proposed a system called **GOTOGene** that determines the semantic relationships among genes and gene products using the concept of existence dependency. Given a set of genes g_1, g_2, \dots, g_n , **GOTOGene** identifies the Semantic Relevant Lowest Common Ancestor (SRLCA) of the terms annotating g_1, g_2, \dots, g_n in the GO graph. The genes annotated by the SRLCA have the closest semantic relationships with g_1, g_2, \dots, g_n . We experimentally evaluated the quality of **GOTOGene** and compared it with (Benabderrahmane et al. 2010, Frohlich 2007, Wang et al. 2007). Results showed that **GOTOGene** outperforms the other three methods.

9 References

- Bada, M., Turi, D., McEntire, R. & Stevens, R. Using Reasoning to Guide Annotation with Gene Ontology Terms in GOAT. SIGMOD Record 33(2004).
- Benabderrahmane, S., Smail-Tabbone, M., Poch, O., Napoli, A., Devignes, M. IntelliGO: a new vector-based semantic similarity measure including annotation origin. BMC Bioinformatics 11:588, 2010.
- Coute F, et al. DI/FCUL TR 03-29. Department of Informatics, University of Lisbon; 2003. Implementation of a Functional Semantic Similarity Measure between Gene-Products, 2003.
- Elmasri, R., Navathe, S. "Fundamentals of Database Systems", Addison-Wesley Computing, six edition, 2011.
- Frohlich, H. GOSim – an R-package for computation of information theoretic GO similarities between terms and gene products. BMC Bioinformatics 8 (1), 166, 2007.
- Gene Ontology (2011). <http://www.geneontology.org/GO.ontology.relations.shtml>
- Guo X, Liu R, Shriver CD, Hu H, Liebman MN: Assessing semantic similarity measures for the characterization of human regulatory pathways. *Bioinformatics* 2006, 22(8):967-973
- Lee SG, et al. A graph-theoretic modeling on GO space for biological interpretation of gene clusters. *Bioinformatics* 2004; 20:381-388.
- Lin D. An information-theoretic definition of similarity, Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy; In Proc. 15th Int'l Conference on Machine Learning, 1998, pp.296-304.
- Lord, P.W., Stevens, R.D., Brass, A. and Goble, C.A. Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation. (2003) *Bioinformatics* 19, 1275-83
- Nagar A, Al-Mubaid H: A New Path Length Measure Based on GO for Gene Similarity with Evaluation using SGD Pathways. 21st IEEE International Symposium on Computer-Based Medical Systems (CBMS 08)
- Pesquita C, Faria D, Bastos H, Ferreira A, Falcao AO, Couto F: Metrics for GO based protein semantic similarity: a systematic evaluation. *BMC Bioinformatics* 2008, 9(Suppl 5):S4.
- Pesquita C, Faria D, Falcao AO, lord P, Couto F: Semantic Similarity in Biomedical Ontologies. *PLoS Comput Biol* 2009, 5(7):e1000443.
- Resnik P. Semantic similarity in a taxonomy: an information-based measure and its application to problems of ambiguity in natural language. *J. Artificial Intelligence Res.* 1999; 11:95-130.
- Schlicker A, Domingues F, Rahnenfuhrer J, Lengauer T. A new measure for functional similarity of gene products based on Gene Ontology. *BMC Bioinformatics* 2006;7:302.
- Sevilla JL, Segura V, Podhorski A, Mato JM, Corrales FJ, Rubio A: Correlation between Gene Expression and GO Semantic Similarity. *IEEE/ACM Trans. Comput. Biol. Bioinformatics* 2005, 2(4):330{338}.
- Snoeck, M. Dedene, G. "Existence Dependency: The key to semantic integrity between structural and behavioral aspects of object types", *IEEE Transactions on Software Engineering*, Vol. 24, No. 24, pp.233- 251, 1998.
- Wang, H. et al. (2004) Gene Expression Correlation and Gene Ontology-Based Similarity: An Assessment of Quantitative Relationships. *IEEE Symposium on Computational Intelligence in Bioinformatics*.
- Wang J., Du Z, Payattakool R, Yu, P., Chen, C. A new method to measure the semantic similarity of go terms. *Bioinformatics* 2007; 23:1274-1281
- Widjaya, N., Taniar, D., Rahayu, W. "Aggregation Transformation of XML Schema to Object-Relational Databases", *Innovative Internet Community Systems, LNCS 2877*, pp. 251-262, 2003.
- Xu, Q., Shi, Y., Lu, Q., Zhang, G., Luo, Q., Li, Y. GORouter: an RDF model for providing semantic query and inference services for Gene Ontology and its associations. *BMC Bioinformatics* 2008.
- XML Query Use Cases, *W3C Working Draft 2007*. Available: <http://www.w3.org/TR/xquery-use-cases/>
- CRAN - Package GOSim, 2012. Available at: <http://cran.r-project.org/web/packages/GOSim/index.html>

Functional Visualisation of Genes using Singular Value Decomposition

Hamid Ghous¹ Paul J. Kennedy¹ Nicholas Ho² Daniel R. Catchpoole²

¹ Centre for Quantum Computation and Intelligent Systems,
School of Software, Faculty of Engineering and Information Technology,
University of Technology, Sydney, PO Box 123, Broadway NSW 2007, AUSTRALIA
Email: Hamid.Ghous@student.uts.edu.au, Paul.Kennedy@uts.edu.au

² Biospecimens Research Group and Tumour Bank, Children's Cancer Research Unit,
The Kid's Research Institute, The Children's Hospital at Westmead,
Locked Bag 4001, Westmead NSW 2145, AUSTRALIA.
Email: Nicholash@chw.edu.au, DanielC@chw.edu.au

Abstract

Progress in understanding core pathways and processes of cancer requires thorough analysis of many coding regions of the genome. New insights are hampered due to the lack of tools to make sense of large lists of genes identified using high throughput technology. Data mining, particularly visualisation that finds relationships between genes and the Gene Ontology (GO), has the potential to assist in functional understanding. This paper addresses the question of how well GO annotations can help in functional understanding of genes. We augment genes with associated GO terms and visualise with Singular Value Decomposition (SVD). Meaning of derived components is further interpreted using correlations to GO terms. The results demonstrate that SVD visualisation of GO-augmented genes matches the biological understanding expected in the simulated data and presents understanding of childhood cancer genes that aligns with published results.

Keywords: singular value decomposition, visualisation, genes, gene ontology.

1 Introduction

It is becoming clear that progress towards new insights in cancer treatment require a thorough analysis of many genes (Jones et al. 2008). The routine use of microarray-based high-throughput technology has made more data available for interpretation and consideration by biologists. However, the sheer scale of this data makes understanding by humans challenging. Also, as integration of multiple datasets becomes commonplace, for example using single nucleotide polymorphisms or the proteome, making sense of the data becomes even more difficult. Adding to this complexity is the fact that since genes do not have a one-to-one mapping to phenotype, genes highlighted by experiments in one area of biology may have been discovered and annotated in a different area. Consequently, the gene name may not assist in understanding gene function. For these reasons, researchers have investigated ways of making sense of lists of genes by augmenting or enriching the data with functional

information from databases such as the Gene Ontology (Ashburner et al. 2000).

The Gene Ontology is a structured vocabulary of gene products and functions curated by biologists, currently consisting of more than 28,000 terms, associated annotations and links to corroborating databases. It is composed of three sub-ontologies: molecular functions, cellular components and biological processes. Terms in these hierarchies relate to the biochemical activity, the physical location and the biological objective of gene products respectively. One or more terms are related to individual genes. Each term may have multiple parents in the sub-ontology using, predominantly, inheritance (or “is-a”) and containment (“kind-of”) relationships. The hierarchical structure between terms facilitates the construction of similarity measures between the genes by calculating the similarities between the terms associated with the genes.

The Gene Ontology project is a collaborative effort since 1998 that aims to address the need for consistent descriptions of gene products in different databases. The Gene Ontology structure is based on terms with each term consisting of (i) a unique alphanumerical identifier (GO:#####); (ii) a term name, e.g., cell, fibroblast growth factor receptor binding or signal transduction; (iii) synonyms (if applicable); and (iv) a definition. Each term belongs to one of the three hierarchies, which are structured as directed acyclic graphs. Each gene has one or more terms related to it and a term may have multiple parents in the hierarchy. Together these terms provide us with a description of the known functionality of a gene. One challenge with using terms from the Gene Ontology is that terms give different amounts of information. For example, some genes are associated with only very general terms shared by many other genes whereas others are associated with very specific terms. Also, some genes are not associated with many terms. In short, the information associated with genes in the Gene Ontology is of mixed quality.

There has been much recent work to explore the problem of applying unsupervised learning methods to lists of genes. Work generally falls into two main areas: defining similarity measures using GO annotations and applying unsupervised methods to visualise the functional relationship between genes. Sheehan et al. (2008) describe several approaches for similarity measures between GO annotations including those based on sets, vectors, graphs and terms. They propose an algorithm that finds specific common ancestors between terms over the hierarchical GO struc-

ture. Richards et al. (2010) assess functional coherence of a gene set using both a graph-based similarity measure and an information content similarity measure. Mistry & Pavlidis (2008) define a term overlap measure for gene functional similarity. They make a set of all the annotations related to a gene and all the parent terms, compare them to other genes and fetch the common terms. The greater the number of common terms the higher the similarity. Mathur & Dinakarandian (2007) use the hierarchical structure of GO to compute similarity between gene products on the basis of common GO terms. Sanfilippo et al. (2007) propose a cross-ontological approach that exploits similarity measures over the ontologies in two ways: firstly, by calculating similarity within a sub-ontology and secondly by finding inter-gene relationships across the three sub-ontologies. The latter method identifies gene annotations in a sub-ontology based on the annotations for similar genes. Yi et al. (2007) find functionally similar genes in close proximity on chromosomes. Lee et al. (2004) find clusters of genes according to significant biological features using the hierarchical GO structure. They define a similarity measure by transforming the directed acyclic graph structure of GO into a distance function, which results in clusters of genes with similar terms or functionality. Similarly Popescu et al. (2004) use GO terms to extract a functional summary of gene clusters. They identify the highest frequency terms by applying fuzzy methods to clusters of genes and produce a hierarchical clustering of genes that results in clusters labelled with the “most representative term” of the contained genes.

Huang et al. (2008) evaluate tools for functional analysis of large gene lists. They classify tools according to key statistical methods and divide them into three categories based on singular enrichment analysis, gene set enrichment analysis and modular enrichment analysis. These categories give users a list of the strengths and limitations of tools. Huang et al. (2007) describe the tool ‘DAVID’ for finding functional relationships between a set of genes using statistical methods such as heuristic fuzzy multiple-linkage partitioning. FuncAssociate (Berriz et al. 2009) has been developed to identify the enriched properties from a list of genes or proteins and uses the hierarchical structure of GO and the synergizer database (Berriz & Roth 2008): a database developed from several different data sources. Similarly, GeneTrail (Backes et al. 2007) helps in finding functional enrichments in gene and protein data sets by using two statistical methods: over-representation methods and gene set enrichment analysis. Speer et al. (2005) and Fröhlich et al. (2007) cluster genes with an information-theoretic kernel function to calculate the similarity between genes using GO. The motivation behind this approach as opposed to a distance measure using the distance over the GO graph is to better handle the variable branching and density of GO. They derive gene clusters by applying a dual k -means clustering algorithm. However few of these reviewed methods are used in routine biomedical research.

In this paper we apply singular value decomposition to visualisation of genes. Our motivation for applying SVD compared to other dimensionality reduction methods such as Principal Component Analysis (PCA) is that genes and terms may be visualised on the same graph. This allows improved understanding of the biological function of genes. The approach is applied to two data sets: a data set used to validate the approach composed of genes selected from the KEGG database (Kanehisa et al. 2008) and a data

set of genes highlighted from biological experiments in childhood cancer. Our approach differs from those above by recognising that functionality needs to be described over several ‘axes’. Rather than looking at only two or three functional dimensions, we find that it is valuable to also examine later dimensions that describe more subtle functional similarities between genes. Our approach differs from commercial products like Metacore and Ingenuity by focusing on gene functionality rather than metabolic pathways. Whilst we agree that metabolic pathways are important, our motivation is to concentrate on full explication of functional interrelationships before augmenting data with pathway interconnectivity.

2 Methods

2.1 Singular Value Decomposition

Singular value decomposition (Golub & Van Loan 1996) is a method that transforms a data matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$ into the orthogonal matrices $\mathbf{U} \in \mathbb{R}^{n \times r}$, $\mathbf{V} \in \mathbb{R}^{m \times r}$ and a diagonal matrix $\mathbf{D} \in \mathbb{R}^{r \times r}$ where $r \leq m$ is the rank of \mathbf{X} .

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T \quad (1)$$

Row vectors of \mathbf{U} relate to the original data points (rows of \mathbf{X}) and rows of \mathbf{V} are associated with the data attributes (columns of \mathbf{X}). The columns of \mathbf{U} are called the left singular vectors of \mathbf{X} and columns of \mathbf{V} are called the right singular vectors. The elements of \mathbf{D} are termed the singular values of \mathbf{X} . Singular value decomposition has been used often in bioinformatics, for example, in visualisation of gene expression values (Tomfohr et al. 2005), but the novelty in our work is to augment lists of genes with knowledge from a domain ontology and to use the later principal components to extract superior understanding.

In this study, we apply SVD to an augmented data matrix that reflects term similarities. Before applying SVD, the matrix is centered and scaled.

2.2 Incorporating functional information into the SVD

Given a set of genes G define T as the set of GO terms directly associated with any of the genes. From G we create a matrix $\mathbf{X} \in \mathbb{R}^{n \times t}$ where n is the number of genes $|G|$ and t the number of GO terms $|T|$. Each element x_{ij} of \mathbf{X} has the value 1 if the gene i is directly associated with term j otherwise 0. This is similar to computational linguistics where “genes” are replaced by “documents”.

This data matrix is augmented by information reflecting inter-term similarities. A symmetric proximity matrix $\mathbf{P} \in \mathbb{R}^{t \times t}$ is created with elements $0 \leq p_{ij} \leq 1$ representing the proximity (or similarity) between GO terms i and j . Terms with a close relationship have values close to 1, with the diagonal elements $p_{ii} = 1$. The proximity between GO terms is based on the number of links (or distance) between them and is defined as $p_{ij} = (d_{ij} + 1)^{-1}$ where d_{ij} is the minimum distance between terms i and j over the hierarchy using “is-a” links which are more frequent than “kind-of” relationships, extracted from GO using SQL. The augmented data matrix is defined as $\mathbf{X}' = \mathbf{X}\mathbf{P}$. SVD is applied to \mathbf{X}' after centring and normalisation. Whilst proximity matrices have been used for text kernels, we are unaware of their use with GO terms. Pearson correlation between GO terms and data projected into PC space is calculated and

important terms are those with higher absolute values of correlation.

2.3 Data sets

Two datasets are interrogated in this study: a validation set of genes selected from known classes and a data set of genes identified from an experiment in the cancer domain.

2.3.1 KEGG data set

A set of genes has been selected from the Kyoto Encyclopaedia of Genes and Genomes (KEGG) database (Kanehisa et al. 2008), which includes a functional classification of genes independent of the GO. The rationale is to validate our approach with genes of known functional similarity. KEGG links genomes to their biological systems and is a series of interconnected databases that interrelate (i) genes and proteins, (ii) chemical building blocks, (iii) molecular interaction pathways and (iv) hierarchies of biological objects. The last of these, KEGG BRITE, links genes into a functional hierarchy called the KEGG Orthology (KO). This hierarchy is different to the GO and has been constructed independently. We validate our approach by extracting genes from classes based on their KO terms and visualise them using GO terms. Our KEGG data set (see Table 1) contains genes (also in GO) from five KO classes: ribosome (ko03010, class 1), RNA polymerase (ko03020, class 2), transcription (ko01210, class 3), pentose phosphate pathway (ko00030, class 4) and pentose and glucuronate interconversions (ko00040, class 5). We expect genes in classes 1, 2 and 3 will be similar (with classes 2 and 3 more similar than to class 1). Genes in classes 4 and 5 should be similar to one another but different to the other classes.

2.3.2 Acute Lymphoblastic Leukaemia data set

Acute Lymphoblastic Leukaemia (ALL) is the most common childhood malignancy with around 250 children in Australia diagnosed annually. Microarray technology has been used extensively in attempts to identify markers that are predictive of treatment outcome in ALL.

The ALL dataset was constructed by building on previous work by Flotho et al. (2007) and Catchpoole et al. (2008). Flotho et al. reported a fourteen gene signature (encompassed by fifteen Affymetrix expression probesets) that separated a cohort of ALL patients treated at the St. Jude Children's Research Hospital into two distinct groups. The observed separation was associated with relapse potential, leading the investigators to conclude the fourteen gene signature as predictive of relapse. Catchpoole and colleagues examined these fourteen genes and found that the signature produced a separation in their cohort of ALL patients treated at The Children's Hospital at Westmead. However, the separation observed in this cohort was not associated with relapse nor treatment outcome.

To further explore this separation observed by both groups of investigators, Ho et al. (submitted) applied Random Forest to identify other probesets that further support this separation. The authors identified the 250 most important probesets that underlie this patient separation and found that the genes encompassed by these probesets are heavily involved in the cell cycle, mitosis, DNA replication, apoptosis and

DNA damage repair mechanisms. Our study will use these 250 probesets for further analysis by SVD and Expectation Maximisation clustering to explore their findings.

3 Results

3.1 Visualising KEGG data set

After transformation of the KEGG dataset with SVD we calculated the Pearson correlation between the data projected to principal components and to the association of GO terms to genes (i.e., \mathbf{X}), the total number of GO terms for each gene and the gene class. There was a very strong correlation of 0.995 between the data projected into principal component 1 (denoted as PC1 in this paper) and the number of terms associated with each gene suggesting that this principal component is a "size" component (Jolliffe 2004). It seems reasonable that the most variation in the dataset is based on the number of terms for genes.

Principal component 2, associated with the next largest variance, generally contrasts the genetic information processing genes with the carbohydrate metabolism genes as can be seen in Figure 1, where PC2 denotes the axis for principal component 2. However, we acknowledge that it is not a completely clear division: there is some overlap. The outlier (circled) with high PC2 and PC3 values is the gene *RHO* which is associated with the largest number of terms in the data. Table 2 shows that the highest correlation to PC2 is with the class label followed by strong positive correlations to GO terms describing carbohydrate metabolism and negative correlations to terms associated with ribosomes.

Apart from the outlier *RHO*, Figure 1 shows that PC3 separates the different kinds of genetic information processing genes as expected because there are more of these than the carbohydrate processing genes. Again, the separation involves some overlap between the classes.

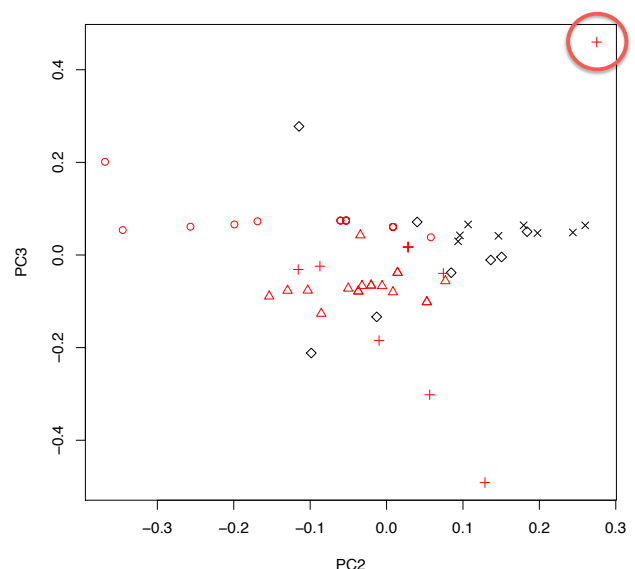


Figure 1: KEGG genes by PC2 and PC3. Ribosome \circ ; RNA polymerase \triangle ; transcription $+$; pentose phosphate pathway \times ; pentose/glucuronate interconversions \diamond . PC2 and PC3 are the axes for principal components 2 and 3 respectively.

Table 1: Genes in the KEGG dataset listed by class identifier. Column 1: class number and symbol in Figure 1. Column 2: KO terms describing class and associated genes.

Class	KO structure and list of genes used
1 ○	genetic information processing : translation : ribosome <i>rpsA, rpsB, rpsC, rpsD, rpsE, rpsF, rplB, rplC, rplD, rplE, rplF, RPS21, RPS23, RPS24, RPS25, rpmB, rpmC, rpmD, rpmE, rpmF</i>
2 △	genetic information processing : transcription : RNA polymerase <i>FLIA, RPOA, RPOB, RPOZ, RPOH, RPON, RPOD, RPB2, RPB1, RPB3, RPA49, RPA14, RPA34, RPA43, RPA12, RPC19, RPC25, RPB7, RPB4</i>
3 +	genetic information processing : transcription <i>GREa, GREB, NUSA, NUSB, NUSG, MBF1, Rcl1, RHO, ELP3, POL-RMT, gtf2a2</i>
4 ×	metabolism : carbohydrate metabolism : pentose phosphate pathway <i>pgl, zwf, edd, rpe, tktA, fbp, rpiA, gcd, rbsK, pgm, eda</i>
5 ◇	metabolism : carbohydrate metabolism : pentose and glucuronate interconversions <i>GUSB, galU, rpe, AKR1, mtlY, mtlD, clpX</i>

Table 2: GO term name and accession for terms with Pearson correlation > 0.5 to PC2 values for KEGG data. “Class” refers to the class identifier for the gene.

Term name and accession	Correlation
Class	0.550
Carbon utilization by utilization of organic compounds (GO:0015978)	0.539
Cellular catabolic process (GO:0044248)	0.539
Ribosome (GO:0005840)	-0.626
Ribonucleoprotein complex (GO:0030529)	-0.626
Intracellular (GO:0005622)	-0.606
Structural constituent of ribosome (GO:0003735)	-0.606
Translation (GO:0006412)	-0.606
Cytosolic small ribosomal subunit sensu Eukaryota (GO:0005843)	-0.577

3.2 Visualising cancer dataset

As with the KEGG dataset, there is a strong correlation between the number of GO terms associated with the genes and principal component 1 (PC1). The second, third and fourth PCs separate GO terms by their respective subontologies as shown in Figure 2. This suggests unsurprisingly that most of the variance in the dataset is based on technicalities rather than biological factors. Consequently we split the GO terms according to the three sub-ontologies and performed SVD on each individually.

Results for the Cellular Component GO terms, shown in Figure 3 (top) highlight two clusters of terms, separated along the PC3 axis. Pearson correlation between GO terms and the PCs (see Table 3) reveals that the separation between terms is associated with the cytoplasmic structure (e.g. GO:0005856 *cytoskeleton* and GO:0005874 *microtubule*) and DNA replication (e.g. GO:0031298 *replication fork protection complex* and GO:0042555 *MCM complex*).

For the Biological Process terms in Figure 3 (middle) PC2 separates terms associated with cell division (e.g. GO:0007067 *mitosis* and GO:0051301 *cell division*) from those related to DNA replication (cluster A). PC3 reveals a tight group of terms (cluster B in Figure 3 middle) associated with development (e.g. GO:0009790 *embryonic development* and GO:0030903 *notochord development*). See also Table 4.

For the Molecular Function terms in Figure 3 (bottom), PC2 shows a cluster of terms separate from the main grouping (cluster C) that is related to DNA helicase activity (see Table 5). Located in close prox-

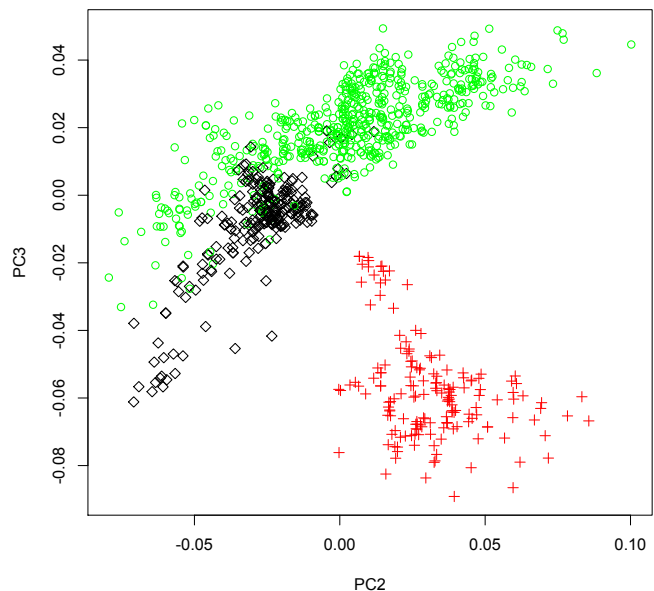


Figure 2: Plot of GO terms by PC2 and PC3 for cancer data. Terms labelled by sub-ontology: cellular component (red +), molecular function (black ◇) and biological process (green ○). PC2 and PC3 denote axes for principal components 2 and 3 respectively.

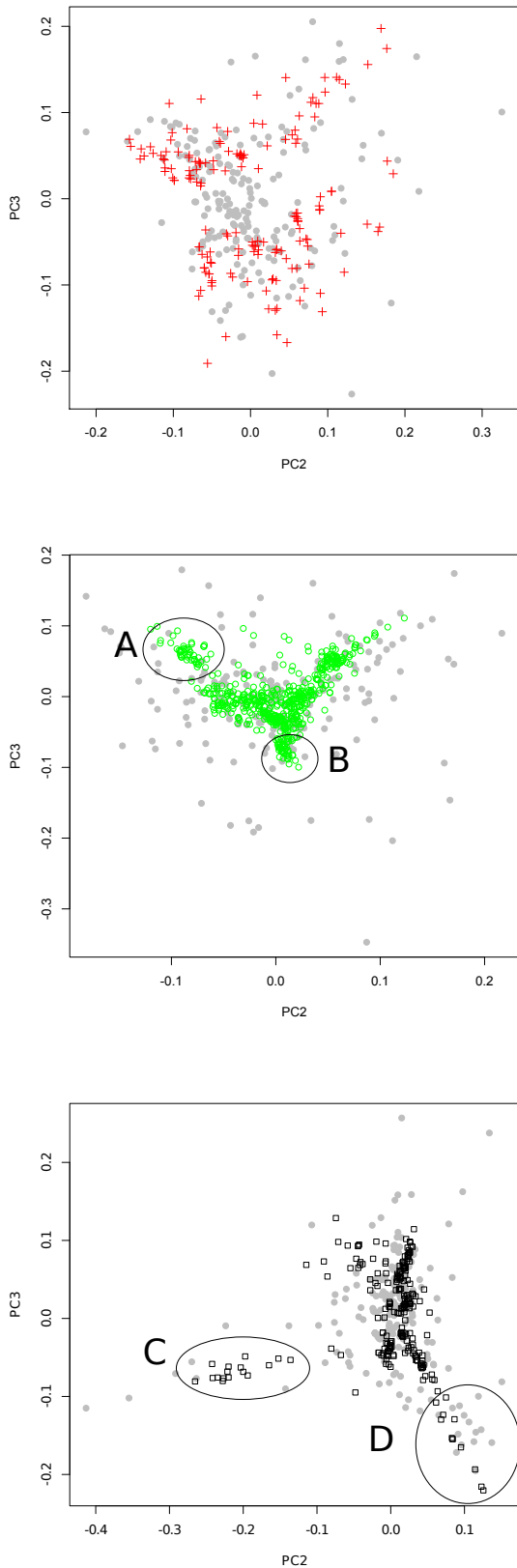


Figure 3: Plot of principal components 2 and 3 for **U** matrix (genes = •) and **V** matrix (terms: red + for cellular component, green o for biological process and black □ for molecular function) for the cancer dataset. Top: cellular component terms, Middle: biological process terms, Bottom: molecular function terms. Circled clusters A, B, C and D are described in the text.

imity is a loose cluster of six genes (the grey circles) that code for mini chromosome maintenance proteins (*MCM2–MCM7*). Both *MCM* and replicative helicase play integral roles in eukaryotic DNA replication: the *MCM* protein complex formed by *MCM2–MCM7* is involved in initiation (Costa & Onesti 2008) and replicative helicase is an enzyme that plays an role in unwinding the strands (Johnson et al. 2007). *MCM10*, the remaining *MCM* gene in the data is found in the main group of genes rather than the cluster. Whilst *MCM10* is involved in DNA replication (Chattopadhyay & Bielinsky 2007) and interacts with *MCM2–MCM7*, it is not part of the *MCM2–MCM7* family (Merchant et al. 1997) and our visualisation can highlight this.

Cluster D in Figure 3 (bottom) shows Molecular Function terms associated with kinase activity. This groups genes with roles in mitosis and, in particular, in mitotic spindle checkpoint signalling and includes *NEK2*, which has been shown to be an important protein in mitotic checkpoint signalling (Lou et al. 2004) and *BUB1*, which functions as a regulator of spindle assembly and has been shown to lead to aneuploidy in leukemic cells lines if mutated (Ru et al. 2002). Also within this cluster is thymidine kinase 1 (*TK1*), which has been reported to be predictive of remission duration (Jahns-Streubel et al. 1997) and relapse (Votava et al. 2007) in acute leukemias, and is essential to DNA synthesis.

SVD visualisation of the cancer data results in a meaningful functional visualisation of the genes, particularly when limited to terms in sub-ontologies. Clusters of terms highlight functional groupings of genes and the genes themselves cluster “behind” the terms that describe them. Correlations describe the PC axes. Each PC describes a different functional aspect of the gene set.

4 Conclusion

We applied SVD to lists of genes augmented with GO terms and inter-term similarities. Two datasets were visualised: validation data from KEGG and a set of genes identified experimentally. Results showed that principal component 1 measured the number of terms associated with genes. Later components allowed visualisation of genes according to their functional information, but the meaning of PCs varied depending on the underlying genes. For the KEGG data PCs described gene functionality. For the larger cancer dataset the early PCs simply identified known hierarchies. Separate visualisation using terms from the individual subontologies was more informative. Correlation between GO terms and PCs improved understanding of the functional meaning of the PCs. These results show that our approach can bring meaningful biological interpretation to gene lists.

We plan to explore other similarity measures, specifically an information-theoretic one (Speer et al. 2005). We will address the bias to genes with many terms by applying methods based on local distance measures. However, unlike the methods in this paper, those methods require parameter tuning, which in turn requires investigation of how to decide whether one visualisation is “better” than another. This will also involve comparing the visualisations derived using our approach more widely with other state-of-the-art methods. Variability of the quality of information throughout GO is an issue and we plan to investigate ways to deal with this.

We acknowledge that interpretation of our results is somewhat subjective. This is a problem gener-

Table 3: GO terms from the cellular component sub-ontology with absolute value of Pearson correlation > 0.35 for PC1–4 values from the cancer data set.

PC	GO term name and accession	Correlation
1	Number of terms	0.855
2	GO:0000777 (condensed chromosome kinetochore)	0.547
	GO:0000775 (chromosome, centromeric region)	0.503
	GO:0000776 (kinetochore)	0.466
	GO:0000778 (condensed nuclear chromosome kinetochore)	0.427
3	GO:0005856 (cytoskeleton)	0.465
	GO:0005874 (microtubule)	0.418
	GO:0005819 (spindle)	0.386
	GO:0031298 (replication fork protection complex)	-0.376
	GO:0042555 (MCM complex)	-0.359
4	GO:0005737 (cytoplasm)	0.383
	GO:0005730 (nucleolus)	0.372
	GO:0005634 (nucleus)	0.365

Table 4: GO terms from the biological process sub-ontology with absolute value of Pearson correlation > 0.35 for PC1–4 values for the cancer data set.

PC	GO term name and accession	Correlation
1	Number of terms	0.950
2	GO:0007067 (mitosis)	0.672
	GO:0051301 (cell division)	0.665
	GO:0007049 (cell cycle)	0.438
	GO:0006260 (DNA replication)	-0.498
3	GO:0009790 (embryonic development)	-0.353
4	GO:0006281 (DNA repair)	0.588
	GO:0006974 (response to DNA damage stimulus)	0.445
	GO:0000724 (double-strand break repair)	0.388
	GO:0006350 (transcription)	-0.488
	GO:0045449 (regulation of transcription)	-0.487

Table 5: GO terms from the molecular function sub-ontology with absolute value of Pearson correlation > 0.35 for PC1–4 values for the cancer data set.

PC	GO term name and accession	Correlation
1	Number of terms	-0.872
2	GO:0043140 (ATP-dependent 3'-5' DNA helicase activity)	-0.604
	GO:0003678 (DNA helicase activity)	-0.575
	GO:0004003 (ATP-dependent DNA helicase activity)	-0.574
	GO:0009378 (four-way junction helicase activity)	-0.529
	GO:0003697 (single-stranded DNA binding)	-0.562
3	GO:0016301 (kinase activity)	-0.565
	GO:0004672 (protein kinase activity)	-0.533
	GO:0004674 (threonine kinase activity)	-0.571
4	GO:0004518 (nuclease activity)	0.670
	GO:0004527 (exonuclease activity)	0.650
	GO:0004523 (ribonuclease H activity)	0.589
	GO:0008409 (5'-3' exonuclease activity)	0.557

ally with visualisation and unsupervised learning. We plan to investigate more informative and objective approaches to characterising clusters than simple Pearson correlation that can also take into account the level of GO terms in the hierarchies.

References

- Ashburner, M. et al. (2000), ‘Gene Ontology: tool for the unification of biology. The Gene Ontology Consortium.’, *Nature Genetics* **25**(1), 25–9.
- Backes, C. et al. (2007), ‘GeneTrail–advanced gene set enrichment analysis’, *Nucleic Acids Research* **35**(suppl 2), W186–W192.
- Berriz, G. & Roth, F. (2008), ‘The Synergizer service for translating gene, protein and other biological identifiers’, *Bioinformatics* **24**(19), 2272.
- Berriz, G. et al. (2009), ‘Next generation software for functional trend analysis’, *Bioinformatics* **25**(22), 3043.
- Catchpoole, D. et al. (2008), ‘Predicting outcome in childhood acute lymphoblastic leukemia using gene expression profiling: Prognostication or protocol selection?’, *Blood* **111**(4), 2486.
- Chattopadhyay, S. & Bielinsky, A. (2007), ‘Human Mcm10 regulates the catalytic subunit of DNA polymerase- α and prevents DNA damage during replication’, *Molecular Biology of the Cell* **18**(10), 4085.
- Costa, A. & Onesti, S. (2008), ‘The MCM complex: (just) a replicative helicase?’, *Biochemical Society Transactions* **36**, 136–140.
- Flotho, C. et al. (2007), ‘A set of genes that regulate cell proliferation predicts treatment outcome in childhood acute lymphoblastic leukemia’, *Blood* **110**(4), 1271.
- Fröhlich, H. et al. (2007), ‘GOSim–An R-package for computation of information theoretic GO similarities between terms and gene products’, *BMC Bioinformatics* **8**, 166.
- Golub, G. & Van Loan, C. (1996), *Matrix computations*, Johns Hopkins University Press.
- Huang, D. et al. (2007), ‘The DAVID Gene Functional Classification Tool: a novel biological module-centric algorithm to functionally analyze large gene lists’, *Genome Biology* **8**(9), R183.
- Huang, D. et al. (2008), ‘Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists’, *Nucleic Acids Research* **37**(1), 1–13.
- Jahns-Streubel, G. et al. (1997), ‘Activity of thymidine kinase and of polymerase alpha as well as activity and gene expression of deoxycytidine deaminase in leukemic blasts are correlated with clinical response in the setting of granulocyte-macrophage colony-stimulating factor-based priming before and during TAD-9 induction therapy in acute myeloid leukemia’, *Blood* **90**(5), 1968–1976.
- Johnson, D. et al. (2007), ‘Single-molecule studies reveal dynamics of DNA unwinding by the ring-shaped T7 helicase’, *Cell* **129**(7), 1299–1309.
- Jolliffe, I. T. (2004), *Principal Component Analysis*, second edn, Springer.
- Jones, S. et al. (2008), ‘Core signaling pathways in human pancreatic cancers revealed by global genomic analyses’, *Science* **321**(5897), 1801–1806.
- Kanehisa, M. et al. (2008), ‘KEGG for linking genomes to life and the environment’, *Nucleic Acids Research* **36**, 480–484.
- Lee, S. et al. (2004), ‘A graph-theoretic modeling on GO space for biological interpretation of gene clusters’, *Bioinformatics* **20**(3), 381–388.
- Lou, Y. et al. (2004), ‘NEK2A interacts with MAD1 and possibly functions as a novel integrator of the spindle checkpoint signaling’, *Journal of Biological Chemistry* **279**(19), 20049.
- Mathur, S. & Dinakarpanian, D. (2007), ‘A New Metric to Measure Gene Product Similarity’, *Bioinformatics and Biomedicine, 2007. BIBM 2007. IEEE International Conference on* pp. 333–338.
- Merchant, A. et al. (1997), ‘A lesion in the DNA replication initiation factor Mcm10 induces pausing of elongation forks through chromosomal replication origins in *Saccharomyces cerevisiae*’, *Molecular and Cellular Biology* **17**(6), 3261.
- Mistry, M. & Pavlidis, P. (2008), ‘Gene ontology term overlap as a measure of gene functional similarity’, *BMC Bioinformatics* **9**(1), 327.
- Popescu, M. et al. (2004), Functional summarization of gene product clusters using Gene Ontology similarity measures, in ‘Proceedings of IEEE Intelligent Sensors, Sensor Networks and Information Processing Conference’, IEEE, pp. 553–558.
- Richards, A. et al. (2010), ‘Assessing the functional coherence of gene sets with metrics based on the Gene Ontology graph’, *Bioinformatics* **26**(12), i79.
- Ru, H. et al. (2002), ‘hBUB1 defects in leukemia and lymphoma cells.’, *Oncogene* **21**(30), 4673–4679.
- Sanfilippo, A. et al. (2007), ‘Combining Hierarchical and Associative Gene Ontology Relations With Textual Evidence in Estimating Gene and Gene Product Similarity’, *IEEE Transactions on Nanobioscience* **6**(1), 51–59.
- Sheehan, B. et al. (2008), ‘A relation based measure of semantic similarity for gene ontology annotations’, *BMC Bioinformatics* **9**(1), 468.
- Speer, N. et al. (2005), Functional grouping of genes using spectral clustering and gene ontology, in ‘Proceedings of the IEEE International Joint Conference on Neural Networks’, pp. 298–303.
- Tomfohr, J. et al. (2005), ‘Pathway level analysis of gene expression using singular value decomposition’, *BMC Bioinformatics* **6**(1), 225.
- Votava, T. et al. (2007), ‘Changes of serum thymidine kinase in children with acute leukemia’, *Anticancer research* **27**(4A), 1925.
- Yi, G. et al. (2007), ‘Identifying clusters of functionally related genes in genomes’, *Bioinformatics* **23**(9), 1053.

A non-time series approach to vehicle related time series problems

Jonathan R. Wells¹, Kai Ming Ting¹ and Chandrasiri P. Naiwala²

¹Gippsland School of Information Technology
Monash University, Australia
Email: {jonathan.wells,kaiming.ting}@monash.edu

²Toyota InfoTechnology Center Co., Ltd., Japan
Email: np-chandrasiri@jp.toyota-itc.com

Abstract

This paper shows that some time series problems can be better served as non-time series problems. We used two unsupervised learning anomaly detectors to analyse a vehicle related time series problem and showed that non-time series treatment produced a better outcome than a time series treatment. We also present the benefits of using unsupervised methods over semi-supervised or supervised learning methods, and rule-based methods.

1 Introduction and Motivation

Time series data treatments rely on the relationship between the points which are in a sequential time order; whereas, non-time series data treatments treat each point independently. Time series data conform to a natural ordering and the data indices often appear at a regular time step interval. Examples of time series applications are stock market, tracking an outbreak of a disease, a heart monitor or weather time series.

However, do all data, with a natural ordering based on time, need to be treated as a time series problem? In this paper, we look at a vehicle related time series problem to examine whether it can be solved as a non-time series problem.

We propose to use two anomaly detectors to solve this problem in a non-time series setting rather than in a time series setting. This paper shows that:

(i) A vehicle related time series problem is better treated as a non-time series problem and it can be effectively and efficiently solved using unsupervised learning anomaly detectors.

(ii) Rule-based approach, currently used in a non-time series setting, produces a set of rules in the form of a fixed linear model where some parameters modify the decision globally in the feature space. Any methods (e.g., McLaughlin et al., 2009; Knipling et al., 1993; Kiefer et al., 1999; Brunson et al., 2002) using this approach either cannot make or have difficulty in making these changes locally. The proposed approach can easily retrain a new model to cater for a new situation that has local changes only.

The key advantage of the proposed approach is that a time series problem becomes a simpler problem when treated as a non-time series problem. As

a result, a simpler model can be used to solve this problem. The two anomaly detectors we employed, *iForest* (Liu, Ting, and Z.-H. Zhou, 2008) and *ORCA* (Bay and Schwabacher, 2003), provide further advantage in providing flexibility in retraining a new model to suit different situations and users.

A vehicle related time series problem has all the characteristics of a time series problem with one unique property: a projection of any given data point can be determined, using the law of physics, by assuming that there are no further changes to the current actions between the driver and the approaching object. With this property, we can determine if there is a potential collision and issue an alert when require. This naturally leads to a non-time series analysis.

In this paper, features such as weather or road conditions are outside the scope because the main focus is to illustrate that a vehicle related time series problem could be solved as a non-time series problem.

We review related work in the next section. Section 3 describes the relationship between time series and non-time series treatments for a vehicle related time series problem. Section 4 provides a brief description of two anomaly detectors we employed to solve a vehicle related time series problem. Section 5 outlines the data sets and the feature selection process, and the type of models generated for evaluation. This is followed by the empirical evaluation in Section 6. We provide the conclusions in the last section.

2 Related Work

This section reviews two existing approaches to vehicle related time series problems. The first approach is represented by non-time series rule-based methods while the second approach is represented by time-series semi-supervised methods.

McLaughlin et al. (2009) evaluated three different collision avoidance systems to prevent rear-end crashes using a subset of the 100-car study (Neale et al., 2002; Dingus et al., 2006). The three different algorithms are rule-based methods using the time series data in a non-time series setting that treats each time step as an independent data point. Up to four sensor readings (range, speed, acceleration, and relative velocity) of the ‘following’ vehicle and up to two computed values (acceleration and velocity) for the ‘leading’ vehicle are used in each of the methods.

The first method, Knipling (Knipling et al., 1993), is a straight rule-based method with a constant allowing for the driver reaction time plus the braking time. The second method, CAMP Linear (Kiefer et al., 1999), adds to the model a set of coefficients which are derived from a regression analysis to accommodate different driver characteristics. The final

	Rule-based	Non-time series - unsupervised	Time series - semi-supervised
feature space	Use the original feature space		Feature space transformation
labels	No labels required		Labels are needed for some time series events
model	No model is built; cannot be retrained.	Model is built and it can be retrained when needed.	
complexity	Simplest because there are no models requiring to be built. Just plug in the values into a pre-defined model.	Simpler. Build a model using existing anomaly detectors and predictions are made from the model.	Complex as a new feature space needs to be constructed before building a model; and the model is non-linear and requires complex learning procedures.

Table 1: Differences between rule-based, non-time series unsupervised and time series semi-supervised methods

method, NHTSA (Brunson et al., 2002) which incorporates both the above features, has an additional feature that allows a driver to set different alert sensitivity settings.

The first two methods compute a warning range and compare this value to the actual sensor range value. An alert is issued when the actual range falls below the computed range. The third method computes a distance for the following vehicle to avoid a collision with the leading vehicle. This is combined with a threshold computed based on the velocity of the following vehicle. If the computed distance is less than the computed threshold then an alert is issued.

However, all these methods are ‘hard-coded’ linear model and do not allow for the changes in driver characteristics over time apart from the three different alert sensitivity levels in NHTSA.

Ning et al. (2010) presented a semi-supervised time series approach where each time series event is labelled ‘crash’ (if it actually happened) or ‘safe’ (if there are no crashes during the whole event). From a transformed feature space, a temporal difference learning (a form of reinforcement learning) is used to learn a ‘danger level’ function from training events, which exhibits low values if the vehicle approaches a crash point, and high values for safe events. A threshold can be used to trigger an alert if the output of the ‘danger level’ function is lower than the threshold. Their proposed non-linear method is found to perform better than a linear method, logistic regression and linear regression.

However, training a non-linear model is computationally expensive. Wang, Zhu, and Gong (2010) attempted to overcome the long training time by using faster parameter updating schemes instead. Otherwise, their approach is similar to Ning et al. (2010) in that both transformed the feature space in a similar manner and then modelled a danger level function from training events with two ‘known’ states.

The advantage of our proposed method over the rule-based methods are that i) we use an unsupervised learning method to train a model for prediction; and ii) the model can be retrained to allow for changes of driver characteristics over time. The advantages over the semi-supervised methods are that the problem is treated as non-time series and no labels are required which make the problem simpler. Table 1 summaries these advantages.

3 Treating time series as non-time series

The vehicle related time series problem (Neale et al., 2002; Dingus et al., 2006) that we investigated contains a number of driving sessions. Each session is a time series recording of the vehicle’s states and any approaching objects detectable by the radars installed in the vehicle. A session starts from the time that a driver begins a journey until the vehicle has stopped

due to either an unforeseen circumstance, such as a crash, or the driver has reached their destination. Each driving session is labelled: near crash or crash (if they occurred), or an incident-free event.

Although time series analysis had been used in the vehicle related time series problem (Ning et al., 2010; Wang, Zhu, and Gong, 2010), it can be treated as non-time series problem because the law of physics govern the vehicle and any impending objects at each time step during the driving session. This allows us to utilise the data in a non-time series setting because we can determine what the outcome would be at any data point, independent of other data points. Although we need to make an assumption that all conditions of that point remain constant, the calculated outcome is still valid to allow a vehicle warning system to issue an alert if there is an impending collision.

As such, every point in the time series is treated as an independent point using the law of physics; and it can be categorised into one of the following three labels: (i) ‘unsafe’ as a driver has minimal or no time to react to any impending collision between the vehicle and an approaching object, (ii) ‘safe’ as a driver has plenty of time to avert a crash, or (iii) ‘alert’ to avoid a crash if a driver is given a warning on time to take actions. We use a simple rule to define these three regions in a feature space in which a vehicle can be, in relation to an approaching object. The following two equations are used.

$$T = \frac{d}{v} \quad (1)$$

$$\mathcal{T} = \frac{v}{a} + c \quad (2)$$

where T is the time to impact between the vehicle and an approaching object; d and v are the distance and the relative velocity, respectively, between the vehicle and the object; $\frac{v}{a}$ is the vehicle deceleration time with the assumption that a certain deceleration rate a has been applied (due to timely braking of the vehicle). \mathcal{T} is the minimum time required to avoid a crash. In order to avoid a crash, T must be greater than \mathcal{T} .

c in Equation 2 is the driver’s response time to brake in order to avoid any potential collisions. It consists of the driver reaction time, R , plus an *extra* alert time W . The driver’s response time is expressed as follows:

$$c = W + R \quad (3)$$

Solve for v using equations 1 and 2 gives:

$$v = \frac{a(\sqrt{c^2 + \frac{4d}{a}} - c)}{2} \quad (4)$$

Using equation 4, we can plot different curves defining the boundaries between regions of potentially

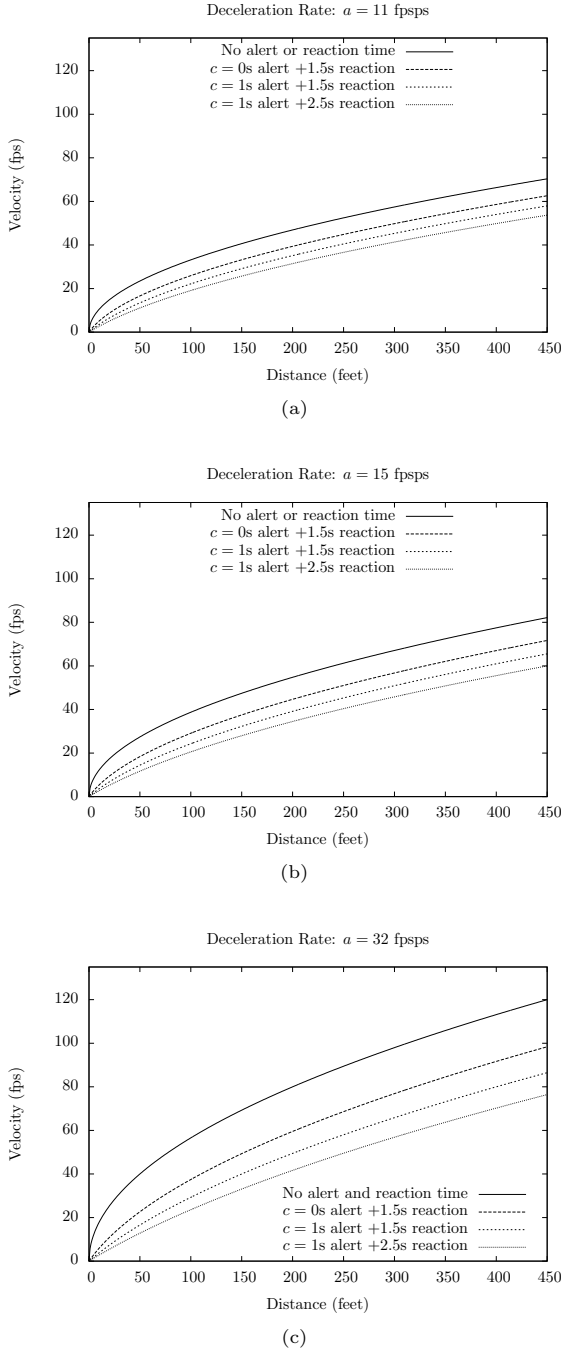


Figure 1: Using equation 4, these charts show the different values for a , c , and d . The different a values represents the different deceleration rates for the driver ability to stop a vehicle. 11 fpsps is a conservative figure. 15 fpsps is the general figure. 32 fpsps is the figure for professional car racing drivers.

‘unsafe’ and ‘safe’ regions. Figure 1 shows the different curves for driver’s ability to stop a standard vehicle and their response times in a given event. The figures show the effect of using different values for a and c (*A Policy on Geometric Design of Highways and Streets* 2004; McLaughlin et al., 2009). The region above the top curve is the ‘unsafe’ region. For different c values, the region below the ‘ c ’ curve is the ‘safe’ region; and the region between the ‘ c ’ curve and the top curve, is the alert region. A professional racing driver will have the ability to stop a vehicle much quicker than an average driver; therefore, has

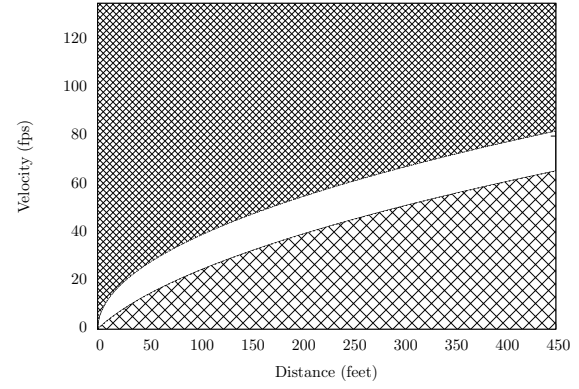


Figure 2: The different zones as determined by the simple rule. The hashed zone, at the bottom, signifies the safe region. The white zone signifies the alert region which also contains the driver reaction time. The hashed zone, at the top, signifies the unsafe region.

a greater ‘safe’ region than an average or beginner driver. This is reflected in the graphs.

The following constants are used to derive a simple rule. Deceleration rate a is set to 15 fpsps (feet per second per second) as outlined in *A Policy on Geometric Design of Highways and Streets* (2004) and the driver reaction time, R , is set to 1.5 seconds which represents the reaction time for 75% of the population (McLaughlin et al., 2009). An alert time, W , of 1 second is also set. Figure 2 shows the zones that were created by the simple rule. The hashed zone, at the top, is the area to be considered ‘unsafe’ according to the rule if there are no further changes to the driver’s current action or the approaching object. The white zone is the area where an alert will be given and the hashed zone, at the bottom, shows the area where it is considered ‘safe’.

The simple rule uses the unit measure of ‘time’ to determine whether a collision is imminent; whereas, the three non-time series methods (Knippling et al., 1993; Kiefer et al., 1999; Brunson et al., 2002), described in Section 2, use the unit measure of ‘distance’ to determine whether a collision is imminent. Both measures, time and distance, are derived using the same law of physics.

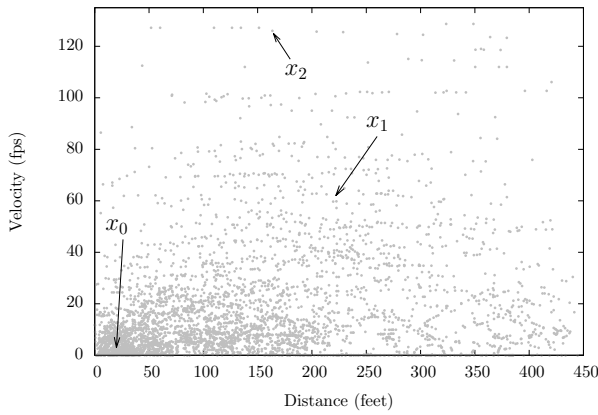
4 Unsupervised learning approach

In this research, we used two anomaly detectors called *iForest* (Liu, Ting, and Z.-H. Zhou, 2008) and *ORCA* (Bay and Schwabacher, 2003). We provide a brief description of each of these detectors in the following two sections.

4.1 *iForest*

Anomalies can be characterised by two quantitative properties: the number of anomaly points is small and these points have values that are significantly different to those of normal points (Liu, Ting, and Z.-H. Zhou, 2008).

Traditionally, anomaly detectors model the profile of the normal points and then identify points that do not conform to the normal profile as anomalies. However, these detectors are optimised for normal points and not for anomalies. Liu, Ting, and Z.-H. Zhou (2008) introduced a different kind of anomaly



(a) Showing 20% of the crash data reported in Section 5. x_0 is in the lower left region; x_1 is in the middle of the data cloud; x_2 is at the edge of the data cloud.

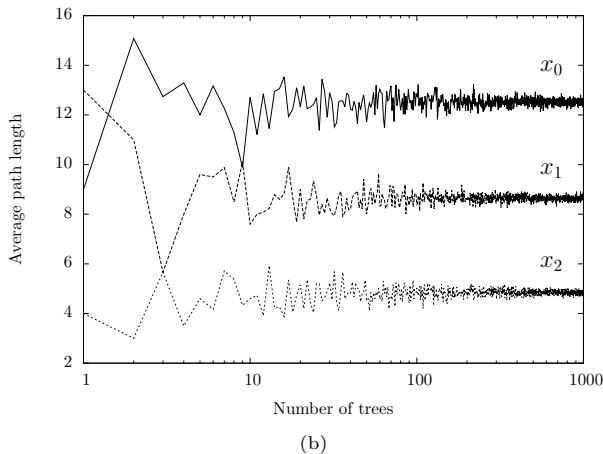


Figure 3: The result of isolating x_0 , x_1 and x_2 using a number of *i*Trees. The average path length required to isolate x_0 , x_1 and x_2 converged to 12.7, 8.5 and 4.8, respectively.

detector called *Isolation Forest* or *iForest* that isolates anomalies from normal points rather than modelling normal points. Given a data cloud, Liu, Ting, and Z.-H. Zhou (2008) show that anomalies can be isolated faster than normal points because anomalies are more susceptible to isolation than normal points.

The isolation mechanism using partitioning can be implemented using a random tree called *Isolation Tree* or *iTree*. *iTree* is a binary tree that is constructed as follows. An attribute is randomly selected at each node; then a random split point is chosen that divides the data space into two sub-regions. This is repeated until every point is isolated from the rest of the points. Points that are isolated quickly will appear at the top of the *iTree*. Therefore, anomalies are defined as those points that have shorter average path lengths than normal points for a given set of *i*Trees.

To provide an example, Figure 3 shows that the average path length required to isolate x_0 , x_1 and x_2 , from data showed in Figure 3(a). The average path length converged to 12.7, 8.5 and 4.8, respectively, as the number of *i*Trees increases; where x_0 is a point at the middle of a data cloud; x_1 is in the middle ring and x_2 is at the fringe.

iForest is an unsupervised learning method that constructs an ensemble of *i*Trees. The parameters required are: number of *i*Trees in the ensemble and

the training subsample size used to build each *iTree*. Note that each *iTree* can be trained using a subsample size significantly smaller than the given data set. This allows *iForest* to be built fast and makes *iForest* one of the fastest anomaly detectors available. For ease of reference, the algorithms used to build an *iForest* are re-produced from Liu, Ting, and Z.-H. Zhou (2008) in the Appendix.

4.2 ORCA

The second anomaly detector we used is a k -nearest neighbour based anomaly detector called ORCA (Bay and Schwabacher, 2003). The algorithm uses a pair of nested loops with a pruning rule. This pruning rule is designed to speed up the nearest neighbour search by removing data points which fall below a threshold. It uses a distance metric to find the k nearest neighbours and then it computes an anomaly score to evaluate each point. The anomaly score can be any nearest neighbour based function such as the distance to the k th nearest neighbour or the average distance of the k nearest neighbours. Readers are referred to Bay and Schwabacher (2003) for other details of the algorithm.

4.3 Applying anomaly detectors to vehicle related time series problems

In applying a batch mode anomaly detector to a vehicle related time series problem, we must first represent it as a non-time series problem where every point is independent of each other. We can then use an anomaly detector to identify rare events, such as crash or near crash, as anomalies; and common no-incident events as normal points. Here, we hypothesise that drivers will be in the no-incident events most of the time and will be in the crash or near crash events a few times and far between events. This hypothesis fits in with the definition of what anomalies are.

However, for this type of application, we can also use the anomaly detector in a slightly different way. Here, the emphasis is not detecting anomalies but to define boundaries between unsafe and safe regions as defined in Figure 2.

4.4 Semi-supervised versus unsupervised

The type of anomaly detectors we explored in this paper are different from those employed in the literature e.g., Wang, Zhu, and Gong (2010); Y. Zhou et al. (2007); Ning et al. (2010). The existing methods mainly employ supervised and semi-supervised learning methods which require labelled data. The type of anomaly detectors we explore are unsupervised learning methods which do not require labelled data. The labels defined in Section 3 (provided by the simple rule) are required for verification and computation of detection performance measure only, and they are not used to train *iForest* and ORCA.

We are unable to compare with semi-supervised learning approaches described in Ning et al. (2010) and Wang, Zhu, and Gong (2010) because neither their software nor the data sets used are available.

5 Data sets used and their characteristics

5.1 Data and attribute selection

We use two existing data sets from the Virginia Tech Transport Institute (VTTI)¹. These data sets were collected from a study of approximately 100 vehicles

¹<http://www.vtti.vt.edu/>

Radar	# of crashed events	# of near crashed events
Front	57 (84%)	644 (85%)
Rear	61 (90%)	688 (91%)
Either	66 (97%)	754 (99%)

Table 2: The values represent the number of events that the radars are active. ‘Either’ indicates that one of the radars is active for a given event. Each figure in bracket is the percentage of events from the total of 68 crash events or 760 near crash events.

driven by 241 primary and secondary drivers who had driven approximately 3.2 million vehicle kilometres in approximately 43,000 hours over a period of twelve months (Neale et al., 2002; Dingus et al., 2006).

The first data set consists of 68 crash events and the second data set consists of 760 near crash events. Each event contains a snapshot of approximately forty seconds; and it is broken down into thirty seconds before the crash (or near crash) and ten seconds after. The data is made up of three distinct sections: sensors, video and the manually added information. The sensor data consists of seven components: pedals (accelerator and brake), indicators (left and right), motion (lateral, longitudinal and yaw), lane tracking, radar (front and rear), light intensity and GPS. The video data consists of the recording from five cameras in the vehicle cabin showing the four different views outside the vehicle and a single view of the driver’s hands and feet, steering wheel and the instrument panels. The manual data is added by an analyst studying the video footages and recording what the driver was doing at the time of the event, the conditions of the road, traffic, weather, time of day (day or night time) and analyst’s analyses.

The focus of our research is on the sensor data. The radar data appears to be the best component for this research because it contains all of the necessary information required to determine whether there will be any impending crashes. The radar data consists of two streams of data: the front radar for tracking objects in front of the vehicle and the rear radar for tracking objects behind the vehicle. Each radar is capable of tracking up to seven objects to a distance of approximately one hundred metres. Each radar provides the following measurements relative to the vehicle: the range to the object (distance), the rate of change (relative velocity) and azimuth (angle).

We analysed the data sets to see if we have enough data. The result of the initial analysis on the radar data are summaries in Table 2. For this research, we can use over 97% of the available crash event data and over 99% for near crash event data. The remaining two crash and six near crash events were detected by other means such as an analyst studying the video. It should be pointed out that only one of the two radars need to be active in order to predict an impending collision provided that the approaching object is in the radar zone of detection.

5.1.1 Non-Time Series Treatment

The non-time series treatment employs two attributes: distance and relative velocity. It treats every point as independent.

5.1.2 Time Series Treatment

For the time series analysis, we constructed a new feature space from the sensors data using the method

100-Car Study	Wang, Zhu, and Gong (2010)
lane distance - left	driver’s lateral lane position
lane distance - right	
vehicle composite speed	driver’s longitudinal velocity
yaw rate	steering angle
longitudinal acceleration and brake = off	longitudinal acceleration due to throttle
longitudinal acceleration and brake = on	longitudinal acceleration due to brake
radar range - forward	minimum range - opposite direction
	minimum range - same direction
radar range - rear	
not available	throttle depression fraction
not available	braking depression fraction

Table 3: Corresponding attributes between the 100-car study (Neale et al., 2002; Dingus et al., 2006) and the study by Wang, Zhu, and Gong (2010).

as outlined by Wang, Zhu, and Gong (2010). This involved computing the relationship between two consecutive data point and then constructing a ‘sliding-window’ on the computed results. Finally, a series of statistical analysis are conducted for each window which produce a set of statistical features.

The selection of attributes is based on the method described by Wang, Zhu, and Gong (2010). They selected nine attributes from a set of thirty eight attributes: lane position, longitudinal acceleration and deceleration, longitudinal velocity, steering angle, throttle and brake depression fraction, and the closest object approaching the front or rear of the vehicle. Table 3 lists the corresponding attributes between the 100-car study (Neale et al., 2002; Dingus et al., 2006) and the study by Wang, Zhu, and Gong (2010). All of the attributes except throttle and brake depression fractions are used. The lane position is separated into two attributes: left and right side lane marking details. This produced eight available attributes to analyse the two data sets.

These eight attributes are then transformed from the original feature space into a new feature space, as in Wang, Zhu, and Gong (2010). For each of the eight attributes, four different attributes are constructed as follows: the original value (f), the first-order forward difference (Δf , f^2 , and Δf^2). This produces a set of thirty two attributes for one time step. A sliding window with a length of 10 time steps is then applied. For each window, the following statistical information are calculated on each of the thirty two attributes: minimum, maximum, mean, and standard deviation. This produces the final set of 128 attributes for each window.

5.2 Data preparation

The crash data set contains 42,098 individual points across 68 events which has a total of 28,962 time steps². This data set is checked for any abnormalities. A total of 138 points are removed because of negative distance measurements; and one point has an impossible velocity reading. Since we are only interested in points that are approaching the vehicle, 20,068 points that have objects moving away from the vehicles are also removed. This leaves 21,891 points to be used for the non-time series data treatment. We used all of the 28,962 time steps for the time series data treat-

²A radar can detect up to 7 objects simultaneously for a single time step. Each object, in the time step, is treated as a single point.

Data	Format	Labels			Not assigned	Total
		Safe	Alert	Unsafe		
Crash	Non-Time Series	16,684	3,122	2,085	15,059	21,891
	Time Series	9,278	2,487	1,526		28,350
Near Crash	Non-Time Series	418,633	46,178	47,138	126,169	511,949
	Time Series	158,587	25,492	25,188		335,436

Table 4: Number of instances in each of the labels as described in Section 5.3.

Model	Data Size	Number of Attributes		Notes
		Radar	New feature space	
1	21,891	2		Attribute used are distance and relative velocity between the object and vehicle.
2	51,891	2		Same as Model 1 plus a Gaussian distribution, with a mean at $x = 450, y = 0$ and variance of 1, consisting of 30,000 points.
A	28,350		128	The 128 attributes are derived from the 8 attributes in the original feature space.
B	28,350		10 - <i>i</i> Forest / 5 - ORCA	Number of attributes selected using Forward Greedy Search from the available 128 attributes.
C	28,350		6	The best six attributes selected by Wang, Zhu, and Gong (2010, shown in Table III).

Table 5: Training data description. Attributes in the original feature space are the attributes from the 100-car study. Attributes in the new feature space are derived from the attributes as described in Section 5.1.2.

ment³ which includes objects moving away from the vehicle in order to maintain the time sequence.

For the near crash data set, there are 897,631 individual points across 760 events which has a total of 342,276 time steps. 13,396 are removed because of negative distance; 12 points are removed because of impossible velocity readings; and 372,274 points are removed for objects moving away from vehicle. This leaves a balance of 511,949 for the non-time series data treatment. All of the 342,276 time steps are used for the time series data treatment for the same reason given above for the crashed data.

5.3 Label assignments

This section describes how the labels are assigned for the purpose of verification and computation of detection performances.

5.3.1 Non-time Series

There are three labels: *unsafe*, *alert*, and *safe*. The individual labels are derived from using the simple rule (equation 4) with the following parameters. The boundary between unsafe and alert is defined as $c = 0$ and $a = 15$ fpsps (feet per second per second). The boundary between alert and safe is defined as $c = 2.5$ seconds and $a = 15$ fpsps as shown in Figure 2.

5.3.2 Time Series

A label is assigned to each time step using distance and relative velocity between the vehicle and the nearest approaching object using the same equation and parameters as in the non-time series treatment with one exception, i.e., no label is assigned if there are no objects being tracked for that particular time step.

Table 4 shows the final class assignments for both the crash and near crash data sets.

³Actually, this figure is reduced to 28,350 because of the sliding window effect. This is because each window has a length of 10 points in our setting, and any windows that does not have 10 points are discarded. This happens at the end of each event ($68 * 9 = 612$). Note that the near crashed data has the same effect.

6 Empirical Evaluations

The aim is to assess the utility of the time series treatment and the non-time series treatment. The evaluation assesses three different time series models in the new feature space: Model A is constructed using all of the 128 attributes; Model B is the best model using a subset of attributes chosen from a Forward Greedy Search; and Model C is constructed by using the best attributes as outlined by Wang, Zhu, and Gong (2010). Two non-time series models are constructed with the two attributes obtained from the radars. Model 1 is constructed using the distance and relative velocity between the vehicle and an approaching object. The current distribution of Model 1 is biased because the data sets used consists of the crash or near crash events only. This does not represent the true distribution because most events has majority of the points in the safe region. Model 2 is constructed by adding a Gaussian distribution, consisting of 30,000 synthetic points, to the lower right corner of the original data space. The details of all five models are summarised in Table 5.

*i*Forest and ORCA are employed to generate each of the above five models to assess the relative performance among the five models and between *i*Forest and ORCA.

6.1 *i*Forest's ranking capability

Figure 4 shows *i*Forest's ranking results of the crash non-time series data. Figure 4(a) shows the result of top rankings which includes the first 500 unsafe points above the top curve. This ranking includes 76 safe points below the bottom curve and 16 alert points between the top and bottom curves which are ranked higher than the top 500th ranked unsafe point. Each of the following figures, 4(b) to 4(d), contains the next set of top rankings which includes the next set of 500 unsafe points along with alert and safe points which were ranked higher than the corresponding 500th ranked unsafe points. The final figure, 4(e), shows the remaining 85 unsafe points along with the other points.

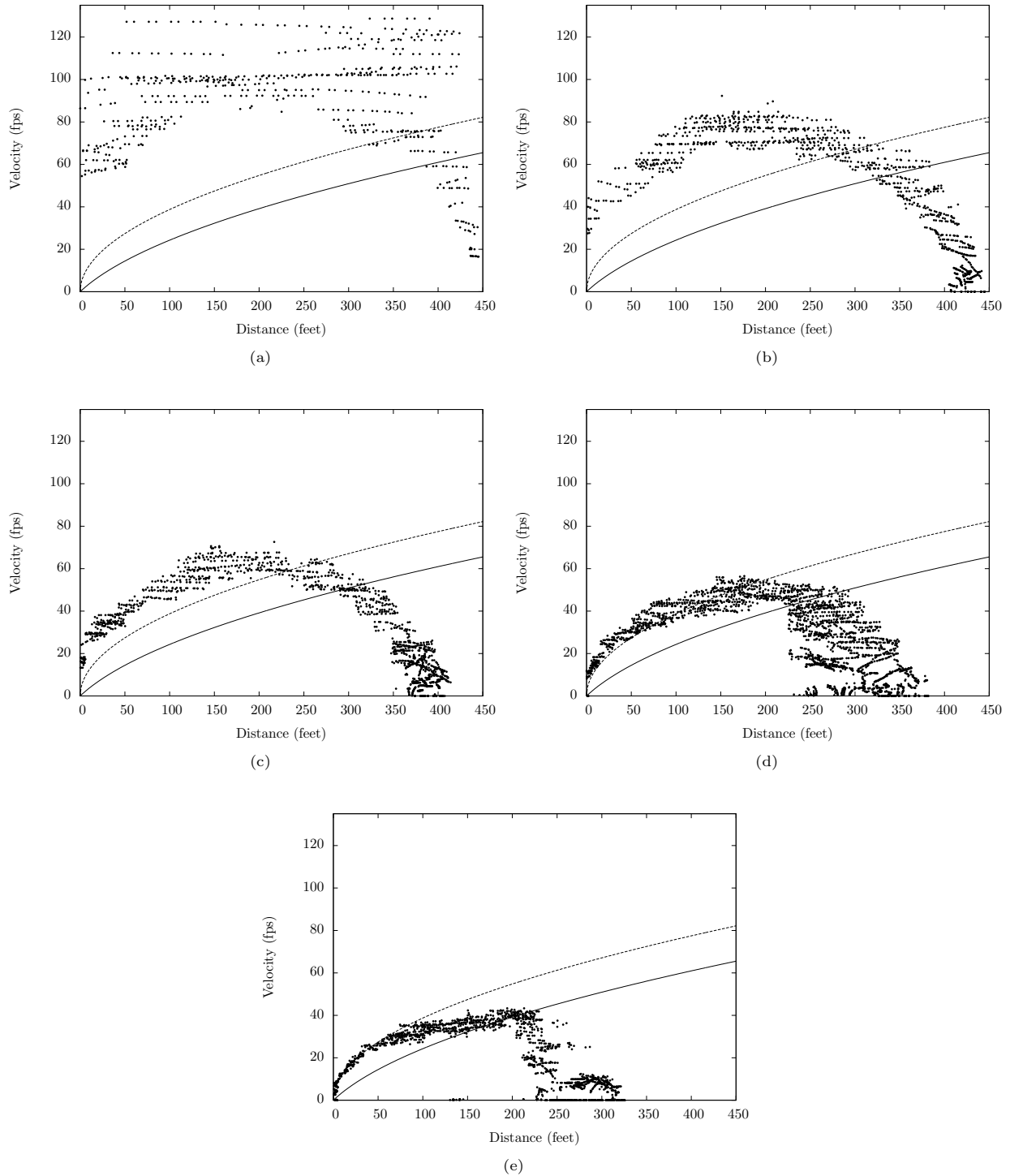


Figure 4: The ranking results of *i*Forest, trained using the crash non-time series data. Figure (a) shows the first 500 unsafe points plus the additional alert and safe points. Figure (b) through to figure (d) are the next subsequent sets of 500 unsafe points respectively. Figure (e) shows the last 85 unsafe points.

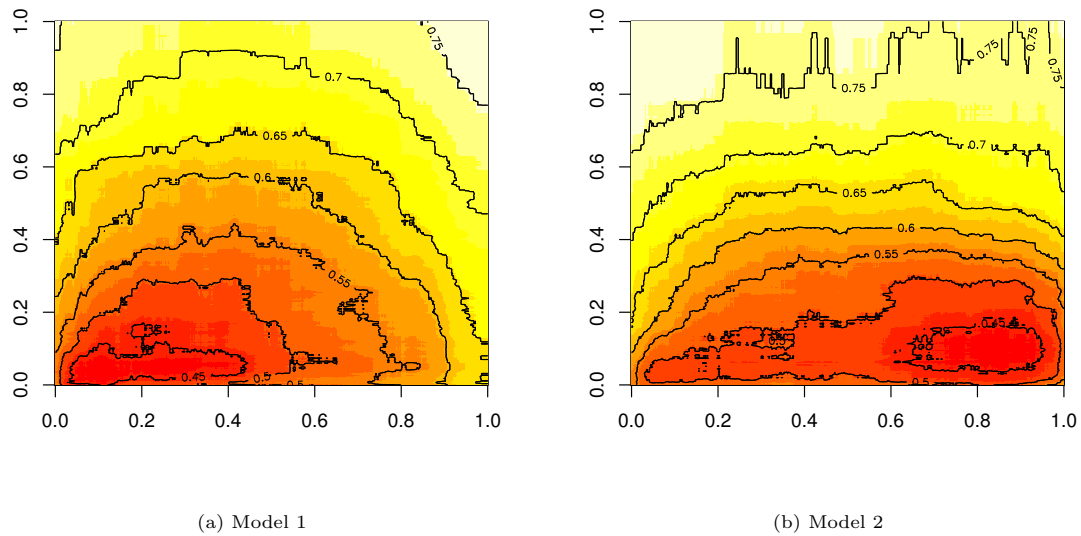


Figure 5: Contour maps of *iForest* for Models 1 and 2. They are produced using the anomaly scores output from *iForest*. The anomaly scores range from 0 to 1. The anomaly score function can be found in the Appendix.

Detector		Crash					Near Crash				
		Time Series			Non-Time Series		Time Series			Non-Time Series	
		Model A	Model B	Model C	Model 1	Model 2	Model A	Model B	Model C	Model 1	Model 2
AUC	<i>iForest</i>	0.7244	0.8997	0.6631	0.9531	0.9955	0.6873	0.8576	0.5259	0.9696	0.9963
	ORCA	0.7579	0.8883	0.6724	0.9389	0.9372	0.6339	0.8384	0.4980	0.9917	0.8359
Time	<i>iForest</i>	12	502	2	1	2	100	21	22	22	21
	ORCA	478	55,446	10,049	26	7720	6,314	9,135	222,416	606	164,991

Table 6: The AUC and timing results using two different anomaly detectors. For ORCA, each best k value is searched from 5, 10, 20, 40, 60, 80, 150, 250, 300, 500, 1000, 2000, 3000, and 4000 while using the crash data. The best k values are 10, 10, 4000, 80, 4000 for Model A, Model B, Model C, Model 1 and Model 2, respectively. The same corresponding k values, for each of the models, are used for the near crash data. For *iForest*, there are no parameter search. The timing results are given in seconds.

Iter #	<i>iForest</i>					ORCA				
	Attribute #	Attribute Name	Stats	AUC		Attribute #	Attribute Name	Stats	AUC	
1	109	min dist front	min Δf^2	0.8724		106	min dist front	mean Δf	0.8043	
2	109,99	min dist front	std dev f	0.8878		106,109	min dist front	min Δf^2	0.8705	
3	109,99,95	Decel	std dev f^2	0.8943		106,109,93	Decel	min Δf^2	0.8859	
4	109,99,95,32	vehicle velocity	max f	0.8928		106,109,93,125	min dist rear	min Δf^2	0.8870	
5	109,99,95,32,58	yaw	mean Δf	0.8950	106,109,93,125,61	yaw	min Δf^2	0.8883		
6	109,99,...,58,106	min dist front	mean Δf	0.8958		106,109,...,61,45	vehicle velocity	min Δf^2	0.8883	
7	109,99,...,106,126	min dist rear	mean Δf^2	0.8970		106,109,...,45,29	right lane marker	min Δf^2	0.8880	
8	109,99,...,126,45	vehicle velocity	min Δf^2	0.8987		106,109,...,29,13	line lane marker	min Δf^2	0.8879	
9	109,99,...,45,104	min dist front	max Δf	0.8970		106,109,...,13,127	min dist rear	std dev Δf^2	0.8874	
10	109,99,...,104,71	acceleration	std dev f^2	0.8997		106,109,...,127,119	min dist rear	std dev f^2	0.8871	

Table 7: The AUC results of Model B for the Forward Greedy Search using *iForest* and ORCA.

The current data distribution has the highest concentration of points to the lower left corner, and the anomalies are around the perimeter to the top and right of the data cloud. Based on this data distribution, the results shown in Figure 4 demonstrated that *iForest* has correctly ranked all these data points—the most normal points are at the center of the data cloud which are ranked at the bottom of the list and the most outlying points are ranked at the top.

The contour maps of *iForest* for Models 1 and 2 are shown in Figure 5(a) and Figure 5(b), respectively. Note that the centre of the data cloud shifted from the bottom-left corner, in Figure 5(a), to the bottom-right corner, in Figure 5(b). This is a result of introducing a Gaussian distribution of synthetic points described in Section 6. Also note that Model 2, shown in Figure 5(b), can now better model the two boundaries shown in Figure 4.

6.2 AUC and runtime comparisons

We measure the detection performance of anomaly detectors in terms of area under ROC curve (AUC). In order to calculate AUC, the three labels need to be converted to a two-label problem. We are only interested in determining the anomalies — unsafe points. Alert is merged with safe to become the second label — safe. A perfect AUC score will have all of the unsafe points ranked at the top of the list.

Table 6 shows the AUC results for both *iForest* and ORCA. The time series results clearly show that Model B, using features selected from the Forward Greedy Search, outperformed Models A and C.

Table 7 shows the results for Model B using the first ten attributes selected by the Forward Greedy Search in the first ten iterations. *iForest* produces the best Model B result using 10 attributes and ORCA uses 5 attributes.

However, Model B in time series treatment still performs worse than Models 1 or 2 in non-time series treatment in terms of AUC, as shown in Table 6. This is the same for both crash and near crash data sets, regardless of *iForest* or ORCA is used.

The results also show that *iForest* runs significantly faster than ORCA, up to five orders of magnitude faster.

These results reveal that it is unnecessary to perform the tasks in time series which requires more features, additional computation and feature space transforming; whereas, treating each point independently as in a non-time series problem works well.

6.3 Summary

This paper investigates whether anomaly detectors can be used to alert a driver of an impending crash. We show that anomaly detectors can correctly rank outlying points of the given data distribution. Because the current data sets are collected solely from crash or near crash events only, the data distribution is bias and does not represent the true distribution. However, we have demonstrated that *iForest* can be easily trained with a different data distribution (ie. adding a Gaussian distribution to the existing data) to correct the data distribution bias and provide a better ranking result for this type of application. The current result also implies that an anomaly detector could potentially become the core of a vehicle warning system that has the flexibility to allow car manufacturers as well as car drivers to tailor the system to suit individual needs. This is because such an anomaly detector can be easily retrained to adapt to

new requirements of individual users. A vehicle warning system based on a rule-based approach (Knipling et al., 1993; Kiefer et al., 1999; Brunson et al., 2002) lacks this kind of flexibility; and the one based on a semi-supervised time series model is unnecessary complicated and may not work as well.

7 Conclusions

In this paper, we presented a study to examine whether a time series problem can be solved more effectively as a non-time series problem. In a vehicle related time series problem, we found that it can be solved as a non-time series problem with significantly improved AUC result compared to that achieved in time series. We also highlighted the disadvantages of existing methods: the rule-based approach is ‘hard-coded’, and the semi-supervised approach requires labels and significantly more features and a feature transformation, making the problem unnecessarily complex with no additional benefits.

We have demonstrated that anomaly detectors can be used in a way to create boundaries between safe and unsafe regions. These boundaries can be locally altered from time to time when required to adapt to the individual driver requirements which are hard or not possible to do with rule-based methods.

Although both *iForest* and ORCA show comparable AUC results, *iForest* is preferred because it runs significantly faster than ORCA, up to five orders of magnitude faster in our experiments.

Compare to the rule-based and semi-supervised approaches and ORCA, we conclude that *iForest* is the best algorithm to be incorporated into a vehicle warning system to provide alerts to drivers for any impending collisions because of its fast execution, simplicity and flexibility.

References

- A *Policy on Geometric Design of Highways and Streets* (2004). 5th. American Association of State Highway and Transportation Officials (AASHTO).
- Bay, Stephen D. and Mark Schwabacher (2003). Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In: *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining (KDD-03)*. ACM, pp. 29–38.
- Brunson, S., E. Kyle, N. Phamdo, and G. Preziotti (2002). *Alert algorithm development program — NHTSA rear-end collision alert algorithm - Final report*. Tech. rep. DOT-HS-809-526. Washington, DC: National Highway Traffic Safety Administration.
- Dingus, T. A. et al. (2006). *The 100 Car Naturalistic Driving Study, Phase II — Results of the 100 Car Field Experiment*. Tech. rep. DOT HS 810 593. Virginia Tech Transportation Institute.
- Kiefer, R.J., D. LeBlanc, M. Palmer, J. Salinger, R. Deering, and M. Shulman (1999). *Forward Collision Warning Systems: Development and Validation of Functional Definitions and Evaluation Procedures for Collision Warning/Avoidance Systems*. Tech. rep. DOT-HS-808-964. Washington, DC: National Highway Traffic Safety Administration.

- Knipling, R., M. Mironer, D. Hendricks, L. Tijerina, J. Everson, J. Allen, and C. Wilson (1993). *Assessment of IVHS countermeasures for collision avoidance: Rear-end crashes*. Tech. rep. DOT-HS-807-995. Washington, DC: National Highway Traffic Safety Administration.
- Knuth, D. E. (1998). *Art of Computer Programming, Volume 3: Sorting and Searching*. 2nd Ed. Addison-Wesley Professional.
- Liu, Fei Tony, Kai Ming Ting, and Zhi-Hua Zhou (2008). Isolation Forest. In: *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 08)*. IEEE Computer Society, pp. 413–422.
- McLaughlin, Shane B., Jonathan M. Hankey, Thomas A. Dingus, and Sheila G. Klauer (2009). *Development of an FCW Algorithm Evaluation Methodology With Evaluation of Three Alert Algorithms*. Tech. rep. DOT HS 811 145. Virginia Tech Transportation Institute.
- Neale, V.L., S.G. Klauer, R.R. Knipling, T.A. Dingus, G.T. Holbrook, and A. Petersen (2002). *The 100 Car Naturalistic Driving Study, Phase 1 — Experimental Design*. Tech. rep. DOT HS 809 536. Virginia Tech Transportation Institute.
- Ning, H., W. Xu, Y. Zhou, Y. Gong, and T. S. Huang (2010). A general framework to detect unsafe system states from multisensor data stream. In: *IEEE Transactions on Intelligent Transportation Systems* 11.3, pp. 4–15.
- Wang, Jinjun, Shenghuo Zhu, and Yihong Gong (2010). Driving Safety Monitoring Using Semisupervised Learning on Time Series Data. In: *IEEE Transactions on Intelligent Transportation Systems* 11.3, pp. 728–737.
- Zhou, Y., W. Xu, H. Ning, Y. Gong, and T. Huang (2007). Detecting unsafe driving patterns using discriminative learning. In: *Proceedings of 2007 IEEE International Conference Multimedia Expo (ICME 2007)*, pp. 1431–1434.

Appendix – iForest Algorithms

Algorithms 1 and 2 show the steps required to train an iForest. The tree height for each iTree is set automatically to $l = \text{ceiling}(\log_2 \psi)$ which is an approximation of the average tree height (Knuth, 1998), where ψ is the sub-sampling size. Each iTree is grown up to the average height because we are only interested in data points that have shorter-than-average path length.

Algorithm 1 : iForest(X, t, ψ)

Inputs: X - input data, t - number of trees, ψ - sub-sampling size

Output: a set of t iTrees

- 1: **Initialise Forest**
 - 2: set height limit $l = \text{ceiling}(\log_2 \psi)$
 - 3: **for** $i = 1$ to t **do**
 - 4: $X' \leftarrow \text{sample}(X, \psi)$
 - 5: $\text{Forest} \leftarrow \text{Forest} \cup \text{iTree}(X', 0, l)$
 - 6: **end for**
 - 7: **return Forest**
-

Algorithm 3 shows how to compute the path length for a given data point. $c(\psi)$ is defined as follows.

$$c(\psi) = 2H(\psi - 1) - \left(\frac{2(\psi - 1)}{\psi}\right). \quad (5)$$

where $H(i)$ is a harmonic number and it is estimated by $\ln(i) + 0.5772156649$ (Eulers constant).

Algorithm 2 : iTree(X, e, l)

Input: X - input data, e - current tree height, l - height limit

Output: an iTree

- 1: **if** $e \geq l$ or $|X| \leq 1$ **then**
 - 2: return $\text{exNode}\{\text{Size} \leftarrow |X|\}$
 - 3: **else**
 - 4: let Q be a list of attributes in X
 - 5: randomly select an attribute $q \in Q$
 - 6: randomly select a split point p from max and min values of attribute q in X
 - 7: $X_l \leftarrow \text{filter}(X, q < p)$
 - 8: $X_r \leftarrow \text{filter}(X, q \geq p)$
 - 9: return $\text{inNode}\{\text{Left} \leftarrow \text{iTree}(X_l, e + 1, l),$
 - 10: $\text{Right} \leftarrow \text{iTree}(X_r, e + 1, l),$
 - 11: $\text{SplitAtt} \leftarrow q,$
 - 12: $\text{SplitValue} \leftarrow p\}$
 - 13: **end if**
-

Algorithm 3 : PathLength(x, T, e)

Inputs : x - an instance, T - an iTree, e - current path length; to be initialized to zero when first called

Output: path length of x

- 1: **if** T is an external node **then**
 - 2: return $e + c(T.\text{size})$ $\{c(\cdot)$ is defined in Equation 5 $\}$
 - 3: **end if**
 - 4: $a \leftarrow T.\text{splitAtt}$
 - 5: **if** $x_a \geq T.\text{splitValue}$ **then**
 - 6: return $\text{PathLength}(x, T.\text{right}, e + 1)$
 - 7: **else** $\{x_a < T.\text{splitValue}\}$
 - 8: return $\text{PathLength}(x, T.\text{left}, e + 1)$
 - 9: **end if**
-

To find out the anomaly score for a data point, the path length is first obtained and then the anomaly score is computed by using equation 6.

$$s(x, \psi) = 2^{-\frac{E(h(x))}{c(\psi)}} \quad (6)$$

where $h(x)$ is the path length from a single iTree; $E(h(x))$ is the expected path length of iForest.

Variance-wise Segmentation for a Temporal-Adaptive SAX

Chao Sun¹

David Stirling¹

Christian Ritz¹

Claude Sammut²

¹ School of Electrical, Computer and Telecommunications Engineering
University of Wollongong
Wollongong NSW 2522, Australia,
Email: {chaos,david.stirling,critz}@uow.edu.au

² School of Computer Science and Engineering
The University of New South Wales
Sydney 2052, Australia
Email: claude@cse.unsw.edu.au

Abstract

The Symbolic Aggregate approXimation algorithm (SAX) is a very popular symbolic mapping technique for time series data, and it is widely employed in pattern identification, sequence classification, abnormality detection and other data mining research. Although SAX is a general approach which is adaptable to most data, it utilises a fixed-size sliding window in order to generate motifs (temporal shapes). When certain target phenomena (activities of interest) are manifested over differing time scales, SAX-motifs are unable to correctly account for all such targets. This paper proposes a new method named the variance-wise segmentation method which can adaptively change the size of the sliding window in a generalised SAX approach. By generating motifs with differing durations, patterns can be found for activities with similar shape but occurring over a significant altered time base. This method is tested on both artificially modified ECG data, as well as, variable tactile vibration data, with improved results compared to the original SAX formulation.

Keywords: Time Series Data, Adaptive Segmentation, SAX

1 Introduction

The time series is an important type of data due to its frequent appearance in various fields, such as finance, weather forecast, medicine and industry. Analysis and data mining on time series are distinct and often difficult compared to other common data types, mainly due to the natural temporal ordering characteristic.

The Symbolic Aggregate approXimation (SAX) developed by Lin et al. (Lin et al. 2003) is a technique designed for time series data mining, and it is widely employed for analysis on time series from many kinds of sources. SAX has been proven to be efficient and reliable for abnormality detection (Keogh & Lin 2005), sequence classification (Lin et al. 2003), similar patterns locating (Mueen & Keogh 2010) and many other tasks. However, the SAX method uses a fixed-size sliding window on the full range temporal data for the generation of symbolic motifs, which are

further used as the basic elements to decompose non-stationary time series, the same type of events with a temporal distortion may not be disassembled into sets containing similar motifs.

The assumption is that if a certain type of event can be recorded as a series of sequences, the shapes (temporal change patterns) of the recorded sequences will be similar to each other. However time distortions on records often happen in the real world when some activities are happening more quickly or slowly, and in this case, the shape patterns are stretched or compressed along the time axis. If the frequency changes happen as the time series data are being recorded, the traditional SAX technique with a fixed-size sliding window will have difficulty in correctly identifying the actual type of events.

This work proposes a dynamic segmentation method that is designed to improve the SAX technique. The new segmentation method is aimed at making the SAX transformation adaptive to time distortions in non-stationary time series by changing the size of sliding window dynamically. The remainder of this paper is structured as follows: Section 2 reviews related research. Section 3 describes the concept of variance-wise segmentation. Section 4 details experimental trials, and a comparison between the new and original SAX methods. Section 5 summarises the advantages and disadvantages of this method, and discusses possible improvements for the future.

2 Related work

A key problem in processing time series data efficiently and effectively is linked to its sequential representation. Transform or approximate representations of time series data can be categorised as either frequency-domain or time-domain (Keogh et al. 1993). Most transform methods, such as the Fourier and Wavelet Transform, belong to the frequency-domain, whilst in the time-domain, approximations such as, Symbolic Mappings and Piecewise Linear Representation (PLR) are widely used for time series representation.

2.1 Symbolic Representation – SAX

SAX is a typical symbolic mapping approximation for time series representation. Originally, SAX (Lin et al. 2003) used a sliding window to extract a section of sequential data, where the section is divided into several sub-sequences of equal length based on the specified SAX word length. Values in each subsequence are averaged and the whole section is converted into a

Piecewise Aggregate Approximation (PAA) representation (Chakrabarti et al. 2002). The PAA is then discretised into a list of symbols or values based on the quantising region that every PAA value lies within. The SAX mapping approach uses a Gaussian distribution to derive the regional breakpoints resulting in the generation of an equiprobable set of symbols, as illustrated in Figure 1, adapted from one of Keogh's paper (Keogh & Lin 2005) as an illustration.

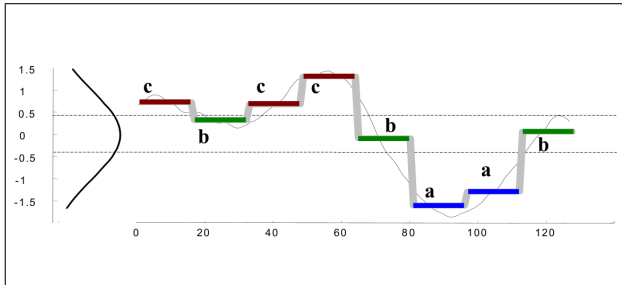


Figure 1: Concept of Original SAX

The SAX representation satisfies a Lower Bounding (LB) distance measurement property such that if two subsequences are similar, then the distance measurement (typically Euclidean) between their SAX motifs will be small, and generally less than the distance between the two original sequences. Lower Bounding is an important characteristic of SAX for pattern searching and indexing, and it promotes the SAX motifs to be used in conjunction with other data mining techniques.

The SAX motif represents the rough shape of the original time series data, which significantly reduces the amount of data to be processed. The technique is designed for quickly locating sections roughly matching a given shape pattern. However, with very long time series, the number of SAX motifs may also grow huge and makes the searching within the motif set difficult. In order to obtain improved index-ability for SAX representations, a multi-resolution extension of SAX, iSAX, was proposed by Shieh and Keogh (Shieh & Keogh 2008). The iSAX representation allows mixed cardinalities in the SAX words, and the symbols are represented as binary strings. This enables resultant SAX words to become hierarchically index-able, and this leads to fast approximate search with various resolutions. However, iSAX still relies on a Gaussian determination of breakpoints for the PAA discretisation, together with a fixed size sliding window as in the original SAX approach.

Because both SAX and iSAX presentation rely on a fixed Gaussian distribution threshold table, they are not adaptive to magnitude distribution changes within time series. Pham et al. (Pham et al. 2010) combined K-Means clustering algorithm to adaptively set the break points for each subsequence in the time series data. With Pham's adaptive SAX/iSAX method, the ranges for SAX symbols change accordingly to the sequential data and generates adaptive SAX representations which are claimed to outperform the original SAX and iSAX approaches. The adaptive (i)SAX can change the breakpoints based on the data in every sliding window, which solves the skew problem over different symbols when the values in datasets have a non-Gaussian distribution. However, for the patterns encased within varying temporal modes, adaptability does not exist for the temporal-modal, or frequency changes within the time series because of the fixed window size. If the SAX words are required to be adaptive in time-domain, a differ-

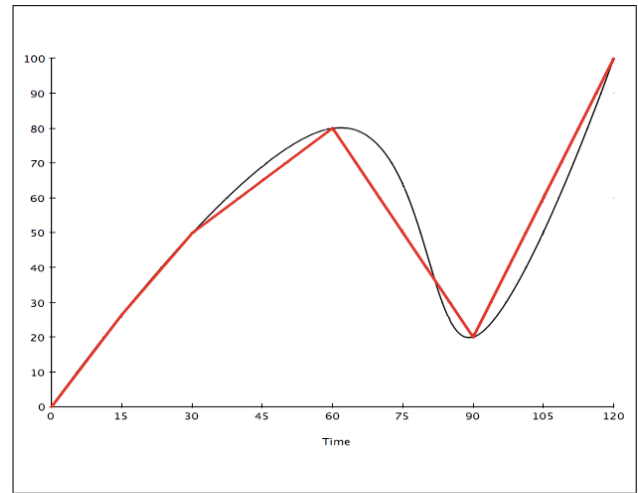


Figure 2: Example of Piecewise Linear Representation

ent segmenting approach has to be employed to adaptively change the size of PAA segments when the time series data alters frequently in nature.

2.2 Techniques for Dynamic Segmentation

Appropriate segmentation methods are always an important factor for time series analysis in the time-domain. Unlike the symbolic mapping approach of SAX, reviewed in the previous section, time series data are often compressed into short sequences which contain fewer samples but preserve the most important information of the original data. Segmentation is usually considered to be an optimisation problem which seeks the best fitting approximation with the lowest distance from the original series data.

The Piecewise Linear Representation (PLR) is the most frequently used representation for time series approximation. The PLR approximates a time series with a number of straight lines, and yields the best representation such that the maximum error for any segment is less than a certain threshold, or that the total error of all segments does not exceed the given threshold.

This error can be measured by the Euclidean distance, furthest points or other metrics. Figure 2 illustrates a simple PLR (red line) of the original data (black curve). Various algorithms can be employed to improve the outcome of PLR, such as Neural Networks (Chang et al. 2009), Hidden Markov Models (Ge & Smyth 2001), or Genetic Algorithms (Ghosh et al. 2011).

In financial time series, such as stock price or market index analysis, methods such as, Perceptually Important Points (PIP) are popular for sequence segmentation. The PIP approach was first proposed by Fu et al. (Fu et al. 2001) for financial time series representation. For each subsequence, the PIP segmentation approach initialises the first and last data points as PIPs, then the new PIP is selected by measuring the distance between the data points and the nearby PIPs. The data point with the maximum distance to the closest existing PIPs is subsequently selected as a new PIP, and these are further generated recursively until a stopping criteria is met.

The PIP is a typical Top-down segmentation approach which partitions the whole given sequence into smaller segments, and therefore is not suited for on-line segmentation of streaming data series. The PIP approach is designed to partition temporal sequences

into meaningful patterns similarly to the qualitative perceptions that humans may form, where the fitting error in each segment is not overly restricted in any way. Since there are many empirical financial patterns summarised by technical stock market analysts, the patterns segmented with PIP can be effectively used to match with the known patterns (Chung et al. 2004, Yu et al. 2010).

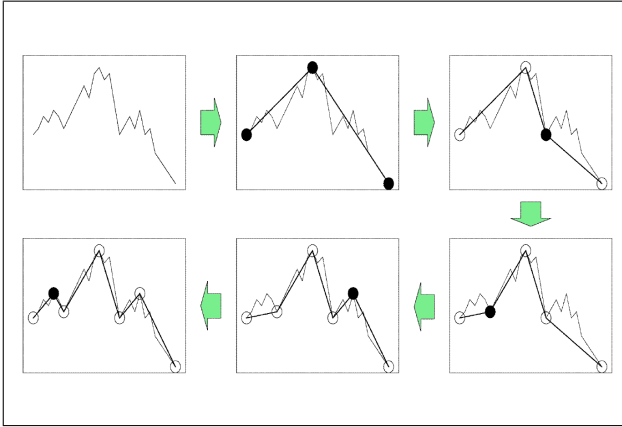


Figure 3: Example of PIP for discovering shoulder-head pattern in financial data

3 Temporal Adaptive SAX – using Variance-Wise Segmentation

One advantage of SAX motifs is that they can represent all possible patterns in a sequence as long as its resolution is high enough, and that these motifs can be further analysed by other data mining techniques. PLR produces adaptive segments and detects the change points, however the PLR segments are very likely to be monotonic because they are fitted with straight lines. Using the PLR segments for SAX motif generation is ineffective as a large number of interesting patterns would be ignored.

The other reviewed technique, the PIP, seeks to find important patterns that contain the most distinct points in a given period, however this top-down segmentation method needs the period and a set number of points to be identified prior to the actual segmentation process. Thus the PIP approach does not appear to be useful for dynamic and adaptive segmentation for an online procedure.

Although neither PLR nor PIP can be readily utilised for this purpose, a new segmentation approach has been inspired by both. The essential idea of this new adaptive segmentation approach is that every segment of data requires a sufficient amount of variance to ensure it partially covers some, or the whole, of an important pattern (similar to the concept of maximising distances in PIP), however the variance in any segment of data still needs to be limited to a reasonable extent to ensure motifs are comparable (similar to the maximum error in each segment in PLR).

The assumption here is that if a certain phenomena is recorded as a sequential pattern (shape) in time series data, then a related or similar natured phenomena generated in different time frame will be recorded as a related pattern with time distortion. Most real world temporal processes change in both time and amplitude, therefore distortions can be found on both dimensions. However, in this work we simplify the

Data: Sequence, Thresh, WordLen, AlphaSize

Result: SAX Motifs, Locations

[saxWords,locs]=vwSax(Seq,Thresh,wLen,alpha)

initialise segStart, segEnd, currSegVar

```

while segEnd ≤ length(Seq) do
    while currSegVar < Thresh do
        | segEnd += 1
    end
    while currSegVar Var ≥ Thresh do
        newSax = SAXFunc(segment, wLen,
            alpha)
        if newSax ≠ lastWord then
            | update(saxWords, locs, lastWord)
            | segStart += 1
        else
            | continue shrinking
        end
    end
end
return saxWords, locs
    
```

Algorithm 1: Variance-wise Segmenting SAX

problem and limit the distortion within the time domain only, and these situations do readily exist in the real world. For example, ECG data is a typical time series that records activities of the human heart, and the shape of each heartbeat cycle is regular for normal people. The frequency of ECG signals may vary significantly during exercise or rest, however the amplitudes of the ECG waves hardly change (Battler et al. 1979, Dori & Bitterman 2008).

Our method can be described as follows, function $F()$ is defined to calculate the total variance within a section of data. Threshold Th is given as the criterion to make segments. For time series data T , any segment $[T_i : T_j]$ must satisfy two conditions: $F(T[i : j]) \geq Th$ and $F(T[i : j - 1]) < Th$. (i and j are the start and end points of the section, and $T[i : j]$ are the sequential data for SAX motif).

The function $F()$ can be defined by different methods, such as the total absolute differences among points, the total square error from the average value or any other variance measurements with similar meaning. The threshold Th is a fixed value or a value adaptively derived from the data. Unlike the PIP method, this segmentation method does not expect that the extracted patterns are physically meaningful. As long as the SAX motifs are generated for further modelling, the meaning and importance of the motifs can be determined later by other data mining techniques.

The segmentation method uses a sliding window with variable size, therefore it suits online SAX motif generation. When the current variance of the segment is less than the threshold, the window is expanded by moving the end point forward. And when the current variance of the segment is greater than the threshold, the window is reduced by advancing the starting point. Similar to original SAX, every segment with enough variance will be converted to a SAX motif, and the new motif is compared with the previous motif. If the new motif differs from the previous one, both segments and the motif will be saved for future use. Figure 4 illustrates how this method works under different situations. Algorithm 1 is the pseudo code for the new segmentation approach (variance function is not included).

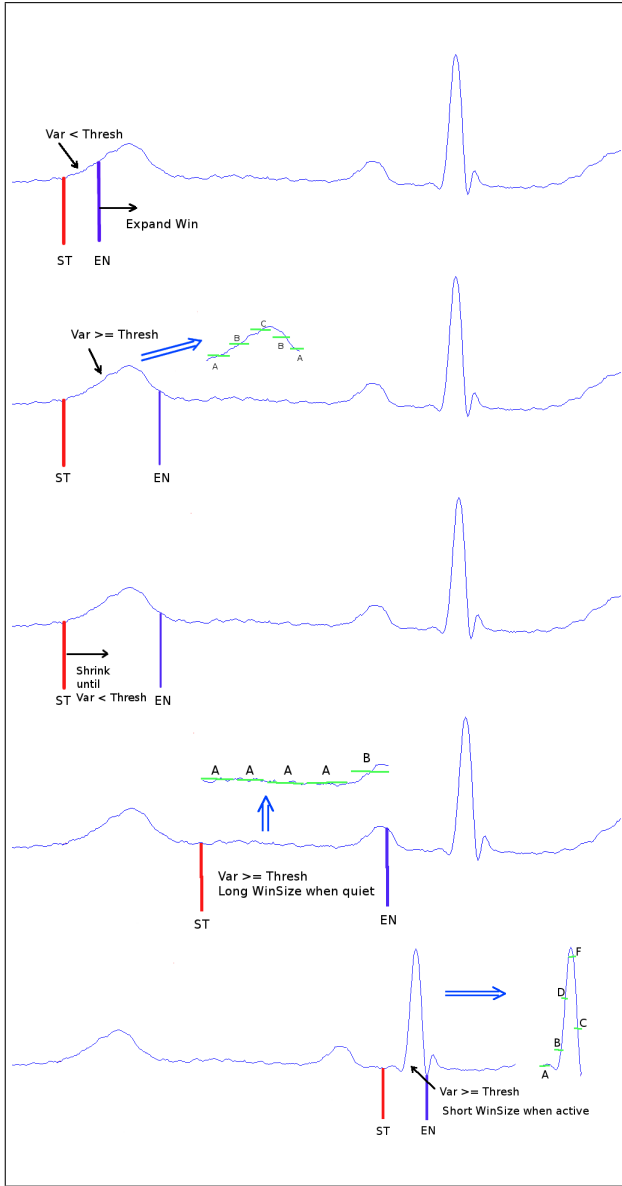


Figure 4: Temporal Adaptive SAX using Variance-wise Segmentation Method

4 Experiment

The results of preliminary experiments are presented in this section. Real ECG data obtained from the MIT-BIH database (Moody & Mark 2001) have been artificially modified for testing purposes. The variance function is set to accumulate the absolute differences of nearby records in a specific segment of data, and the threshold for variance is selected to be certain multiples of the standard deviation of the same data.

$$F(T[i:j]) = \Sigma(|diff(T[i:j])|)$$

$$Thresh = N \times std(T)$$

Besides the ECG data, a set of real data from a tactile sensor used by Jamali and Sammut (Jamali & Sammut 2012) are also tested using the same code.

4.1 MIT-BIH ECG Data and Setups

Original SAX is used in the first experiment, and the word length and alphabet size are both set to 5. Figure 5 is one of the artificially modified ECG sequences

Table 1: Var-wise SAX Event Stats (ECG)

Evt	SampNo	MotNo	AvgWin	MinWin	MaxWin
N1	327	29	67	22	173
N1*	932	39	200	60	440
N2	309	30	62	20	173
N2*	878	35	170	50	399
PVC	380	18	107	40	304
PVC*	1080	23	283	108	799

tested, captured from the No. 119 ECG record in the MIT-BIH database. The whole sequence in Figure 5 contains 7700 samples, where the first 2000 samples are from the original ECG signal (stage 1) including four normal and a premature ventricular contraction (PVC) heartbeats sampled at 360 Hz. The rest of the data (stage 2) are interpolated from the stage 1 data using the “resample” function in the Matlab time-series toolbox (Matlab 2010b). The signal in stage 2 is increased to 5700 samples and accordingly its sampling rate is 1026 Hz.¹

In Figure 5, the normal and PVC heartbeats in stage 1 have distinct temporal shape patterns, and these shape patterns are well maintained in stage 2 after the interpolation. The second and third normal heartbeats and the PVC in stage 1 are highlighted and labelled as N1, N2 and PVC, and accordingly their up-sampled versions are N1*, N2* and PVC*. Using the original SAX approach, a sliding window covering the whole ECG event in stage 1 can only cover less than half the event in stage 2, thus obviously the motif set representing N1 will be different from the motif set representing N1*. In both SAX approaches, a window (with either fixed or varying size) is used to slide through the whole sequence and generates SAX words continuously. Therefore, for a pre-defined event, some SAX motifs are fully included and some others are overlapped with the event. The motif set for an event is constructed with all fully included SAX motifs within the period of such event. Table 1 lists the sample number, number of motifs, average window size and min/max window size in the motifs sets for all the 6 events highlighted in Figure 5.

A few facts can be observed from the statistics summary in Table 1:

1. For the variance-wise SAX, the sliding window does adaptively increase its size during stage 2, and the change ratio is close to the actual up-sample ratio. (2.6–2.9:1 on the average window sizes against the real up-sample ratio at 2.85:1).
2. The sliding window may vary more than double the size according to the activity on time series data. Increasing the threshold increases the possible coverage of each segment, and once the segment is able to cover a whole normal event, the window size will vary less during the normal periods. Some later experiments show that if the threshold is increased to 15 times the standard deviation, the window size for N1 event vary less (185–217 samples). However, for irregular events with significant changes, such as PVC and PVC*, the window size still varies significantly (100–306 samples) during the abnormal period.

¹Please note that the frequency difference on the ECG signal is artificially manipulated to an extent that will never happen in real. It is common to see the heartbeat rate on the same subject to increase from 60 bpm to 100 bpm, however for subjects in the MIT-BIH database (mostly patients), there is no record containing such rate variance. Therefore the data used in this work is generated for validation purposes only, and the frequency change of the data can also be variable.

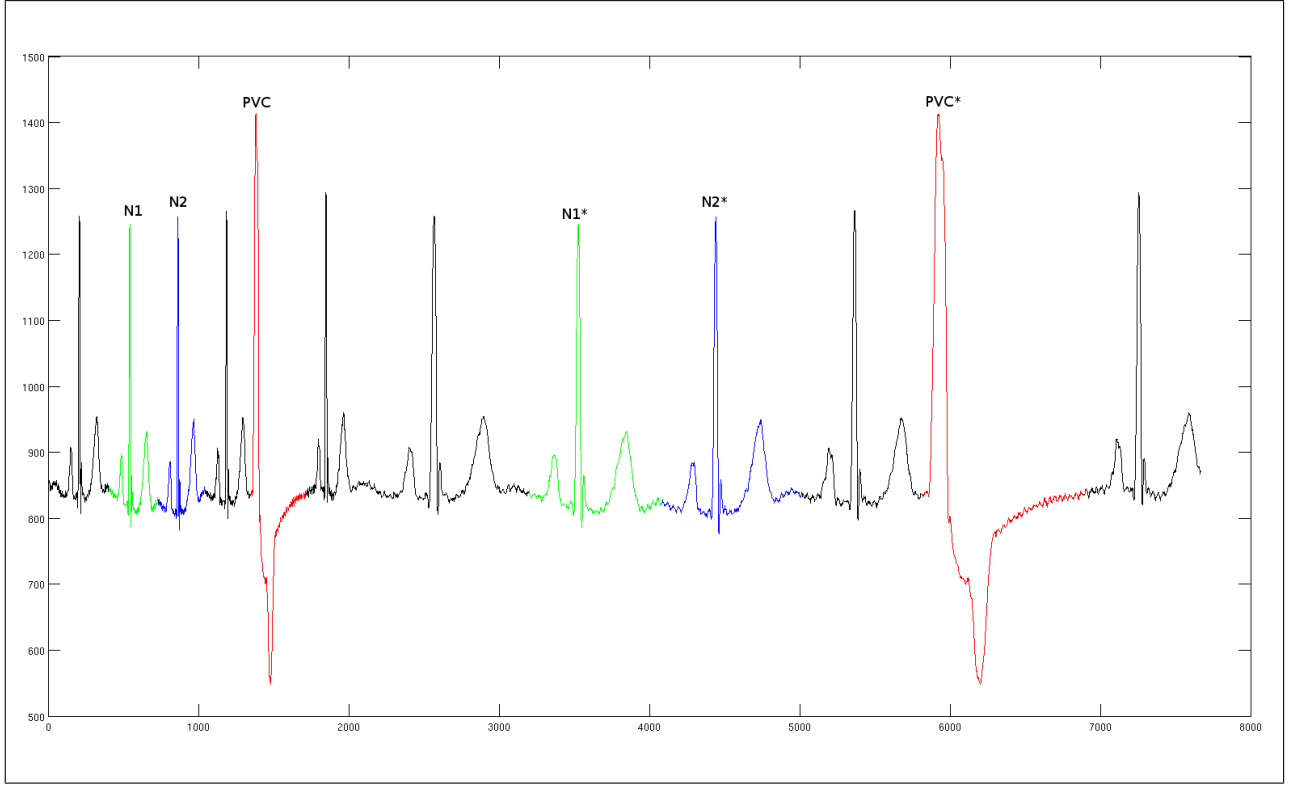


Figure 5: Artificially Changed ECG Signal

3. With the adaptive sliding window, the sizes of the motif set for the same event in both stage 1 and 2 are close, and similar motif patterns can be expected from both stages regardless the frequency difference.

4.2 ECG outcomes

As the number of motifs for each event varies from 18 to 39, it is not necessary to list all of the actual SAX motifs. In order to compare the motif sets efficiently, we design a table to show the similarity correlations between different motif sets. Every row of the table represents an event, and the columns indicate the event to be compared. Every element is constructed by two values (C/D), Common(C) is the number of motifs which find a match in the comparison motif set, and Different(D) is the number of motifs that miss a match in the comparison set. When the total absolute difference of two motifs is less or equal to 1, we define it as a similar match. For example, the total difference between [2,2,3,4,5] and [2,2,2,4,5] is 1, so they are considered a match. However the differences from [2,2,3,4,5] to [2,3,4,4,5] and [4,2,3,4,5] are both 2, so they are not similar matches. A high (C/D) ratio means that the SAX representation set for the row event is similar to the motif set or part of the set as comparison.

The threshold for the variance-wise segmentation method is set to 10 times the standard deviation, and for the normal SAX method, a sliding window size of 160 is used, which is close to half the period of a normal heartbeat event in stage 1. Results for both variance-wise SAX and normal SAX are listed in Table 2 and 3 respectively.

In both Table 2 and 3, the comparisons between a normal ECG event and the same event in artificially stretched ECG signal are highlighted with bold fonts. For example, the first row in both tables compares

Table 2: Motif Set Similarity – Var-Wise SAX (ECG)

C/D	N1	N2	PVC	N1*	N2*	PVC*
N1	29/0	28/1	0/29	29/0	25/4	0/29
N2	28/2	30/0	0/30	29/1	27/3	0/30
PVC	0/18	0/18	18/0	1/17	2/16	18/0
N1*	38/1	36/3	1/38	39/0	36/3	1/38
N2*	26/9	29/6	3/32	32/3	35/0	3/32
PVC*	0/23	0/23	23/0	1/22	2/21	23/0

Table 3: Motif Set Similarity – Normal SAX (ECG)

C/D	N1	N2	PVC	N1*	N2*	PVC*
N1	31/0	27/4	0/31	16/15	13/18	2/29
N2	27/4	31/0	1/30	14/17	15/16	7/24
PVC	0/18	1/17	18/0	4/14	4/14	17/1
N1*	15/41	15/41	5/51	56/0	52/4	8/48
N2*	17/56	17/56	6/67	65/8	73/0	10/63
PVC*	1/32	4/29	18/15	6/27	6/27	33/0

the motif set from event N1 with motif sets from all other events. According to Table 2, the variance-wise SAX generates 29 unique motifs during the period of N1, and in which 28 motifs found similar matches in the motif set for event N2, therefore N1 is very similar to N2 based on their SAX motif components. N1 is dissimilar with PVC as no motif finds a match in the PVC's motif set. Comparison between N1 and the stretched events N1* and N2* also show good similarities using the variance-wise SAX. All motifs from N1 can find matches in the motif set for N1*, and 25 out of 29 motifs find matches in the motif set for N2*. This means the stretched normal heartbeats (N1*, N2*) produce motif sets similar to the motif set of original heartbeat (N1). A comparison of the motifs between N1 and PVC* shows they are very dissimilar as no motif can be matched between these two. Comparisons among all other events are similarly listed in

Table 4: Motif Set Similarity – Tactile Data

	VW SAX			Orig SAX		
C/D	Ev1	Ev2	Ev3	Ev1	Ev2	Ev3
Ev1	24/0	19/5	16/8	41/0	26/15	7/34
Ev2	22/4	26/0	22/4	32/10	42/0	23/19
Ev3	14/13	23/4	27/0	11/17	22/6	28/0

the rest part of Table 2.

In Table 3, 31 motifs are generated for N1 with the normal SAX approach, and the comparison results are close to the variance-wise SAX when they are done within the same frequency range. However, when N1 is compared with stretched normal heartbeats N1* and N2*, the number of similar matches drop to about half of the total motif number (16 with N1* and 13 with N2*). Comparison between N1 and PVC* shows that 29 out of 31 motifs are dissimilar, leaving 2 motifs with matches in the PVC* motif set.

Analysis of the tables above shows the advantage of variance-wise SAX as its C/D ratios are not affected significantly by changes in frequency. The distinctions of motif sets mostly correlate with the difference of event types. However when using the traditional SAX approach for the same task, the C/D ratio between motif sets of an event and its stretched version drops significantly from 100:0 to approximately 50:50.

The work conducted on the artificial ECG data illustrates how the variance-wise segmentation method helps on generating temporal-adaptive SAX motifs. In the next section some data captured from a real application will be used for further evaluation.

4.3 Tactile Data

These sequential data were captured during texture recognition experiments with an artificial finger. Vibrations are detected by several polyvinylidene fluoride (PVDF) sensors embedded in the finger (Jamali et al. 2009).

The density of a moving set of mechanical ridges that are drawn across the surface of the artificial finger changes from 10 units to 30 units and then 50 units, these are referred to as three different stages, S1, S2 and S3. We randomly select data from each stage as follows: 1450 samples in S1, including 4 ridges; 1000 samples in S2, including 10 ridges; 1000 samples in S3, including 17 ridges. The resultant data are seen in Figure 6, and one example event is randomly selected from each stage as highlighted with differing colours, similar to previous work with ECG. The sampling rate for PVDF signal is 500 Hz, thus Figure 6 represents about 7 seconds worth of data.

Unlike the artificially stretched ECG data used in previous section, the tactile data contain a significant amount of noise, therefore the performance can be expected to be lower. Both the variance-wise SAX and normal SAX are tested to evaluate the performance with frequency-varying data.

Because the noise and temporal distortion are significant in the tactile data, we allow higher tolerance when comparing motifs. In this case, if the total difference between two motifs is less than or equal to 2, they are considered to match. The threshold for tactile data is set to 7 times the standard deviation. For the normal SAX, the size of sliding window is 30 (Approximately half period of the stage 3 data). The motif set similarity matrices for both methods are listed in Table 4.

Table 5: Match Rate among All Tactile Events

	VW SAX (avg(std)%)		
	Stage1	Stage2	Stage3
Stage1	96.7(6.4)	74.4(16.3)	46.8(17.1)
Stage2	78.2(13.7)	91.2(11.8)	64.6(19.3)
Stage3	40.7(13.1)	59.1(17.2)	66.2(20.9)
	Orig SAX (avg(std)%)		
Stage1	99.5(1.2)	65.2(17.6)	23.9(12.3)
Stage2	70.6(16.8)	89.9(10.3)	43.3(14.4)
Stage3	42.7(16.9)	62.2(17.1)	81.9(16.8)

The temporal adaptive SAX with the variance-wise segmentation method generates similar motif sets across all three stages with different frequency patterns, and the C/D ratios are generally higher than results with traditional SAX approach in Table 4. Table 5 presents the comparison results among all events from Stage 1, 2 and 3. Because the sizes of motif sets are not the same, in this table we use the average percentage of all matching rates instead of the number of matches. The standard deviation of the matching rate is also listed in the brackets.

The high noise level, especially in Stage 3, affects the performance of variance-wise SAX. The noise not only affects the overall shape similarity between stages, but also introduces a lot of variances that affect segmentation and give rise to new shape patterns that do not exist under normal condition. It is believed that improved techniques for calculating the variance function and adding a noise filter to remove random events will provide improved performance for the tactile data.

5 Conclusion and Future Work

The paper proposes a new method to dynamically change the size of the sliding window in the SAX algorithm. To generate SAX motifs which are adaptable to time distortions, the total variance within a period is used as the main criterion for segmentation. The method is tested on artificially modified ECG signal and vibration data from a recent real application. The results show the time-adaptive SAX outperforms traditional SAX on sequences with different temporal-modal patterns.

The variance-wise segmenting method is designed to improve the SAX technique, and it can be employed by both the SAX and iSAX. Because the key idea of SAX is not changed, the new time-adaptive SAX motifs are still guaranteed to be lower bounding, and can be utilised for online processing. Further analysis is needed for studying how the new segmenting method changes the lower bound tightness of the SAX motifs.

This work concentrates on frequency varying time series data only, whilst in practice distortions commonly happen on both the time and magnitude axes. Future research will focus on the follows:

1. Appropriate variance function for the type of time series.
2. Evaluation of the method on more real data, such as the tactile signal.
3. Combination of this method and magnitude adaptive (i)SAX for a fully adaptive SAX technique.

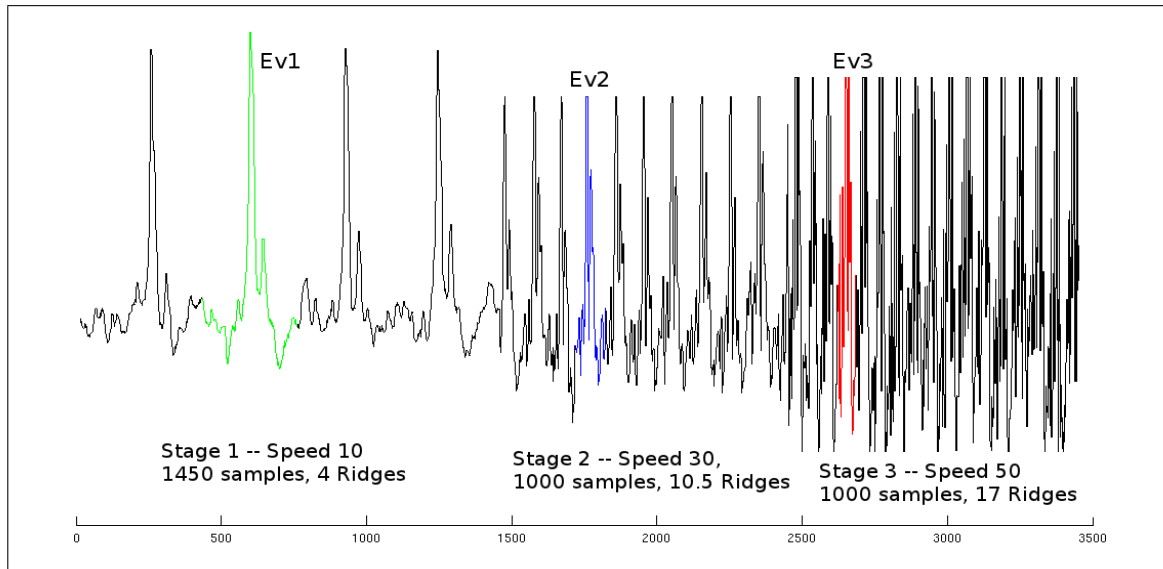


Figure 6: Tactile Data

References

- Battler, A., Froelicher, V., Slutsky, R. & Ashburn, W. (1979), 'Relationship of qrs amplitude changes during exercise to left ventricular function and volumes and the diagnosis of coronary artery disease.', *Circulation* **60**(5), 1004–13.
- Chakrabarti, K., Keogh, E., Mehrotra, S. & Pazzani, M. (2002), 'Locally adaptive dimensionality reduction for indexing large time series databases', *ACM Trans. Database Syst.* **27**(2), 188–228.
- Chang, P.-C., Fan, C.-Y. & Liu, C.-H. (2009), 'Integrating a piecewise linear representation method and a neural network model for stock trading points prediction', *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* **39**(1), 80–92.
- Chung, F.-L., Fu, T.-C., Ng, V. & Luk, R. (2004), 'An evolutionary approach to pattern-based time series segmentation', *Evolutionary Computation, IEEE Transactions on* **8**(5), 471–489.
- Dori, G. & Bitterman, H. (2008), 'Ecg variability contour-a reference for evaluating the significance of amplitude ecg changes in two states.', *Physiol Meas* **29**(8), 989–997.
- Fu, T.-C., lai Chung, F., Ng, V. & Luk, R. (2001), 'Evolutionary segmentation of financial time series into subsequences', in 'Evolutionary Computation, 2001. Proceedings of the 2001 Congress on', Vol. 1, pp. 426–430 vol. 1.
- Ge, X. & Smyth, P. (2001), 'Segmental semi-markov models for endpoint detection in plasma etching', *IEEE Transactions on Semiconductor Engineering*.
- Ghosh, S., Ray, A., Yadav, D. & Karan, B. (2011), 'A genetic algorithm based clustering approach for piecewise linearization of nonlinear functions', in 'Devices and Communications (ICDeCom), 2011 International Conference on', pp. 1–4.
- Jamali, N. & Sammut, C. (2012), 'Slip prediction using hidden markov models: Multidimensional sensor data to symbolic temporal pattern learning', in 'Robotics and Automation (ICRA), 2012 IEEE International Conference on', pp. 215–222.
- Jamali, N., Byrnes-Preston, P., Salleh, R. & Sammut, C. (2009), 'Texture recognition by tactile sensing', in 'Australasian Conference on Robotics and Automation (ACRA), December 2-4, 2009, Sydney, Australia'.
- Keogh, E. & Lin, J. (2005), 'Hot sax: Efficiently finding the most unusual time series subsequence', pp. 226–233.
- Keogh, E., Chu, S., Hart, D. & Pazzani, M. (1993), 'Segmenting time series: A survey and novel approach', in 'In an Edited Volume, Data mining in Time Series Databases. Published by World Scientific', Publishing Company, pp. 1–22.
- Lin, J., Keogh, E. J., Lonardi, S. & chi Chiu, B. Y. (2003), 'A symbolic representation of time series, with implications for streaming algorithms', in 'DMKD', pp. 2–11.
- Moody, G. & Mark, R. (2001), 'The impact of the mit-bih arrhythmia database', *Engineering in Medicine and Biology Magazine, IEEE* **20**(3), 45–50.
- Mueen, A. & Keogh, E. (2010), 'Online discovery and maintenance of time series motifs', in 'Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining', KDD '10, ACM, New York, NY, USA, pp. 1089–1098.
- Pham, N. D., Le, Q. L. & Dang, T. K. (2010), 'Two novel adaptive symbolic representations for similarity search in time series databases', in 'Proceedings of the 2010 12th International Asia-Pacific Web Conference', APWEB '10, IEEE Computer Society, Washington, DC, USA, pp. 181–187.
- Shieh, J. & Keogh, E. J. (2008), 'isax: indexing and mining terabyte sized time series', in 'KDD', pp. 623–631.
- Yu, H.-H., Tseng, V., Chen, C.-H. & Hong, T.-P. (2010), 'A pip-based evolutionary approach for time series segmentation and pattern discovery', in 'Computer Symposium (ICS), 2010 International', pp. 705–710.

Coding of Non-Stationary Sources as a Foundation for Detecting Change Points and Outliers in Binary Time-Series

Peter Sunehag Wen Shao Marcus Hutter

Research School of Computer Science
Australian National University
ACT 0200 Australia

Email: {peter.sunehag, wen.shao, marcus.hutter}@anu.edu.au

Abstract

An interesting scheme for estimating and adapting distributions in real-time for non-stationary data has recently been the focus of study for several different tasks relating to time series and data mining, namely change point detection, outlier detection and online compression/ sequence prediction. An appealing feature is that unlike more sophisticated procedures, it is as fast as the related stationary procedures which are simply modified through discounting or windowing. The discount scheme makes older observations lose their influence on new predictions. The authors of this article recently used a discount scheme for introducing an adaptive version of the Context Tree Weighting compression algorithm. The mentioned change point and outlier detection methods rely on the changing compression ratio of an online compression algorithm. Here we are beginning to provide theoretical foundations for the use of these adaptive estimation procedures that have already shown practical promise.

Keywords: Non-stationary sources, time-series, compression, detection, change point, outlier

1 Introduction

Data mining in time series data is an active and vast area of research with many applications Fu (2011) relating to various tasks like change detection Guralnik and Srivastava (1999), Kawahara and Sugiyama (2012) and outlier detection Fawcett and Provost (1999), Zhang et al. (2009). A unifying framework for these two tasks were developed in Yamanishi and Takeuchi (2002), Takeuchi and Yamanishi (2006) based on online learning in non-stationary environments using probabilistic modeling which discounts experiences over time so as to focus on recent observations. Recently in Kawahara and Sugiyama (2012), this was further developed into a real-time change detection method based on sequential discounting normalized maximum likelihood coding that was applied

to security applications, in particular malware detection. In the framework of Yamanishi and Takeuchi (2002), Takeuchi and Yamanishi (2006), Kawahara and Sugiyama (2012), a scoring function based on log loss, or in other words on arithmetic code length, was used to decide if recent observations were anomalous. If the average score over a number of consecutive time steps is sufficiently much higher than before, then a change has been detected. In compression terminology, the compressed size of those observations is higher than those before. This basic idea is also underlying the classical works E.S. (1955), Lorden (1971) on detecting change in a distribution.

Encoding a data source into a more compact representation is a long standing problem. In this paper, we are only concerned with the task of lossless data compression, which requires reproducing the exact original data from the compressed encoding. A number of different techniques for lossless data compression have been developed, for example Ziv and Lempel (1977, 1978), Cleary and Witten (1984), Cormack and Horspool (1987), Burrows and Wheeler (1994) to name a few. Many data compressors make use of a concept called arithmetic coding Rissanen (1976), Rissanen and Langdon (1979), which when provided with a probability distribution for the next symbol can be used for lossless compression of the data. In general, however, the true distribution for the next symbol is unknown and must be estimated. For stationary distributions, this estimation task is in many situations a solved problem and arithmetic coding based on the estimated distribution is optimal. For non-stationary distributions, estimating the true distribution is a much harder task. The Bayesian approaches Zacks (1983), Barry and Hartigan (1993) are attractive in that they are principled and automatically optimal but they are usually much more computationally expensive in their full form and, therefore, require approximation, in particular if they are going to run online R.P and D.J. (2007), Turner et al. (2009). If one is only interested in sequence prediction in the presence of change points and not in the change points themselves, the Bayesian approach Willems (1996) offers the possibility of using a mixture over all possible segmentations into piece-wise stationary intervals. Instead of using segmentation for sequence prediction, the framework by Yamanishi and Takeuchi (2002), Takeuchi and Yamanishi (2006), Kawahara and Sugiyama (2012) uses sequence prediction for segmentation.

Our interest here lies in methods that are as fast as their counterpart for the stationary case. Kawahara and Sugiyama (2012) achieves this using a simple discounting scheme and a similar technique is used by the authors of this article in O'Neill et al. (2012) to create a sequence prediction and compression algo-

This work was supported by ARC grant DP120100950.

Copyright ©2012, Australian Computer Society, Inc. This paper appeared at the 10th Australasian Data Mining Conference (AusDM 2012), Sydney, Australia, December 2012. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 134, Yanchang Zhao, Jiuyong Li, Paul Kennedy, and Peter Christen, Ed. Reproduction for academic, not-for-profit purposes permitted provided this text is included.

rithm for non-stationary environments based on the Context Tree Weighting (CTW) algorithm Willems et al. (1995), which relies upon the Krichevsky-Trofimov (KT) estimator. In O'Neill et al. (2012), we introduce an adaptive version of the KT estimator and use this to define the adaptive CTW algorithm. In the case of non-stationary binary sequences, the algorithm of Kawahara and Sugiyama (2012) would also naturally be based on this estimator. By proving redundancy bounds for the adaptive KT estimator for interesting classes of environments, we automatically get a bound for adaptive CTW as well as a theoretical foundation for the empirically successful change detection algorithm from Kawahara and Sugiyama (2012). An alternative approach to discounting for dealing with non-stationarity is to use a moving window. The discounting version can be viewed as an approximation of this approach. The windowed KT and the resulting windowed CTW was studied in Kawabata and Rashid (2003) and redundancy bounds was proved for stationary (d :th order) Markov sources. We are instead first going to consider a source whose Bernoulli parameter moves within an interval that is small in the Kullback-Leibler sense and then consider drifting sources as well as sources when the parameters (or interval of parameters) can jump significantly but rarely.

Related work. Stationary sources have been extensively studied, Krichevsky and Trofimov (1981) provides a good survey. For example, Krichevsky (1968) provides an asymptotic lower bound for redundancy of block to variable universal code for Bernoulli sources; Krichevsky (1970) provides a corresponding upper bound. Trofimov (1974) provides a finite bound for stationary d :th order Markov sources which is also studied by Kawabata and Rashid (2003). Krichevsky (1998) looks at asymptotic one step redundancy bound for stationary Bernoulli sources.

2 Windowed Krichevsky-Trofimov Estimation for Non-Stationary Sources

The KT estimator Krichevsky and Trofimov (1981), in this article often referred to as the regular KT estimator, is obtained using a Bayesian approach by assuming a $(\frac{1}{2}, \frac{1}{2})$ -Beta prior on the parameter of a Bernoulli distribution. Let $y_{1:t}$ be a binary string containing a zeros and b ones. We write $P_{kt}(a, b)$ to denote $P_{kt}(y_{1:t})$. The KT estimator can be incrementally calculated by: $P_{kt}(a+1, b) = \frac{a+1/2}{a+b+1} P_{kt}(a, b)$ and $P_{kt}(a, b+1) = \frac{b+1/2}{a+b+1} P_{kt}(a, b)$ with $P_{kt}(0, 0) = 1$.

Allowing changes in the underlying sources suggests that 'outdated' histories do not necessarily provide useful and accurate information for predicting the next bit as it does in the stationary case. The regular KT estimator is very slow to update once many samples have been collected, so it cannot quickly adapt to a change in the source. Therefore, we will in this section look at a scheme where we estimate the probability of the next bit using the KT estimator, however, as opposed to counting the number of zeros and ones in the entire history, we only take the latest n bits into account. We call this moving window KT or windowed KT.

Redundancy bounds for windowed KT. We are interested in one-step prediction. Assuming a stationary Bernoulli source θ , an estimation for the probability of the next bit x when given the latest n bits, as a string w , yields a code length $-\ln \hat{p}(x|w)$. We then

take an expectation over all possible x and history w to define the (expected) redundancy by

$$R_\theta(n) = \sum_{|w|=n} p_\theta(w) \sum_{x \in \mathcal{B}} p_\theta(x) (-\ln \hat{p}(x|w)) - H(\theta)$$

where $H(\theta)$ is the entropy of source θ . $p_\theta(w)$ and $p_\theta(x)$ are the probabilities of observing string w and x under θ respectively. $\hat{p}(x|w)$ is given by the KT-estimator

$$\hat{p}(x|w) = \frac{r_x(w) + 1/2}{n+1} \quad (1)$$

where $r_x(w)$ is the number of x that appears in w . For a non-stationary Bernoulli source, the one step redundancy is defined accordingly. Suppose $x_{1:m}$ is generated by a non-stationary Bernoulli process, with x_i being sampled according to θ_i , the one step redundancy $R_m(n)$ at step m given a window size n is

$$\sum_{|w|=n} p_{\theta_{m-n+1:m}}(w) \sum_{x \in \mathcal{B}} p_{\theta_{m+1}}(x) (-\ln \hat{p}(x|w)) - H(\theta_{m+1}).$$

Theorem 1. Suppose that a binary sequence is generated by a non-stationary Bernoulli process, with parameters θ_i where $\theta_i = \theta^1$ when $i \leq n$ and $\theta_i = \theta^2$ when $i > n$. We estimate the probability of the $(n+1)$:th letter by the KT-estimator. If $\theta^1, \theta^2 \in (0, 1)$, $\theta^1 \leq \theta^2$, then

$$R(n) \leq KL(\theta^2 || \theta^1) + \frac{1+o(1)}{n} + \frac{\theta^2(3-4\theta^1)}{2n\theta^1} + \frac{(1-\theta^2)(4\theta^1-1)}{2n(1-\theta^1)}$$

The proof technique used is largely borrowed from Krichevsky (1998) where the following Lemma was proven.

Lemma 2. Krichevsky (1998). Let

$$b(n, k, \theta) = \binom{n}{k} \theta^k (1-\theta)^{n-k}.$$

There is a constant C such that the inequality

$$\sum_{k=0}^{\lambda-\delta} b(n, k, \theta) < C \lambda e^{-(\delta^2/\lambda) + ((\delta+1)^2/2)(\lambda-\delta)}$$

holds for $n > 2$, $0 < \theta < 1$, $\lambda = np$, $1 < \delta < \lambda$.

Proof for Theorem 1. The one step redundancy we want to bound is $R(n) =$

$$\sum_{|w|=n} p_{\theta_{m-n+1:m}}(w) \sum_{x \in \mathcal{B}} p_{\theta_{m+1}}(x) (-\ln \hat{p}(x|w)) - H(\theta_{m+1}) \quad (2)$$

where $\hat{p}(x|w)$ is given by the classic KT-estimator

$$\hat{p}(x|w) = \frac{r_x(w) + 1/2}{n+1} \quad (3)$$

with $r_x(w)$ being the number of x in string w . More specifically, the redundancy for the special case of this

Theorem, $R_{\theta^1, \theta^2}(n)$, can be rewritten as

$$\sum_{|w|=n} p_{\theta^1}(w) \sum_{x \in \mathcal{B}} p_{\theta^2}(x) (-\ln \hat{p}(x|w)) - H(\theta^2) \quad (4)$$

We can rewrite $H(\theta^2)$ as

$$H(\theta_R) = \ln n - \frac{1}{n} (\lambda_{R,1} \ln \lambda_{R,1} + \lambda_{R,0} \ln \lambda_{R,0}) \quad (5)$$

where $\lambda_{R,x}$ is the number of expected x that appear in n , i.e. $\lambda_{R,x} = n\theta_R^x(1-\theta_R)^{(1-x)}$. Noticing that $-\ln \hat{p}(x|w)$ in equation (2) contains $\ln(n+1)$ while $H(\theta_R)$ contains an $\ln n$ term, we Taylor expand the function $\ln(n+1)$ at the origin and get that

$$\ln(n+1) < \ln n + \frac{1}{n} - \frac{1}{2n^2} \quad (6)$$

Plugging equation (3,5,6) into (4) yields

$$\begin{aligned} nR_{\theta^1, \theta^2}(n) &\leq 1 + \frac{1}{n} - \frac{1}{2n^2} \\ &+ \lambda_{R,1} \ln \lambda_{R,1} - \lambda_{R,1} \sum_{k=0}^n b(n,k,\theta^1) \ln(k + \frac{1}{2}) \\ &+ \lambda_{R,0} (\ln \lambda_{R,0} - \sum_{k=0}^n b(n,k,1-\theta^1) \ln(k + \frac{1}{2})) \end{aligned} \quad (7)$$

where $b(n,k,\theta) = \binom{n}{k} \theta^k (1-\theta)^{n-k}$. Letting

$$\begin{aligned} F(n,\theta,\theta') &= \\ \frac{1}{2} + \lambda \ln \lambda - \lambda \sum_{k=0}^n b(n,k,\theta') \ln(k + \frac{1}{2}) \end{aligned} \quad (8)$$

where $\lambda = n\theta$, we can rewrite equation (7) as

$$\begin{aligned} nR_{\theta^1, \theta^2}(n) &\leq \frac{1}{n} - \frac{1}{2n^2} + \\ F(n,\theta^2,\theta^1) + F(n,1-\theta^2,1-\theta^1) \end{aligned} \quad (9)$$

and to bound this we are going to show that

$$F(n,\theta,\theta') \leq \frac{1}{2} + n\theta \ln \frac{\theta}{\theta'} + C''n$$

for some constant C'' . Next we Taylor expand $\ln(k + \frac{1}{2})$ at $\lambda' = n\theta'$

$$\ln(k + \frac{1}{2}) = \ln \lambda' + \frac{k + \frac{1}{2} - \lambda'}{\lambda'} + \mathcal{R}(k) \quad (10)$$

The remainder $\mathcal{R}(k)$ is

$$\mathcal{R}(k) = -\frac{(k + \frac{1}{2} - \lambda')^2}{2\xi(k)^2} \quad (11)$$

where $\xi(k)$ lies between λ' and $k + \frac{1}{2}$. By plugging

equation (10) into (8), we get

$$F(n,\theta,\theta') = \quad (12)$$

$$\frac{1}{2} + n\theta \ln \frac{\theta}{\theta'} - \frac{\theta}{2\theta'} - \lambda \sum_{k=0}^n b(n,k,\theta') \mathcal{R}(k) \quad (13)$$

Take $n > \frac{1}{\theta'} + 1$ and choose a natural number δ with $1 < \delta < \lambda'$. We split the summation in the last term into two parts: $0 \leq k \leq \lambda - \delta$ and $\lambda - \delta < k \leq n$. To bound the first part, we use that $\xi(k) > \frac{1}{2}$. Putting $\delta = \lambda^{3/4}$ and using the previous lemma it follows that $\mathcal{R}(k) \geq -2(k - \lambda + \frac{1}{2})^2$ and therefore

$$\begin{aligned} -\lambda \sum_{k=0}^{\lambda-\delta} b(n,k,\theta') \mathcal{R}(k) &\leq \\ 2\lambda(\lambda' + \frac{1}{2})^2 \sum_{k=0}^{\lambda-\delta} b(n,k,\theta') &< C' \lambda(\lambda' + \frac{1}{2})^2 e^{-\sqrt{\lambda'}} \end{aligned}$$

for some constant C' . To deal with the second part, we choose n large enough such that $\xi(k) > \frac{\lambda'}{2}$ and then we have

$$\mathcal{R}(k) \geq -\frac{2(k - \lambda' + \frac{1}{2})^2}{\lambda'^2}$$

Therefore,

$$-\lambda \sum_{k=\lambda-\delta}^n b(n,k,\theta') \mathcal{R}(k) \leq$$

$$\frac{2\lambda}{\lambda'^2} \sum_{k=0}^n b(n,k,\theta') (k - \lambda' + \frac{1}{2})^2$$

Using the second central moment of binomial distribution $m_2 = \lambda'(1-\theta')$ together with the first central moments, we have

$$-\lambda \sum_{k=\lambda-\delta+1}^n b(n,k,\theta') \mathcal{R}(k) \leq \frac{2\theta(1-\theta')}{\theta'} + \frac{\theta}{n\theta'^2}$$

Thus, we conclude that for large enough n

$$F(n,\theta,\theta') \leq \frac{1}{2} + n\theta \ln \frac{\theta}{\theta'} \quad (14)$$

$$+ \frac{\theta(3-4\theta')}{2\theta'} + C' \lambda(\lambda' + \frac{1}{2})^2 e^{-\sqrt{\lambda'}} \quad (15)$$

The last term decrease exponentially as $n \rightarrow \infty$ and can be replaced by $o(1)$, and write

$$F(n,\theta,\theta') < \frac{1}{2} + n\theta \ln \frac{\theta}{\theta'} + \frac{\theta(3-4\theta')}{2\theta'} + o(1)$$

Therefore, through Equation 9 we have

$$\begin{aligned} R_{\theta^1, \theta^2}(n) &\leq KL(\theta^2 || \theta^1) + \frac{1+o(1)}{n} \\ &+ \frac{\theta^2(3-4\theta^1)}{2n\theta^1} + \frac{(1-\theta^2)(4\theta^1-1)}{2n(1-\theta^1)} \end{aligned}$$

□

If we allow the parameters to move within an in-

terval $[\theta_L, \theta_R]$, then Theorem 1 above deals with the worst case situation, namely when θ_i is at one end point for m steps and then jumps to the other.

Corollary 3. Suppose $x_{1:m}$ is generated by a non-stationary Bernoulli process, with x_i being sampled according to θ_i . We estimate the probability of the i :th letter by the KT-estimator with a moving window of size $n < m$. If i is such that $m-n+1 \leq i \leq m+1$, $\theta_i \in [\theta_L, \theta_R]$, $\theta_L, \theta_R \in (0,1)$ and $\theta_L \leq \theta_R$, then the redundancy for this prediction is bounded by

$$R(n) \leq \frac{1+o(1)}{n} + \max\{KL(\theta_L||\theta_R), KL(\theta_R||\theta_L)\} + \frac{1}{n} \max\left\{\frac{\theta_R(3-4\theta_L)}{2\theta_L} + \frac{(1-\theta_R)(4\theta_L-1)}{2(1-\theta_L)}, \frac{\theta_L(3-4\theta_R)}{2\theta_R} + \frac{(1-\theta_L)(4\theta_R-1)}{2(1-\theta_R)}\right\}$$

Example 4. In the above bounds we notice that the constant factor in the $O(1/n)$ term grows unboundedly when the parameters tend to 0 or 1. This is not just a problem with the bounds but a genuine phenomenon. Suppose that $\theta_i=1$ for n time steps and then switch to $\theta < 1$. The redundancy for the next time step is then $O(\log(1+n))$. We conclude that if we want a uniform constant for the $O(1/n)$ term we need to assume that we are a minimum distance away from the end points.

Corollary 5. Suppose $x_{1:m}$ is generated by a non-stationary Bernoulli process, with x_i being sampled according to $\theta_i \in [L, R]$ where $0 < L \leq R < 1$. We estimate the probability of the i :th letter by the KT-estimator with a moving window of size $n < m$. If i is such that $m-n+1 \leq i \leq m+1$, $\theta_i \in [\theta_L, \theta_R]$, $\theta_L, \theta_R \in [L, R]$ and $\theta_L \leq \theta_R$, then, the redundancy for this prediction is bounded by

$$R(n) \leq \max\{KL(\theta_L||\theta_R), KL(\theta_R||\theta_L)\} + C/n$$

where C does depend on L and R but not on θ_L or θ_R .

Remark 6. For the case when $\theta_L = \theta_R$ we do not have a problem at the end points. Consider $\theta_i=1 \forall i$ which means that we will almost surely have a constant sequence. Then the redundancy is $-\log \frac{1/2+n}{n+1} = \log(1 + \frac{1}{2(n+1)}) \leq \frac{1}{2(n+1)}$. Corollary 5 holds for $L=0$ and $R=1$ as long as $\theta_R = \theta_L$.

Geometrically drifting sources. Suppose $x_{1:m}$ is generated by a non-stationary Bernoulli process, with x_i being sampled according to θ_i . If the source is such that for all i , $KL(\theta_{\max(i,n)}, \theta_{\min(i,n)}) \leq g(n)$, where

$$\theta_{\min(i,n)} = \min_{i \leq j \leq i+n} \{\theta_j\}$$

$$\theta_{\max(i,n)} = \max_{i \leq j \leq i+n} \{\theta_j\}$$

we can for any fixed i , apply Theorem 1. We next define a class of drifting sources for which there is a simple function g of this sort.

Definition 7 (Geometrically drifting source). Suppose a sequence $\{x_i\}_{i=1}^\infty$ is generated by a non-stationary Bernoulli process, identified by $\{\theta_i\}_{i=1}^\infty$ (with $\theta_1 \in (0,1)$) with each x_i sampled according to θ_i . We say that the source is geometrically drifting

if and only if $1 \leq \max\{\frac{\theta_i}{\theta_{i+1}}, \frac{1-\theta_i}{1-\theta_{i+1}}\} \leq c$ for all i and some constant $c \geq 1$.

The idea behind this definition is that the source can only drift, i.e. increase or decrease by a certain ratio c . This notion of drift allows us to bound the KL divergence of the maximum and minimum θ during n consecutive steps.

$$KL(\theta_i||\theta_{i+1}) = \theta_i \ln \frac{\theta_i}{\theta_{i+1}} + (1-\theta_i) \ln \frac{1-\theta_i}{1-\theta_{i+1}} < \ln c$$

for all i and it holds that

$$\begin{cases} 1 \leq \frac{\theta_{\max}}{\theta_{\min}} \leq c^n \\ 1 \leq \frac{1-\theta_{\min}}{1-\theta_{\max}} \leq c^n \end{cases}$$

which results in a bound for $KL(\theta_{\max}||\theta_{\min})$ (and the same for $KL(\theta_{\min}||\theta_{\max})$), namely

$$KL(\theta_{\max}||\theta_{\min}) = \theta_{\max} \ln \frac{\theta_{\max}}{\theta_{\min}} + (1-\theta_{\max}) \ln \frac{1-\theta_{\max}}{1-\theta_{\min}} \leq n \ln c$$

3 Discounted Estimation

When dealing with non-stationary sources, it is natural that one wants to weight recent history higher. We define an adaptive KT estimator, which we call discounting KT based on replacing a_n and b_n in the definition of the KT estimator with discounted counts. These counts are defined by applying the following discounting operation after adding a new zero ($a_n = a_n + 1$) or a new one ($b_n = b_n + 1$),

$$a_{n+1} := (1-\gamma) a_n \quad b_{n+1} := (1-\gamma) b_n$$

where $\gamma \in [0,1)$ denotes the discount rate. For discounting KT with $\gamma > 0$, we have an effective horizon of length $\frac{1}{1-\gamma}$. The windowed estimator from the previous section can be viewed as a hard version of this scheme.

Consider a situation where we have a stationary source ($\theta_i = \theta \forall i$) where we use a windowed KT with window length $n = \frac{1}{1-\gamma}$. Compare the distribution for the coefficient a (the number of zeroes in the window) with the distribution for the a coefficient defined from discounting from an infinite history. Both distributions are symmetric around the same mean but the one arising from the discounting has more mass close to the mean. Hence the discounting method will have a lower redundancy. This is not surprising in this situation because the discounting estimate gets to use an infinite history of observations and if we use the full history KT we have zero redundancy. This is, however, not the situation that we want to use discounting KT in. Discounting KT effectively only depends on a small number of observations. The reason we let it depend at all on things further back is for convenience, it yields a very simple update formula where nothing has to be stored. This is very convenient when, as in the CTW algorithm, a KT estimator is created for every node in a tree, which might be deep. The conclusion is that the upper bounds for the redundancy of windowed KT should at least approximately also hold for discounting KT. Furthermore, when the

source has been close to stationary for longer than the window length, one should expect marginally better from the discounting algorithm. Another case when one expect better from the discounting algorithm is for slowly drifting sources. We will below provide a class of drifting sources that is such that for any window, we are going to satisfy the assumption of Corollary 5 and one can conclude a redundancy bound for moving window which does tell us what we should at least expect from discounting KT.

4 Implications for Compression and Detection of Change Points and Outliers

We have showed that if the parameters stay within a small interval and we have a large enough (though not too large) window, the expected redundancy for windowed KT is small. This also applies if instead the parameter drift is small. We argue that this is not only true for windowed KT with a suitable window length (for the amount of drift) but for discounting KT with an appropriate discount factor. In this section we discuss the implications for the motivating applications.

Compression. Since expectation is a linear operation, the total redundancy is the sum of the per step redundancies. In the case of a stationary source, a source constrained to a small interval or a slowly drifting source within a larger interval our per bit redundancy bounds are simply multiplied by the file length to get the total redundancy. When we have a source with a small number of jumps and otherwise only slow (geometric) drift, Theorem 1 tells us that we have to add the sum of the KL-divergences times the window length for the jumps to the estimate. Note that with a window length n , the worst code length for a step is less than $\log(1+n)$. If we have $h(N)$ jumps during the first N time steps, the total accumulated redundancy is less than $h(N)\log(n+1)+O(N/n)$. Hence, if jumps are rare, they will not affect the total compressed length significantly. Regular KT has unbounded one step redundancy and the regular KT estimator also continues to be affected by all old data as much as newer observations. Hence, if there is substantial change in the middle of the file, the first part will adversely affect the rest of the file. In O'Neill et al. (2012), an empirical advantage of adaptive CTW over regular CTW was demonstrated on files, which were created by concatenating two different shorter files.

Here we have so far only proved bounds for the adaptive KT estimators and not the CTW algorithm which is more relevant for practical compressions since it takes context into account. However, bounds for the adaptive KT estimator is all that is needed to prove bounds for adaptive CTW. This can be done easily because there are three parts contributing to the redundancy bounds for CTW Willems et al. (1995): (1) redundancy used to find the 'right' tree (2) parameter redundancy given the 'right' tree (3) arithmetic coding redundancy (always bounded by 2). The first term has nothing to do with the underlying estimator but with the code length of the 'right' tree. The problem is thus reduced to the second term (the main term), which is the redundancy of the underlying KT estimator. For regular KT in the stationary case, this is $O(1/m)$ for the m :th time step. This adds up to a logarithmic term for the first N time steps and this is the only non-constant term. The $O(1/n)$ terms is for windowed KT replaced by an $O(1/n)$ term (Theorem 1) in the stationary case (where n is the size of the window) which accumulates to $O(N/n)$ if N is the total length of the file. In the interval case the

term is replaced by $KL+O(1/n)$, again according to Theorem 1. In these cases the total redundancy of adaptive CTW is $O(N/n)$. If we have $h(N)$ jump point of KL-divergence at most D , the redundancy is $h(N)D+O(N/n)$.

Detecting change points and outliers. The algorithm for detecting change points and outliers in Yamanishi and Takeuchi (2002), Takeuchi and Yamanishi (2006), Kawahara and Sugiyama (2012) relies on $\log Pr(x_{t+1} | x_{1:t})$ as a score. The idea is that this score is large in expectation if the distributions have changed significantly and smaller if it has not. They average over a number of time steps to get a good estimate of this expectation. That it will be large if the distribution change by much is clear but for it to be a well founded method one also needs to be able to say that it will be small if the distribution has at most changed a little. This is what we provide a theory for in this article.

5 Conclusion

Some recent advances in real-time online change point detection, outlier detection and compression have relied on discounting when estimating parameters of a distribution that is changing over time. A closely related alternative for dealing with non-stationarity in a computationally efficient manner is to only use the last few observations for the estimation. In this article we have provided a theoretical analysis of how these estimators behave for important classes of non-stationary environments and outlined the implication of the results for the application of the mentioned algorithms.

References

- Barry, D. & Hartigan, J. A. (1993), 'A bayesian analysis for change point problems', *Journal of the American Statistical Association* **88**(421), 309–319.
- Burrows, M. & Wheeler, D. (1994), 'A block-sorting lossless data compression algorithm', *Digital SRC Research Report*.
- Cleary, J. G. & Witten, I. H. (1984), 'Data compression using adaptive coding and partial string matching', *IEEE Transactions on Communications* **32**, 396–402.
- Cormack, G. V. & Horspool, R. N. S. (1987), 'Data compression using dynamic Markov modelling', *The Computer Journal* **30**(6), 541–550.
- E.S., P. (1955), 'A test for change in a parameter occurring at an unknown point', *Biometrika* **42**.
- Fawcett, T. & Provost, F. (1999), Activity monitoring: noticing interesting changes in behavior, in 'Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining', KDD '99, ACM, New York, NY, USA, pp. 53–62.
- Fu, T. (2011), 'A review on time series data mining', *Eng. Appl. Artif. Intell.* **24**(1), 164–181.
- Guralnik, V. & Srivastava, J. (1999), Event detection from time series data, in 'KDD', pp. 33–42.
- Kawabata, T. & Rashid, M. M. (2003), 'A zero redundancy estimator for the context tree weighting method with a finite window', *IEEE International Symposium on Information Theory* p. 114.

- Kawahara, Y. & Sugiyama, M. (2012), 'Sequential change-point detection based on direct density-ratio estimation', *Stat. Anal. Data Min.* **5**(2), 114–127.
- Krichevsky, R. E. (1968), 'The connection between the redundancy and reliability of information about the sources', *Problems of Information Transmission* **4**(3), 48–57.
- Krichevsky, R. E. (1970), Lectures on information theory, Technical report, Novosibirsk State University.
- Krichevsky, R. E. (1998), 'Laplace's law of succession and universal encoding', *IEEE Transactions on Information Theory* **44**, 296–303.
- Krichevsky, R. E. & Trofimov, V. K. (1981), 'The performance of universal encoding', *IEEE Transactions on Information Theory* **27**(2), 199–207.
- Lorden, G. (1971), 'Procedures for reacting to a change in distribution', *Annals of Mathematical Statistics* **42**(6), 1897–1908.
- O'Neill, A., Hutter, M., Shao, W. & Sunehag, P. (2012), Adaptive context tree weighting, in '2012 Data Compression Conference, Snowbird, UT, USA, April 10-12, 2012', IEEE Computer Society, pp. 317–326.
- Rissanen, J. J. (1976), 'Generalized Kraft inequality and arithmetic coding', *IBM Journal of Research and Development* **20**(3), 198–203.
- Rissanen, J. J. & Langdon, G. G. (1979), 'Arithmetic coding', *IBM Journal of Research and Development* **23**, 149–162.
- R.P., A. & D.J., M. (2007), 'Bayesian online change-point detection'.
- Takeuchi, J. & Yamanishi, K. (2006), 'A unifying framework for detecting outliers and change points from time series', *IEEE Trans. Knowl. Data Eng.* **18**(4), 482–492.
- Trofimov, V. K. (1974), 'The redundancy of markov source encoding', *Problems of Information Transmission* **10**(4), 16–24.
- Turner, R., Saatci, Y. & Rasmussen, C. E. (2009), Adaptive sequential Bayesian change point detection, in 'Advances in Neural Information Processing Systems (NIPS): Temporal Segmentation Workshop'.
- Willems, F. M. J. (1996), 'Coding for a binary independent piecewise-identically-distributed source', *IEEE Transactions on Information Theory* **42**(6), 2210–2217.
- Willems, F. M. J., Shtarkov, Y. M. & Tjalkens, T. (1995), 'The context tree weighting method: Basic properties', *IEEE Transactions on Information Theory* **41**, 653–664.
- Yamanishi, K. & Takeuchi, J. (2002), A unifying framework for detecting outliers and change points from non-stationary time series data, in 'Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26, 2002, Edmonton, Alberta, Canada', ACM, pp. 676–681.
- Zacks, S. (1983), *Survey of classical and Bayesian approaches to the change-point problem: Fixed sample and sequential procedures in testing and estimation*, Academic Press.
- Zhang, K., Hutter, M. & Jin, W. (2009), A new local distance-based outlier detection approach for scattered real-world data, in 'Proc. 13th Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD'09)', Vol. 5467 of *LNAI*, Springer, Bangkok, pp. 813–822.
- Ziv, J. & Lempel, A. (1977), 'A universal algorithm for sequential data compression', *IEEE Transactions on Information Theory* **23**, 337–342.
- Ziv, J. & Lempel, A. (1978), 'Compression of individual sequences via variable-rate coding', *IEEE Transactions on Information Theory* **24**, 530–536.

ABC-SG: A New Artificial Bee Colony Algorithm-Based Distance of Sequential Data Using Sigma Grams

Muhammad Marwan Muhammad Fuad

Department of Electronics and Telecommunications
Norwegian University of Science and Technology (NTNU)
NO-7491 Trondheim, Norway

marwan.fuad@iet.ntnu.no

Abstract

The problem of similarity search is one of the main problems in computer science. This problem has many applications in text-retrieval, web search, computational biology, bioinformatics and others. Similarity between two data objects can be depicted using a similarity measure or a distance metric. There are numerous distance metrics in the literature, some are used for a particular data type, and others are more general. In this paper we present a new distance metric for sequential data which is based on the sum of n-grams. The novelty of our distance is that these n-grams are weighted using artificial bee colony; a recent optimization algorithm based on the collective intelligence of a swarm of bees on their search for nectar. This algorithm has been used in optimizing a large number of numerical problems. We validate the new distance experimentally.

Keywords: Artificial Bee Colony, Extended Edit Distance, Sequential Data, Distance Metric, n-grams.

1 Introduction

Similarity search is one of the fundamental problems in computer science. It has many applications in text, video and image retrieval, pattern recognition, bioinformatics, web search, fingerprint databases, and many others. In this problem a pattern is given and the algorithm searches the database, or the web, to return all or most, depending on whether the search is exact or approximate, of the data objects that are “close” to that pattern according to some semantics of closeness. This closeness between two data objects is depicted using a principal concept which is the similarity measure or its stronger form; the distance metric.

Of the different paradigms proposed to manage the similarity search problem, the metric model with its properties (reflexivity, non-negativity, symmetry, triangle

inequality) stands out as one that is applicable to different data types. The distance metric on which the metric model is based is a strong mathematical tool which helps the researchers build different data structures specific to metric spaces. Other techniques, such as the pivot technique, are based on the triangle inequality; one of the axioms of the metric model. All these advantages of this model make of it a rich field of research in information retrieval.

The main distance used to compare two strings is the *Edit Distance* (ED) presented by Wagner and Fischer (1974), it is also called the *Levenshtein distance*, and it is defined as the minimum number of delete, insert, and change operations needed to transform string S into string T . As mentioned above, this distance is the main distance used to compare two strings. However, this distance has its limitations because it considers local similarity only.

Muhammad Fuad and Marteau (2008a) (2008b) presented a new distance metric; *The Extended Edit Distance* (EED), which they applied to symbolically represented time series. Unlike ED, EED considers a global level of similarity in addition to the local one presented by ED. EED is based on the idea of computing the frequencies of common characters between two strings. Later, Muhammad Fuad and Marteau (2008c) presented another distance, MREED, which computes the frequencies of common bi-grams in addition to common characters. However, the parameters used in these two distances (one in EED and two in MREED) were defined using very basic heuristics which, on the one hand, substantially limited the search space (it was limited to 5 values only for each parameter), and on the other hand, using such basic heuristics makes it practically impossible to extend this distance beyond that of bi-grams because training time is very long even in the case of bi-grams where two parameters only are used.

In this paper we propose a new general distance metric that applies to strings. We call it the *Artificial Bee Colony-Sigma Gram Distance* (ABC-SG). This distance is based on computing the sigma grams. The particularity of this distance is that it uses the artificial bee colony algorithm to set its parameters.

The rest of this paper is organized as follows: Section 2 is a background section, In Section 3 we present the new distance and we validate its performance in Section 4, we conclude this paper in Section 5 with some perspectives.

2 Background

Muhammad Fuad and Marteau (2008a) (2008b) presented the *Extended Edit Distance* (EED) which is defined as follows:

This work was carried out during the tenure of an ERCIM “Alain Bensoussan” Fellowship Programme. This Programme is supported by the Marie-Curie Co-funding of Regional, National and International Programmes (COFUND) of the European Commission.

Copyright © 2012, Australian Computer Society, Inc. This paper appeared at the 10th Australasian Data Mining Conference (AusDM 2012), Sydney, Australia, December 2012. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 134, Yanchang Zhao, Jiuyong Li, Paul Kennedy, and Peter Christen, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

Let Σ be a finite alphabet, and let Σ^* be the set of strings on Σ . Let $f_a^{(S)}, f_a^{(T)}$ be the frequency of the character a in S and T , respectively. Where S, T are two strings in Σ^* . EED is defined as:

$$EED(S, T) = ED(S, T) + \lambda \left[|S| + |T| - 2 \sum_{a \in \Sigma} \min(f_a^{(S)}, f_a^{(T)}) \right] \quad (1)$$

Where $|S|, |T|$ are the lengths of the two strings S, T respectively, and where $\lambda \geq 0$ ($\lambda \in R$). λ is called the co-occurrence frequency factor.

EED is based on the intuition that the ED distance does not take into account whether the change operation used a character that is more “familiar” to the two strings or not, because ED considers a local level of similarity only, while EED adds to this local level of similarity a global one. This modification makes EED more intuitive as shown by Muhammad Fuad and Marteau (2008a) (2008b).

Muhammad Fuad and Marteau (2008c) also showed that EED is a distance metric (symmetry, identity, triangle inequality). Search in metric spaces has many advantages, the most famous of which is that a single indexing structure can be applied to several kinds of queries and data types that are so different in nature. This is mainly important in establishing unifying models for the search problem that are independent of the data type. This makes metric spaces a solid structure that is able to deal with several data types as mentioned by Zezula et al. (2005).

3 The Artificial Bee Colony Sigma Gram Distance (ABC-SG)

3.1 Definition-The Number of Distinct n -Grams (NDnG)

Given two strings S, T . The number of distinct n -grams (substrings of length n) that the two strings S and T contain is defined as:

$$ND_nG(S, T) = |\{n\text{-gram}(S)\} \cup \{n\text{-gram}(T)\}| \quad (2)$$

where $n\text{-gram}()$ is the set of n -grams that a string consists of.

Example :

Given the following strings: $R = \text{oxygen}$, $S = \text{exogen}$, $T = \text{emolen}$. The sets of n -grams for these strings are given by:

n	R	S	T
1	o, x, y, g, e, n	e, x, o, g, e, n	e, m, o, l, e, n
2	ox, xy, yg, ge, en	ex, xo, og, ge, en	em, mo, ol, le, en
3	oxy, xyg, yge, gen	exo, xog, oge, gen	emo, mol, ole, len
4	oxyg, xyge, ygen	exog, xoge, ogen	emol, mole, olen
5	oxyge, xygen	exoge, xogen	emole, molen
6	oxygen	exogen	emolen

Comparing $ND_nG(S, R)$, and $ND_nG(S, T)$ gives:

n	$ND_nG(S, R)$	$ND_nG(S, T)$
1	5	4
2	2	1
3	1	0
4	0	0
5	0	0
6	0	0
$\sum_{n=1}^6 ND_nG$	8	5

The above comparison shows a greater similarity between S and R than between S and T , which is intuitive. But if we compute the edit distance we get: $ED(S, R) = ED(S, T) = 2$.

Muhammad Fuad and Marteau (2008a) (2008b) showed how EED, which considers the frequencies of characters, can capture this intuitive similarity that ED can not capture.

Although EED has advantages over ED as shown by Muhammad Fuad and Marteau (2008a), the way parameter λ is defined remains problematic. On the one hand, the search space is very limited, on the other hand, generalizing EED to use higher order frequencies of common grams using the same basic heuristics to define the different parameters makes the parameter defining process, for the different grams, inefficient and yet limited to very small regions in the search space.

In the following we present a generalizing of EED which uses an artificial bee colony based approach to determine the different parameters. This makes the search process more efficient and effective.

3.2 ABC-SG

Let Σ be a finite alphabet, and let Σ^* be the set of strings on Σ . Given n , let $f_{a_n}^{(S)}$ be the frequency of the n -gram a_n in S , and $f_{a_n}^{(T)}$ be the frequency of the n -gram a_n in T , where S, T are two strings in Σ^* . Let N be the set of integers, and N^+ the set of positive integers.

For notation convenience, we define the function:

$$g : \mathbb{N}^+ \times \Sigma^* \rightarrow \mathbb{N}$$

$$g(n, S) = n \quad \text{if } 1 \leq n \leq |S|$$

$$g(n, S) = |S| + 1 \quad \text{if } |S| < n$$

The ABC-SG distance between S and T is thus defined as:

$$ABC-SG(S, T) = \sum_{n=1}^{\max(|S|, |T|)} \lambda_n \cdot \left[|S| + |T| - g(n, S) - g(n, T) + 2 - 2 \cdot \sum_{a_n \in A^n} \min(f_{a_n}^{(S)}, f_{a_n}^{(T)}) \right] \quad (3)$$

where $|S|, |T|$ are the lengths of the two strings S, T respectively, and where $\lambda_n \in \mathbb{R}^+ \cup \{0\}$.

ABC-SG is based on the same concept of familiarity on which EED is based, but this concept is extended to the familiarity of n-grams instead of that of single characters.

It is important to notice that ABC-SG is actually the generalization of both EED (Muhammad Fuad and Marteau 2008a), (Muhammad Fuad and Marteau 2008b) and MREED (Muhammad Fuad and Marteau 2008c), so it includes the same advantages that these two distances have.

ABC-SG is proved to be a distance metric. For space limitations, the proof is not presented here. However, the proof is an extension of the proof presented by Muhammad Fuad and Marteau (2008a) (2008b).

As indicated earlier, the parameters λ_n are determined using the artificial bee colony algorithm.

3.3 Artificial Bee Colony

Bee-inspired optimization is a family of optimization algorithms that emerged from a larger family which is *swarm intelligence*. Baykasoğlu, Özbakır and Tapkan (2007) classify the behavioral characteristics of bee-based algorithms into three categories: foraging behaviors, marriage behaviors, and queen bee concept. One of the foraging behavior-based algorithms is *Artificial Bee Colony* (ABC) which was introduced by Karaboga (2005). In ABC each food source represents a potential solution to the optimization problem at hand and the quality of the food represents the value of the objective function to be optimized. Artificial bees explore and exploit the search space. These bees communicate and share information about the location and quality of food sources. This exchange of information takes place in the dancing area in the hive by performing a *waggle dance*.

In ABC there are three kinds of bees:

Employed bees: These are the bees that search in the neighborhood of a food source. They perform a dance with a probability that is proportional to the quality of the food source.

Onlooker bees: These bees are found on the dance floor. They watch the dances of the employed bees and place themselves on the most profitable food source.

Scouts: These bees explore the search space randomly.

As mentioned by Parpinelli, Benitez, and Lopes (2010), the balance between exploration and exploitation is maintained in ABC algorithm by combining local search methods, carried out by the employed and the onlooker bees, with global search methods, carried out by the scouts.

There are several variations of the ABC algorithm. In the following we present the standard ABC introduced by Karaboga and Basturk (2007a) (2007b), and by Diwold Beekman, and Middendorf (2010). The first step of ABC is generating a randomly distributed population size (*pop_size*) of food sources which correspond to potential solutions. Each solution $\vec{x}_i, i \in \{1, \dots, pop_size\}$ is a vector whose dimension is (*nr_par*) which is equal to the number of parameters of the function f to be optimized. The population is subject to change for a number of cycles (*nr_cycles*). In each cycle every employed bee perturbs the current solution using a local search procedure. The perturbation produces a new solution:

$$\vec{x}_i^* = \vec{x}_i + rand(-1, 1)(\vec{x}_i - \vec{x}_k) \quad , i \neq k \quad (4)$$

The above relation is not applied to all parameters but only to a certain number of them. The parameters to be altered are chosen randomly. The algorithm uses a greedy selection to decide if the new solution should be kept or discarded, i.e. :

Algorithm1 Artificial Bee Colony (ABC)

Require *pop_size*, *nr_par*, *nr_cycles*,
max_nr.

```

1: Initialize  $\vec{x}_i$ 
2: for cycle=1 to nr_cycles do
3:   for all employed bees do
4:      $\vec{x}_i^* = \vec{x}_i + rand(-1, 1)(\vec{x}_i - \vec{x}_k)$  ,  $i \neq k$ 
5:     if  $f(\vec{x}_i^*) < f(\vec{x}_i)$  then  $\vec{x}_i \leftarrow \vec{x}_i^*$ 
6:   end for
7:   calculate  $p_i = \frac{f(\vec{x}_i)}{\sum_{k=1}^{pop\_size} f(\vec{x}_k)}$ 
8:   for all onlooker bees do
9:      $\vec{x}_i^* = \vec{x}_i + rand(-1, 1)(\vec{x}_i - \vec{x}_k)$  ,  $i \neq k$ 
10:    if  $f(\vec{x}_i^*) < f(\vec{x}_i)$  then  $\vec{x}_i \leftarrow \vec{x}_i^*$ 
11:  end for
12: if nr_of_trials == max_nr then
13:   abandon current solution
14: end for
```

Figure 1. The Artificial Bee Colony Algorithm

$$\bar{x}_i = \begin{cases} \bar{x}_i^* & \text{if } f(\bar{x}_i^*) < f(\bar{x}_i) \\ \bar{x}_i & \text{otherwise} \end{cases} \quad (5)$$

After all employed bees have modified their positions the onlooker bees choose one of the current solutions depending on a probability that corresponds to the fitness value of that solution according to the following rule:

$$p_i = \frac{f(\bar{x}_i)}{\sum_{k=1}^{pop_size} f(\bar{x}_k)} \quad (6)$$

After that the onlooker bees try to improve the solution using the same mechanism that was described in (4). The number of trials the algorithm attempts to improve the same solution is limited by a maximum number (*max_nr*) after which the solution is abandoned and the bees employed by that food source become scouts. The abandoned solution is replaced by a new solution found by the scouts. Figure 1 outlines the ABC algorithm.

4 Performance Evaluation

We tested the new distance ABC-SG on symbolically represented time series. However, we think that ABC-SG is more appropriate for other sequential data types such as those encountered in bioinformatics and text mining.

Time series data are normally numeric, but there are different methods to transform them to symbolic data. The most important symbolic representation method of time series is the *Symbolic Aggregate Approximation* (SAX) introduced by Lin, Keogh, Lonardi, and Chiu (2003). SAX is based on an assumption that normalized time series have Gaussian distribution, so by determining the breakpoints that correspond to a particular alphabet size, one can obtain equal-sized areas under the Gaussian curve. SAX is applied as follows:

- 1-The time series are normalized.
- 2-The dimensionality of the time series is reduced using PAA; a representation method presented independently by Keogh, Chakrabarti, Pazzani, and Mehrotra (2000) and by Yi and Faloutsos (2000).
- 3-The PAA representation of the time series is discretized by determining the number and locations of the breakpoints (The number of the breakpoints is chosen by the user). Their locations are determined, as mentioned above, using Gaussian lookup tables. The interval between two successive breakpoints is assigned to a symbol of the alphabet, and each segment of PAA that lies within that interval is discretized by that symbol.

The last step of SAX is using the following similarity measure:

$$MINDIST(\hat{S}, \hat{R}) = \sqrt{\frac{n}{N}} \sqrt{\sum_{i=1}^N (dist(\hat{s}_i, \hat{r}_i))^2} \quad (7)$$

Where n is the length of the original time series, N is the length of the strings (the number of the segments),

\hat{S} and \hat{R} are the symbolic representations of the two time series S and R , respectively, and where the function $dist()$ is implemented by using the appropriate lookup table.

We also need to mention that the similarity measure used in PAA is:

$$d(S, R) = \sqrt{\frac{n}{N}} \sqrt{\sum_{i=1}^N (\bar{s}_i - \bar{r}_i)^2} \quad (8)$$

It is important to mention that MINDIST is not a distance metric (because it violates the axioms of distance metric) but a similarity measure.

We tested our new distance ABC-SG on a time series classification task based on the first nearest-neighbor (1-NN) rule using leaving-one-out cross validation. This means that every time series is compared to the other time series in the dataset. If the 1-NN does not belong to the same class, the error counter is incremented by 1.

We conducted experiments using datasets of different sizes and dimensions available at UCR of Keogh, Zhu, Hu, Hao, Xi, Wei, and Ratanamahatana (2011). This archive makes up between 90% and 100% of all publicly available, labeled time series data sets in the world, as mentioned by Ding, Trajcevski, Scheuermann, Wang, and Keogh (2008).

As indicated earlier, we tested ABC-SG on symbolically represented time series. This means that the time series were transformed to symbolic sequences using the first three steps of SAX presented earlier in this section, but instead of using MINDIST given in relation (7), we use our distance ABC-SG. The parameters λ_n in the definition of ABC-SG (relation (3)) are defined using ABC. This means, for each value of the alphabet size we formulate an artificial bee colony optimization problem where the fitness function is the classification error and the parameters of the optimization problem are λ_n . Theoretically n can take any value that does not exceed that of the shortest string of the two strings S, T . However, in the experiments we conducted we tested the new distance for $n \in \{1, 2, 3\}$ because these are the values of interest for time series. Notice that ABC-SG can be applied to strings of different lengths, which is one of its advantages since most similarity measures in time series mining are applied only to time series of the same length.

Concerning the control parameters of the ABC we used, the population size (the number of food sources) *pop_size* was 20. The number of cycles *nr_cycles* was set to 20. The number of trials of a certain food source *max_nr* was set to 10. The number of parameters *nr_par*, as mentioned earlier, was tested for $n \in \{1, 2, 3\}$. As for λ_n , their values are in fact unconstrained, but for simplicity we optimized them in the interval $[0, 2]$. Table 1 summarizes the symbols used in the experiments together with their corresponding values.

<i>pop_size</i>	Population size	20
<i>nr_cycles</i>	Number of cycles	20
<i>max_nr</i>	Number of trials	10
<i>nr_par</i>	Number of parameters	{1,2,3}

Table 1. The symbol table of ABC together with the corresponding values used in the experiments

For each dataset we use ABC on the training datasets to get the vector λ_n that minimizes the classification error on this training dataset, and then we use these optimal values of λ_n on the corresponding testing dataset to get the final classification error for each dataset.

We compared ABC-SG with dynamic time warping (DTW). DTW is a similarity measure that has been developed by the speech recognition community and later was used by Berndt, and Clifford (1994) on time series. DTW is an algorithm to find the optimal path through a matrix of points representing possible time alignments between the signals. Guo and Siegelmann (2004) state that the optimal alignment can be efficiently calculated via dynamic programming.

The dynamic time warping between the two time series $S = \{s_1, s_2, \dots, s_n\}$, $R = \{r_1, r_2, \dots, r_m\}$ is defined as follows:

$$DTW(i, j) = d(i, j) + \min \begin{cases} DTW(i, j-1) \\ DTW(i-1, j) \\ DTW(i-1, j-1) \end{cases} \quad (9)$$

where $1 \leq i \leq n, 1 \leq j \leq m$.

We chose to compare ABC-SG with DTW because DTW is known to give very good results in several time series data mining tasks such as classification and clustering. Another reason for choosing DTW is because it is applicable to time series of different lengths, which is the case with ABC-SG. However, ABC-SG has a complexity of $O(n^2)$, while that of ABC-SG is $O(N^2)$ ($N = n/4$ for compression ratio 1:4; the compression ratio usually used with SAX). So as we can see, ABC-SG has a much lower complexity than that of DTW. Another advantage that ABC-SG has over DTW is that ABC-SG is a distance metric while DTW is a similarity measure because it violates the triangle inequality.

It is important to mention that DTW is applied to the original time series and not to their symbolic representation.

In Table 2 we present some of the results we obtained for alphabet size equal to 3, 10, and 20, respectively.

As we can see from the results, the classification errors of ABC-SG are quite comparable to those of DTW despite the difference in complexity. In fact, in the majority of cases, ABC-SG even outperformed DTW. The results of other datasets in the archive were similar.

Beef				
	ABC-SG			DTW
	n=1	n=2	n=3	
$\alpha = 3$	0.567	0.567	0.567	0.5
$\alpha = 10$	0.5	0.5	0.467	
$\alpha = 20$	0.333	0.367	0.367	

(*: α is the alphabet size)

ECG				
	ABC-SG			DTW
	n=1	n=2	n=3	
$\alpha = 3$	0.18	0.21	0.22	0.23
$\alpha = 10$	0.2	0.22	0.22	
$\alpha = 20$	0.23	0.22	0.25	

FaceFour				
	ABC-SG			DTW
	n=1	n=2	n=3	
$\alpha = 3$	0.057	0.057	0.057	0.170
$\alpha = 10$	0.045	0.057	0.068	
$\alpha = 20$	0.09	0.102	0.102	

OSULeaf				
	ABC-SG			DTW
	n=1	n=2	n=3	
$\alpha = 3$	0.351	0.343	0.331	0.409
$\alpha = 10$	0.298	0.306	0.298	
$\alpha = 20$	0.322	0.330	0.330	

Table 2. Comparison between the classification error of ABC-SG and DTW for different values of the alphabet size and for different n-grams.

Another interesting remark which makes this distance meaningful is that we did not witness any correlation between the number of grams used and the classification error, which makes sense since ABC-SG, as mentioned in Section 3, is based on the concept of familiarity of n-grams between the two strings, and this familiarity is not related to the length of the n-gram.

Finally, in Table 3 we present, for reproducibility purposes, the values of λ_n obtained on the training datasets. As indicated earlier, when applying these values to the corresponding testing datasets we obtain the final classification errors presented in Table 2

Dataset	Alphabet size	n-gram	λ_n
Beef	3	1	[0.14897]
		2	[0.059199 0.64354]
		3	[0.18031 0.51181 0.0013076]
	10	1	[1.2358]
		2	[0.97218 0.38007]
		3	[0.80256 0.79378 0.19502]
	20	1	[0.19036]
		2	[0.91433 0.18418]
		3	[0.76518 0.0053491 0.067656]
ECG	3	1	[0.80394]
		2	[0.82499 0.25791]
		3	[0.66397 0.58473 0.54427]
	10	1	[0.74811]
		2	[0.1243 0.55862]
		3	[0.11591 1.7659 0.81072]
	20	1	[0.076999]
		2	[0.022884 0.84058]
		3	[0.027087 0.74227 0.439]
FaceFour	3	1	[0.22144]
		2	[0.21865 0.26487]
		3	[0.22744 0.13813 0.031161]
	10	1	[0.17061]
		2	[0.048042 0.10718]
		3	[0.054286 0.076333 0.11382]
	20	1	[0.030194]
		2	[0.22075 0.10383]
		3	[0.26036 0.053799 0.024754]
OSULeaf	3	1	[0.53548]
		2	[0.66298 0.060555]
		3	[0.65041 0.039146 0.052083]
	10	1	[0.54977]
		2	[0.51024 0.020506]
		3	[0.38619 0.039477 0.19883]
	20	1	[0.34882]
		2	[0.28517 0.025615]
		3	[0.073991 0.020775 0.063922]

Table 3. The optimal values of λ_n obtained by applying ABC to the training datasets.

5 Conclusion

In this paper we presented a new distance metric; ABC-SG, which is applied to strings. This distance considers the frequencies of n-grams, which adds a global level of similarity, in addition to the local one. The particularity of this distance is that it uses the artificial bee colony algorithm to determine the values of its parameters. We tested the new distance and we compared it to a very competitive similarity measure; DTW, on a time series classification task. We showed that our distance ABC-SG gave better results in most cases, despite the difference in complexity.

In order to represent the time series symbolically, we had to use SAX because this is the most widely used symbolic representation method of time series. Nonetheless, a representation technique prepared specifically for ABC-SG may even give better results.

Although we used ABC as an optimization algorithm to set the parameters of the new distance, we think other stochastic and bio-inspired optimization algorithms can also be used with the new distance.

6 References

- Baykasoglu A, Ozbakir L, Tapkan P (2007): Artificial bee colony algorithm and its application to generalized assignment problem. In: *Swarm intelligence focus on ant and particle swarm optimization*. I-Tech Education and Publishing, Vienna, Austria, pp 113-144.
- Berndt, D. and Clifford, J. (1994): Using dynamic time warping to find patterns in time series. In *Proc. AAAI Workshop on Knowledge Discovery in Databases*.
- Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., and Keogh, E. (2008): Querying and mining of time series data: experimental comparison of representations and distance measures. In *Proc of the 34th VLDB*.
- Diwold K, Beekman M, Middendorf M (2010): Honeybee optimisation an overview and a new bee inspired optimisation scheme. In: Hiot LM, Ong YS, Panigrahi BK, Shi Y, Lim MH (eds) *Handbook of swarm intelligence, adaptation, learning, and optimization*, vol 8. Springer, Berlin/Heidelberg, Germany, pp 295–327.
- Guo, AY., and Siegelmann, H(2004): Time-warped longest common subsequence algorithm for music retrieval, in *Proc. ISMIR*.
- Karaboga, D. (2005): An idea based on honey bee swarm for numerical optimization. Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department
- Karaboga, D., Basturk, B. (2007a): A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm, *Journal of Global Optimization* 39 (3) 459–471.
- Karaboga, D., Basturk, B. (2007b) In: *Advances in Soft Computing: Foundations of Fuzzy Logic and Soft Computing*, LNCS, vol. 4529/2007, Springer-Verlag, 2007, pp. 789–798 (Chapter Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems).
- Keogh, E., Chakrabarti, K., Pazzani, M., and Mehrotra (2000): Dimensionality reduction for fast similarity search in large time series databases. *J. of Know. and Inform. Sys.*
- Keogh, E., Zhu, Q., Hu, B., Hao, Y., Xi, X., Wei, L. & Ratanamahatana (2011), The UCR Time Series Classification/Clustering Homepage: www.cs.ucr.edu/~eamonn/time_series_data/ C. A..
- Lin, J., Keogh, E., Lonardi, S., Chiu, B. Y. (2003): A symbolic representation of time series, with implications for streaming algorithms. *DMKD 2003*: 2-11.
- Muhammad Fuad, M.M., Marteau, P.F. (2008a) : Extending the edit distance using frequencies of common characters. *19th International Conference on Database and Expert Systems Applications - DEXA'08*, Turin, Italy, 1-5 September 2008. Lecture Notes in Computer Science, 2008, Volume 5181/2008.
- Muhammad Fuad, M.M., Marteau, P.F. (2008b) : The Extended Edit Distance Metric, *Sixth International Workshop on Content-Based Multimedia Indexing (CBMI 2008)*, London, UK, 18-20th June 2008.
- Muhammad Fuad, M.M., Marteau, P.F. (2008c) : The multi-resolution extended edit distance. *Third International ICST Conference on Scalable Information Systems, Infoscale*,

- Vico Equense, Italy. June 4-6 2008. ACM Digital Library, 2008.
- Parpinelli, R.S., Benitez, C.M.V., Lopes, H.S. (2011): Parallel approaches for the artificial bee colony algorithm. In: Panigrahi, B.K., Shi, Y., Lim, M.H., Hiot, L.M., Ong, Y.S. (eds.) *Handbook of Swarm Intelligence, Adaptation, Learning, and Optimization*, vol. 8, pp. 329–345. Springer, Berlin.
- Wagner, R.A., Fischer, M. J. (1974): The string-to-string correction problem, *Journal of the Association for Computing Machinery*, Vol. 21, No. 1, January 1974, pp. 168-173.
- Yi, B. K., and Faloutsos, C. (2000): Fast time sequence indexing for arbitrary L_p norms. *Proceedings of the 26th International Conference on Very Large Databases*, Cairo, Egypt.
- Zeuzula et al., (2005) :*Similarity Search - The Metric Space Approach*, Springer.

Improving Classifications for Cardiac Autonomic Neuropathy Using Multi-level Ensemble Classifiers and Feature Selection Based on Random Forest

A.V. Kelarev^{1,2}

A. Stranieri¹

J.L. Yearwood¹

J. Abawajy²

H.F. Jelinek³

¹School of Science, Information Technology and Engineering
University of Ballarat, P.O. Box 663, Ballarat, Victoria 3353, Australia
Email: {a.kelarev,a.stranieri,j.yearwood}@ballarat.edu.au

²School of Information Technology,
Deakin University, Victoria 3125, Australia
Email: {kelarev,jemal.abawajy}@deakin.edu.au

³School of Community Health, Charles Sturt University
P.O. Box 789, Albury, NSW 2640, Australia
Email: hjelinek@csu.edu.au

Abstract

This paper is devoted to empirical investigation of novel multi-level ensemble meta classifiers for the detection and monitoring of progression of cardiac autonomic neuropathy, CAN, in diabetes patients. Our experiments relied on an extensive database and concentrated on ensembles of ensembles, or multi-level meta classifiers, for the classification of cardiac autonomic neuropathy progression. First, we carried out a thorough investigation comparing the performance of various base classifiers for several known sets of the most essential features in this database and determined that Random Forest significantly and consistently outperforms all other base classifiers in this new application. Second, we used feature selection and ranking implemented in Random Forest. It was able to identify a new set of features, which has turned out better than all other sets considered for this large and well-known database previously. Random Forest remained the very best classifier for the new set of features too. Third, we investigated meta classifiers and new multi-level meta classifiers based on Random Forest, which have improved its performance. The results obtained show that novel multi-level meta classifiers achieved further improvement and obtained new outcomes that are significantly better compared with the outcomes published in the literature previously for cardiac autonomic neuropathy.

Keywords: Random Forest, ensembles of ensembles, multi-level ensembles, meta classifiers, feature selection, cardiac autonomic neuropathy.

1 Introduction

The investigation of medical applications of data mining is very important and has been considered, for example, in recent articles by Al-Oqaily et al. (2008), Han et al. (2006), Kennedy et al. (2008), Li et al.

(2009), Liang & Zhang (2011), Sinha et al. (2011), Shouman et al. (2011), Sun et al. (2011), Tayebi et al. (2011), Van et al. (2011), where more background information and further references can be found. In particular, valuable information concerning cardiac patients has been obtained using data mining methods, for example, by Cornforth & Jelinek (2007), Han et al. (2006), Jelinek et al. (2010) and Van et al. (2011).

This article is devoted to experimental investigation of several data mining methods for a new application to the study of cardiac autonomic neuropathy (CAN), which is a cardiac condition quite common in diabetes patients. We used an extensive database created by the Diabetes Complications Screening Research Initiative (DiScRi) at Charles Sturt University and concentrated on the particular task of monitoring the progression of cardiac autonomic neuropathy.

First, we compared the performance of many base classifiers for various sets of the most essential features in this database and determined that Random Forest significantly and consistently outperforms all base classifiers in this new application. Second, we used Random Forest feature selection and found a new set of features, which has turned out much better than all sets of features considered previously for CAN in the literature. We verified that Random Forest remained the very best classifier for the new set of features too. Third, we carried out a systematic investigation of various ensemble meta classifiers and found that ensembles based on Random Forest also outperform ensemble meta classifiers based on other classifiers, and that ensemble techniques can be used for further improvement of the performance of Random Forest for this dataset.

Many effective applications of ensemble techniques in data mining have been developed recently. Let us refer, for example, to Ting et al. (2009), Ting et al. (2011), Webb (2008), Webb & Zheng (2004), Yang et al. (2005). In particular, it is well known that various constructions of meta classifiers creating ensembles of base classifiers are capable of improving the stability and effectiveness of classifications.

This article concentrates, in particular, on a systematic empirical investigation of the performance of novel large multi-level meta classifiers for monitoring of CAN progression in diabetes patients. To the best of our knowledge such ensembles of ensembles

or multi-level ensemble meta classifiers have not been considered in the literature before, probably because personal computers routinely used in research have only recently become powerful enough to train them for data sets large enough to justify the use of such large classification systems. On the other hand, the motivation and inspiration for our study originally came from many different multi-stage procedures that had been treated previously, for example, by Christen (2007), Islam & Abawajy (2012), Jiangning et al. (2012) and Madjarov et al. (2011).

Diabetes is a condition requiring continuous everyday monitoring of medical tests to adjust the diet, administer medication, update or modify treatment plans and provide further assistance (Wickramasinghe et al. 2011). These tasks make the development of data mining algorithms for the analysis of test results for diabetes patients particularly valuable. To monitor the progression of a specific clinical condition one has to find a small set of features to be collected and efficient algorithms for the processing of these features.

Experimental research comparing various algorithms applied to particular areas is important, since previous experience of such investigations can be used to guide further implementations and achieve better performance in future practical applications. Indeed, there does not exist a single algorithm that is best for all application domains. The effectiveness of any given category of algorithms depends on the size of a data set, number and types of attributes, and the nature of functional relations and dependencies among the attributes. This is also confirmed by the so-called “no-free-lunch” theorems, which imply that there does not exist one algorithm, which is best for all problems (Wolpert 1996). The present paper concentrates on testing multi-level meta classifiers for the classification of cardiac autonomic neuropathy progression, see Section 3 for details. Our experiments included multi-level meta classifiers combining diverse meta classifiers on two levels. These new results show, in particular, that Random Forest performed best in this setting, and that novel multi-level meta classifiers can be used to achieve further improvement of the classification outcomes for cardiac autonomic neuropathy progression. The multi-level meta classifiers based on Random Forest achieved better performance compared with the results published in the literature (Huda et al. 2010, Kelarev, Dazeley, Stranieri, Yearwood & Jelinek 2012, Kelarev, Stranieri, Yearwood & Jelinek 2012).

The paper is organised as follows. Section 2 describes the Diabetes Complications Screening Research Initiative (DiScRi) organised at Charles Sturt University,² and the corresponding data set. Section 3 contains background information on cardiac autonomic neuropathy. Section 4 deals with the methods used in our experiments. Section 6 presents the experimental results and discussion comparing the efficiencies of several base classifiers and multi-level meta classifiers based on Random Forest for this application. The conclusions are presented in Section 7.

2 Diabetes Complications Screening Research Initiative

In order to investigate the data mining algorithms for diabetes patients, we used a large database of test results and health-related parameters collected at the Diabetes Complications Screening Research Initiative (DiScRi) organised at Charles Sturt University (Cornforth & Jelinek 2007). Many patients suffering

from diabetes develop complications that require 24/7 cardiac monitoring.

The collection and analysis of data in the project has been approved by the Ethics in Human Research Committee of the university. The participants were instructed not to smoke and refrain from consuming caffeine containing drinks and alcohol for 24 hours preceding the tests as well as to fast from midnight of the previous day until tests were complete. The measurements were conducted from 9:00am until 12mid-day and were recorded in the DiScRi database along with various other clinical data including age, sex and diabetes status, blood pressure (BP), body-mass index (BMI), blood glucose level (BGL), and cholesterol profile. Reported incidents of a heart attack, atrial fibrillation and palpitations were also recorded. The most important set of features recorded for CAN determination is the *Ewing battery* (Ewing et al. 1980, 1985). There are five Ewing tests in the battery: changes in heart rate associated with lying to standing, deep breathing and valsalva manoeuvre and changes in blood pressure associated with hand grip and lying to standing. In addition features from the ten second samples of 12-lead ECG for all participants were extracted from the database. These included the QRS, PQ, QTc and QTd intervals, heart rate and QRS axis explained below. The QRS complex reflects the depolarization of the ventricles of the heart. The duration of the QRS complex is called the QRS duration. The time from the beginning of the P wave until the start of the next QRS complex is called the PQ interval. The longest distance from the Q wave to the next T wave is called the QT interval. The period from the beginning of the QRS complex to the end of the T wave is denoted by QT interval, which if corrected for heart rate becomes the QTc. It represents the so-called refractory period of the heart. The difference of the maximum QT interval and the minimum QT interval over all 12 leads is known as the QT dispersion denoted by QTd. It is used as an indicator of repolarisation of ventricular. The deflection of the electrical axis of the heart measured in degrees to the right or left is called the QRS axis.

Several expert editing rules were used to reduce the number of missing values in the database. These rules were collected during discussions with the experts maintaining the database. Preprocessing of data using these rules produced 1029 complete rows with complete values of all fields, which were used for the experimental evaluation of the performance of data mining algorithms. The whole database contained over 200 features.

3 Cardiac Autonomic Neuropathy

Cardiac autonomic neuropathy (CAN) is a condition associated with damage to the autonomic nervous system innervating the heart (Ewing et al. 1980, 1985, Khandoker et al. 2009). The classification of disease progression associated with CAN is important, because it has implications for planning of timely treatment, which can lead to an improved well-being of the patients and a reduction in morbidity and mortality associated with cardiac arrhythmias in diabetes. The most important tests required for identification of CAN rely on assessing responses in heart rate and blood pressure to various activities, usually consisting of tests described by Ewing et al. (1980, 1985): lying to standing heart rate change (LSHR), deep breathing heart rate change (DBHR), valsalva manoeuvre heart rate change (VAHR), hand grip blood pressure change (HGBP), lying to standing blood pres-

sure change (LSBP). QRS width has also been shown to be indicative of CAN (Fang et al. 2004) and is also included. For discussion of the outcomes of our experiments we use the acronyms for the DiScRi features listed in Figure 1. The same acronyms are used in the original DiScRi database.

Acronym	Feature
LSHR	Lying to standing heart rate change
LSHRresu	Categorical variable based on LSHR defined by Ewing et al. (1980)
DBHR	Deep breathing heart rate change
DBHRresu	Categorical variable based on LSHR defined by Ewing et al. (1980)
VAHR	Valsalva manoeuvre heart rate change
VAHRresu	Categorical variable based on LSHR defined by Ewing et al. (1980)
HGBP	Hand grip blood pressure change
LSBP	Lying to standing blood pressure change
QRSaxis	QRS axis degree 10sec
SLHR	Standing to lying heart rate
QRS 10sec	QRS duration
PQ 10sec	PQ duration
QTc 10sec	Corrected QT interval duration
QTd 10sec	QT dispersion

Figure 1: Acronyms for several features in DiScRi database

We investigated three original classifications of cardiac autonomic neuropathy progression introduced by Ewing et al. (1980, 1985). They have 2, 3 and 4 classes, respectively. The first one divides all patients into two classes allocating each patient either to the ‘normal’ class, or to ‘definite’ class. The second one divides all patients into three classes allocating each patient to one of the following classes: ‘normal’, ‘early’, ‘definite’. The fourth classification divides all patients into four classes, allocated each patient to one of the following classes: ‘normal’, ‘early’, ‘definite’, and ‘severe’.

4 Methods

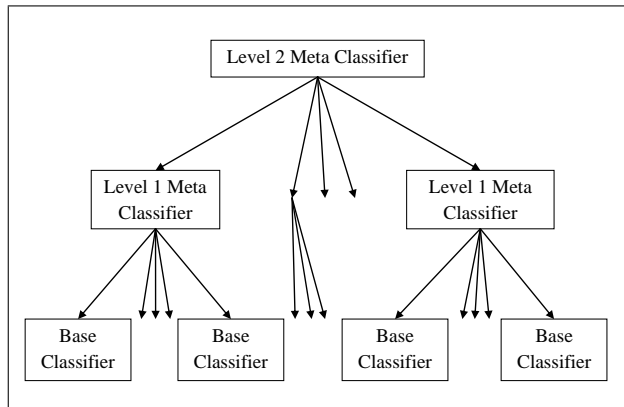


Figure 2: Multi-level meta classifiers

4.1 Random Forest

Random Forest plays a special role in this paper, and so we introduce it in a separate subsection. Random Forest is an ensemble meta classifier hardwired to a particular base classifier, Random Tree. It constructs a forest of random trees following Breiman (2001) building many decision tree predictors with randomly selected variable subsets and utilizing 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). We used Random Forest feature selection in R (version 2.15.1) with Rattle (Williams 2009, 2011). Weka implementation of Random Forest was used to combine it with other meta classifiers available in Weka via SimpleCLI. (This implementation can handle missing values.) In applying the Random Forest classifier and its feature selection we followed the recommendations and conclusions based on previous experiments for a different database of cardiac patients presented by Van et al. (2011).

4.2 Base Classifiers

We tested many preliminary base classifiers available in Weka (Hall et al. 2009) and have chosen the following classifiers for a series of complete tests with outcomes presented in this paper. These robust classifiers performed well for DiScRi data set during our initial testing. They represent several essential categories of classifiers.

- *DecisionTable* builds and uses a decision table majority classifier (Kohavi 1995).
- *FURIA* is a fuzzy unordered rule induction algorithm due to Huehn & Huellermeier (2009).
- *J48* generates a pruned or unpruned C4.5 decision tree (Quinlan 1993).
- *NBTree* uses a decision tree with naive Bayes classifiers at the leaves (Kohavi 1996).
- *SMO* uses Sequential Minimal Optimization for training a support vector classifier (Hastie & Tibshirani 1998, Keerthi et al. 2001, Platt 1998). Initially, we tested all kernels of SMO available in Weka and used it with polynomial kernel that performed best for our data set.

4.3 Meta Classifiers

We investigate the performance of the following meta classifiers: Bagging, Boosting, Dagging, Decorate, Grading, HBGF, MultiBoost and Stacking.

- *Bagging* (bootstrap aggregating), generates a collection of new sets by resampling the given training set at random and with replacement. These sets are called *bootstrap samples*. New classifiers are then trained, one for each of these new training sets. They are amalgamated via a majority vote (Breiman 1996, Liang & Zhang 2011).
- *Boosting* trains several classifiers in succession. Every next classifier is trained on the instances that have turned out more difficult for the preceding classifier. To this end all instances are assigned weights, and if an instance turns out difficult to classify, then its weight is increased

at the next boosting step. We used highly successful AdaBoost classifier described by Freund & Schapire (1996).

- *Consensus functions* can be used as a replacement for voting to combine the outputs of classifiers in the ensemble. Here we used the HBGF consensus function, following the recommendations of Fern & Brodley (2004) and our previous experience with consensus functions for other data sets (Yearwood et al. 2009). It utilizes a bipartite graph with two sets of vertices: clusters and elements of the data set. A cluster C and an element d are connected by an edge in this bipartite graph if and only if d belongs to C . An appropriate graph partitioning algorithm is then applied to the whole bipartite graph, and the final clustering is determined by the way it partitions all elements of the data set.
- *Dagging* is useful in situations where the base classifiers are slow. It divides the training set into a collection of disjoint (and therefore smaller) stratified samples, trains copies of the same base classifier and averages their outputs using vote (Ting & Witten 1997).
- *Decorate* constructs special artificial training examples to build diverse base classifiers (Melville & Mooney 2005).
- *Grading* trains meta-classifiers, which grade the output of base classifiers as correct or wrong labels, and these graded outcomes are then combined (Seewald & Fuernkranz 2001).
- *MultiBoost* extends the approach of AdaBoost with the wagging technique (Webb 2000). Wagging is a variant of bagging where the weights of training instances generated during boosting are utilized in selection of the bootstrap samples (Bauer & Kohavi 1999).
- *Stacking* can be regarded as a generalization of voting, where meta-learner aggregates the outputs of several base classifiers (Wolpert 1992).

4.4 Multi-level Meta Classifiers

The main focus of this paper is on a systematic investigation of several novel multi-level meta classifiers for DiScRi data set. These classifiers have not been considered in the literature before, since personal computers regularly used in research have only recently become powerful enough to train them for large data sets. It turns out easy to set up and use these multi-level meta classifiers in Weka SimpleCLI command line. To demonstrate how such classifiers can be set up and executed, we include Figure 7 with complete commands used in SimpleCLI to run two very best options in our experiments and Figure 8, which shows how to enter these commands, see also Section 6. Our experiments compared several multi-level meta classifiers with two levels and various base classifiers. However, the best results were obtained by classifiers, which can also be viewed as multi-level meta classifiers with three levels of ensembles, since they are based on Random Forest, see Section 6 and Subsection 4.1.

5 Measures of Performance of Classifiers

We looked at several standard measures of performance of classifiers: Area Under Curve, accuracy,

precision, recall, sensitivity and specificity. Following Van et al. (2011), we used the Area Under Curve, AUC, as the main measure of performance of classifiers. It is also known as the Receiver Operating Characteristic or ROC area. Let us refer to Van et al. (2011) for more detailed discussion of measures of performance and references to other relevant articles. In the rare cases where two classifiers produced equal AUC, or where one classifier performed with the same AUC for two sets of features, to finetune the ordering of such cases we used accuracy of the classification as the second important metric to guide our experiments. The tables presenting the results of our experiments in this paper contain only AUC as the main measure of performance.

Here we include only a brief overview of the measures we used conducting our experiments, since there is a variety of terms used to discuss clinical experiments. Notice that for multi-class classifiers, like those considered in the present article, weighted average values of the performance metrics are usually used. This means that they are calculated for each class separately, and a weighted average is found then. In particular, our tables in this paper include the weighted average values of AUC over all classes. In contrast, the *accuracy* is defined for the whole classifier as the percentage of all patients classified correctly, which means that this definition does not involve weighted averages in the calculation. The accuracy can be expressed as the probability that the prediction of the classifier for an individual patient is correct.

The *Area Under Curve*, AUC, for a given class, is an area under the ROC graph that plots true positive rates for this class against false positive rates for a series of cut-off values. Equivalently, the ROC graph can be defined as a curve graphically displaying the trade-off between sensitivity and specificity for each cut-off value. *Sensitivity* is the proportion of positives (patients with CAN) that are identified correctly. *Specificity* is the proportion of negatives (patients without CAN) that are identified correctly.

Sensitivity and specificity are measures evaluating binary classifications. For multi-class classifications they can be also used with respect to one class and its complement. Sensitivity is also called *True Positive Rate*. *False Positive Rate* is equal to $1 - \text{specificity}$. These measures are related to recall and precision. *Precision* of a classifier, for a given class, is the ratio of true positives to combined true and false positives. *Recall* is the ratio of true positives to the number of all positive samples (i.e., to the combined true positives and false negatives). The recall calculated for the class of patients with CAN is equal to sensitivity of the whole classifier.

For example, in the case of the two-class classification of CAN For the class of patients with CAN, the precision is the ratio of the number of patients correctly identified as having CAN to the number of all patients identified as having CAN. For the cohort of patients without CAN, the precision is the ratio of the number of patients correctly identified as having no CAN to the number of all patients identified as free from CAN. The precision of the classifier as a whole is a weighted average of its precisions for these classes.

Likewise, for the class of patients with CAN, the recall is the ratio of the number of patients correctly identified as having CAN to the number of all patients with CAN. For the cohort of patients without CAN, the recall is the ratio of the number of patients correctly identified as being free from CAN to the number of all patients without CAN. The recall of

the classifier is a weighted average of its recalls for both classes.

6 Experiments and Discussion

We used Rattle (Williams 2009) and R software (Williams 2011) for Random Forest feature selection, and Weka SimpleCLI command line to train and test classifiers and meta classifiers. One of the standard options for preventing overfitting is 10-fold cross validation. It is implemented in Weka and is invoked in SimpleCLI by default as stratified 10-fold cross validation. (Default stratified 10-fold cross validation can be switched off or modified by indicating command line arguments -no-cv, -split-percentage and -preserve-order in SimpleCLI.) It divides data into ten stratified folds and creates training sets and hold out testing sets ten times for ten consecutive tests with hold out sets automatically. Thus, we used 10-fold cross validation to assess the performance of various base classifiers, meta-classifiers and multi-level meta classifiers. All tables with outcomes included in this paper contain average performance against the validate sets found in stratified 10-fold cross validation.

First, we tested the performance of DecisionTable, FURIA, J48, NBTree, RandomForest and SMO for all subsets of the Ewing battery, which is the set of the most important features. All of these base classifiers are available in Weka Explorer, and we use Weka to test them. These experiments demonstrated that Random Forest consistently outperformed all other classifiers for all of these subsets of features. To illustrate these results, we include only Table 1.

	Number of classes		
	2	3	4
DecisionTable	0.905	0.900	0.897
FURIA	0.942	0.936	0.932
J48	0.933	0.930	0.922
NBTree	0.923	0.917	0.914
RandomForest	0.982	0.977	0.973
SMO	0.861	0.856	0.854

Table 1: AUC of base classifiers for the subset LSHR, DBHR, VAHR, LSBP of Ewing features

Then we used Random Forest feature selection in Rattle (Williams 2009). It produced feature ranking presented in Figure 3. We tested the performance of Random Forest for all sets beginning with the most significant feature and adding more features in the order of their significance. This demonstrated that the best set of features consists of the first 8 attributes: DBHR, VAHRresu, VAHR, DBHRresu, LSHRresu, LSHR, HGBP, QRS axis (degree) 10sec.

We tested all base classifiers for this set of 8 features too. The results of these experiments are given in Table 2 and Figure 4. The outcomes show that Random Forest remains the best classifier for this set of attributes too. Thus, in all our tests Random Forest has consistently performed as the very best base classifier for all sets of features of DiScRi database. We see that Random Forest feature selection has made it possible to improve the outcomes significantly.

Next, we used SimpleCLI command line in Weka to investigate the performance of meta classifiers in

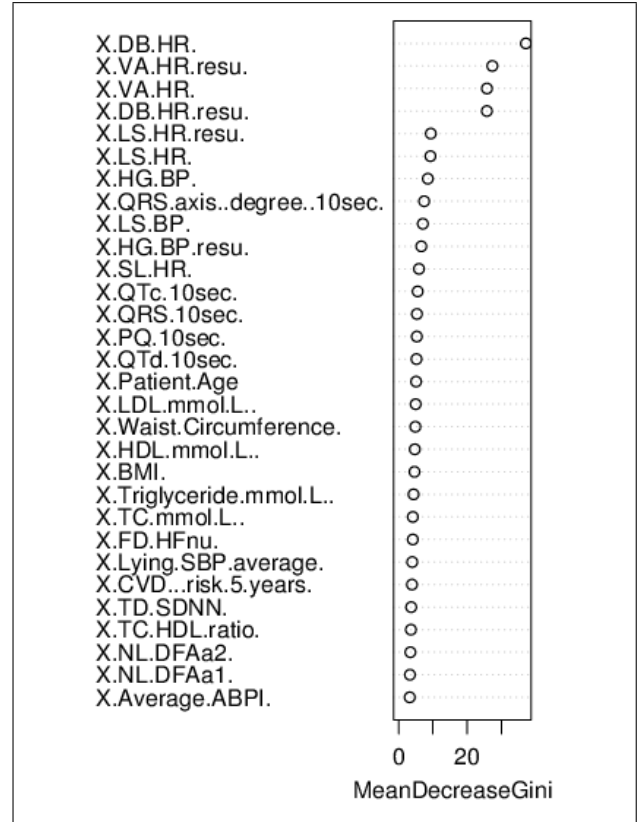


Figure 3: Random Forest feature ranking

	Number of classes		
	2	3	4
DecisionTable	0.963	0.960	0.956
FURIA	0.979	0.976	0.968
J48	0.967	0.964	0.959
NBTree	0.980	0.975	0.972
RandomForest	0.990	0.987	0.981
SMO	0.954	0.949	0.944

Table 2: AUC of base classifiers for the best subset DBHR, VAHRresu, VAHR, DBHRresu, LSHRresu, LSHR, HGBP, QRS axis (degree) 10sec.

their ability to achieve further improvement to performance. We tested the following meta classifiers: AdaBoost, Bagging, Dagging, Decorate, Grading, HBGF, MultiBoost, and Stacking. Our tests have also shown that the outcomes remained better when Random Forest was used as a base classifier for these meta classifiers and that the results became worse when Random Forest was replaced by other base classifiers. We conducted complete set of evaluations of the meta classifiers based on Random Forest. These results are included in Table 3 and Figure 5. We see that AdaBoost, Bagging, Decorate and MultiBoost performed better than other meta classifiers.

Finally, for the four meta classifiers that performed well in the previous step, we investigated all their multi-level combinations. The experimental results comparing the performance of multi-level meta classifiers are presented in Table 4 and Figure 6. To provide more details on how these multi-level classifiers can be set up and executed, we include Figure 7 with

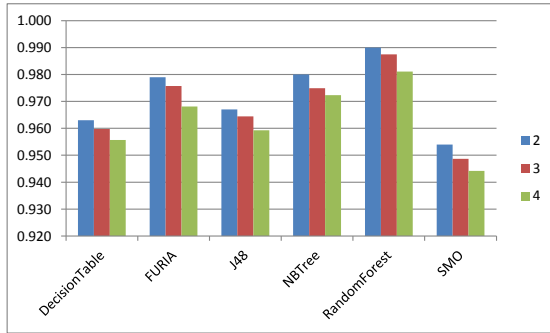


Figure 4: Base classifiers

	Number of classes		
	2	3	4
AdaBoost	0.987	0.984	0.979
Bagging	0.995	0.990	0.987
Dagging	0.971	0.967	0.964
Decorate	0.996	0.993	0.988
Grading	0.969	0.965	0.960
HBGF	0.974	0.969	0.967
MultiBoost	0.988	0.984	0.981
Stacking	0.978	0.976	0.969

Table 3: AUC of meta classifiers based on Random Forest

SimpleCLI command line arguments used to run two very best options given in our Table 4. These results show that several multi-level combinations of ensemble classifiers made additional improvements and produced very good outcomes in Table 4. The very best result was obtained by two options combining Bagging and Decorate into one multi-level ensemble classifier. In the first option Bagging was used in the 2nd level after applications of Decorate based on Random Forest in the first level. In the second option Decorate was used in the 2nd level to combine the results of Bagging applied to Random Forest as a base classifier.

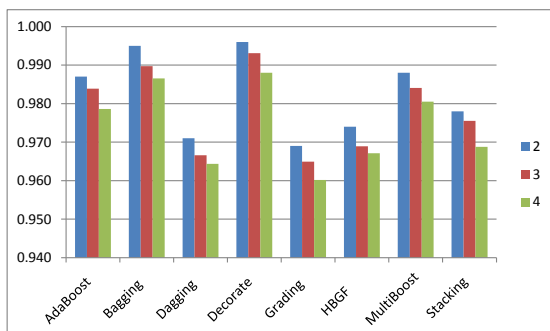


Figure 5: Meta classifiers based on Random Forest

Level 2	Level 1	Number of classes		
		2	3	4
AdaBoost	Bagging	0.990	0.987	0.984
AdaBoost	Decorate	0.992	0.988	0.986
AdaBoost	MultiBoost	0.989	0.987	0.982
Bagging	AdaBoost	0.994	0.990	0.987
Bagging	Decorate	0.997	0.993	0.990
Bagging	MultiBoost	0.994	0.992	0.988
Decorate	AdaBoost	0.996	0.992	0.989
Decorate	Bagging	0.997	0.994	0.990
Decorate	MultiBoost	0.996	0.992	0.990
MultiBoost	AdaBoost	0.989	0.986	0.983
MultiBoost	Bagging	0.985	0.982	0.979
MultiBoost	Decorate	0.990	0.987	0.983

Table 4: AUC of multi-level meta classifiers based on Random Forest

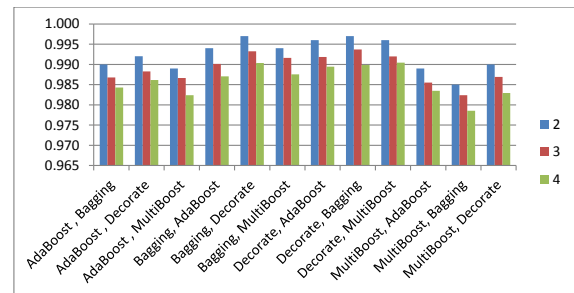


Figure 6: Multi-level meta classifiers

2 levels	SimpleCLI command line
Decorate, Bagging	java weka.classifiers.meta.Decorate -E 10 -R 1.0 -S 1 -I 10 -W weka.classifiers.meta.Bagging -P 100 -S 1 -num-slots 1 -I 10 -W weka.classifiers.trees.RandomForest -I 10 -K 0 -S 1
Bagging, Decorate	java weka.classifiers.meta.Bagging -P 100 -S 1 -num-slots 1 -I 10 -W weka.classifiers.meta.Decorate -E 10 -R 1.0 -S 1 -I 10 -W weka.classifiers.trees.RandomForest -I 10 -K 0 -S 1

Figure 7: SimpleCLI command lines with parameters of two best multi-level meta classifiers

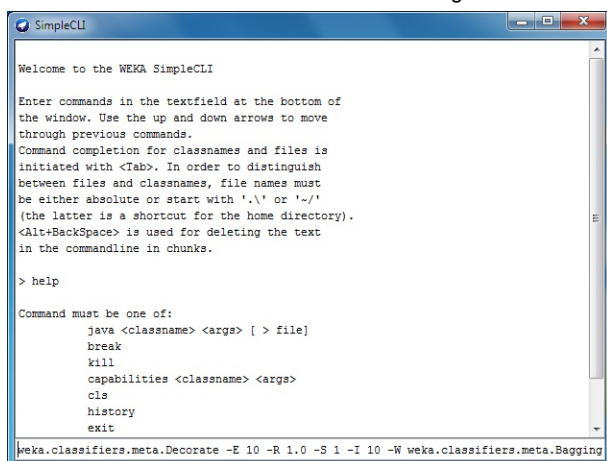


Figure 8: Multi-level meta classifier in SimpleCLI

7 Conclusion

Our experiments demonstrated that for DiScRi data set Random Forest consistently produced better outcomes than all other base classifiers. Feature selection based on the ranking obtained by the implementation of Random Forest in Rattle further improved the outcomes of all classifiers, and again Random Forest produced the best outcomes for the set of features obtained. Finally, the results show that meta classifiers and multi-level ensemble meta classifiers can be used to improve the classifications even more. The best outcomes have been obtained by the novel combined multi-level ensemble classifiers combining Bagging and Decorate based on Random Forest. These methods can be recommended for the monitoring of cardiac autonomic neuropathy progression in those situations where the energy and memory used are not an issue. In situations where it is very important to conserve the energy and use less memory, as it is the case for example, in mobile applications, then Random Forest can be recommended, since it has also produced excellent outcomes.

DiScRi is a very large and unique data set containing a comprehensive collection of tests related to CAN. Using Random Forest feature selection and multi-level meta classifiers has made it possible to achieve a serious improvement in performance compared with outcomes obtained in previous publications using only basic decision trees for classification.

The level of performance of multi-level classifiers for DiScRi data set is also quite good in comparison with the outcomes obtained recently for other data sets in closely related areas using different methods, for example, by Kang et al. (2006), Kelarev et al. (2006), Jelinek et al. (2010, 2011), Yearwood et al. (2008).

In conclusion, let us note that Random Forest is also an ensemble classifier hard wired to a particular base classifier, Random Tree. Therefore, in fact the multi-level meta classifiers included in Table 4 can be considered as ensemble classifiers with three levels where ensemble methods are used.

Acknowledgements

This work was supported by a Deakin-Ballarat collaboration grant.

The authors are grateful to three referees for comments and corrections that have helped to improve the paper.

References

- Al-Oqaily, A., Kennedy, P., Catchpoole, D. & Simoff, S. (2008), Comparison of visualization methods of genome-wide SNP profiles in childhood acute lymphoblastic leukaemia, in J. Roddick, J. Li, P. Christen & P. Kennedy, eds, 'Data Mining and Analytics 2008, Proceedings of the Seventh Australasian Data Mining Conference, AusDM 2008', Vol. 87 of *CRPIT*, ACS, Glenelg, South Australia, pp. 111–121.
- Bauer, E. & Kohavi, R. (1999), 'An empirical comparison of voting classification algorithms: Bagging, boosting, and variants', *Machine Learning* **36**, 105–139.
- Breiman, L. (1996), 'Bagging predictors', *Machine Learning* **24**, 123–140.
- Breiman, L. (2001), 'Random Forests', *Machine Learning* **45**, 5–32.
- Christen, P. (2007), A two-step classification approach to unsupervised record linkage, in P. Christen, P. Kennedy, J. Li, I. Kolyshkina & G. Williams, eds, 'Sixth Australasian Data Mining Conference (AusDM 2007)', Vol. 70 of *CRPIT*, ACS, Gold Coast, Australia.
- Cornforth, D. & Jelinek, H. (2007), 'Automated classification reveals morphological factors associated with dementia', *Applied Soft Computing* **8**, 182–190.
- Ewing, D., Campbell, J. & Clarke, B. (1980), 'The natural history of diabetic autonomic neuropathy', *Q. J. Med.* **49**, 95–100.
- Ewing, D., Martyn, C., Young, R. & Clarke, B. (1985), 'The value of cardiovascular autonomic function tests: 10 years experience in diabetes', *Diabetes Care* **8**, 491–498.
- Fang, Z., Prins, J. & Marwick, T. (2004), 'Diabetic cardiomyopathy: evidence, mechanisms, and therapeutic implications', *Endocr. Rev.* **25**, 543–567.
- Fern, X. & Brodley, C. (2004), Solving cluster ensemble problems by bipartite graph partitioning, in '21st International Conference on Machine Learning, ICML'04', Vol. 69, ACM, New York, NY, USA, pp. 36–43.
- Freund, Y. & Schapire, R. (1996), Experiments with a new boosting algorithm, in 'Proc. 13th Internat. Conf. Machine Learning', pp. 148–156.
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. & Witten, I. (2009), 'The WEKA data mining software: an update', *SIGKDD Explorations* **11**, 10–18.
- Han, L., Embrechts, M., Szymanski, B., Sternickel, K. & Ross, A. (2006), Random forests feature selection with K-PLS: detecting ischemia from magnetocardiograms, in 'Proc. European Symposium on Artificial Neural Networks, ESANN', Vol. 14, pp. 221–226.
- Hastie, T. & Tibshirani, R. (1998), Classification by pairwise coupling, in 'Advances in Neural Information Processing Systems'.

- 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 'Sixth International Conference on Intelligent Sensors, Sensor Networks and Information Processing, ISSNIP 2010', pp. 297–302.
- Huehn, J. & Huellermeier, E. (2009), 'FURIA: An algorithm for unordered fuzzy rule induction', *Data Mining and Knowledge Discovery* **19**, 293–319.
- Islam, R. & Abawajy, J. (2012), 'A multi-tier phishing detection and filtering approach', *Journal of Network and Computer Applications* p. to appear soon.
- Jelinek, H., Khandoker, A., Palaniswami, M. & McDonald, S. (2010), 'Heart rate variability and QT dispersion in a cohort of diabetes patients', *Computing in Cardiology* **37**, 613–616.
- Jelinek, H., Rocha, A., Carvalho, T., Goldenstein, S. & Wainer, J. (2011), Machine learning and pattern classification in identification of indigenous retinal pathology, in 'Proceedings IEEE Conference Eng. Med. Biol. Soc.', pp. 5951–5954.
- Jiangning, S., Tan, H., Wang, M., Webb, G. & Akutsu, T. (2012), 'TANGLE: Two-level support vector regression approach for protein backbone torsion angle prediction from primary sequences', *PLoS ONE* **7**, e30361.
- Kang, B., Kelarev, A., Sale, A. & Williams, R. (2006), A new model for classifying DNA code inspired by neural networks and FSA, in 'Advances in Knowledge Acquisition and Management', Vol. 4303 of *Lecture Notes in Computer Science*, pp. 187–198.
- Keerthi, S., Shevade, S., Bhattacharyya, C. & Murthy, K. (2001), 'Improvements to Platt's SMO algorithm for SVM classifier design', *Neural Computation* **13**(3), 637–649.
- Kelarev, A., Dazeley, R., Stranieri, A., Yearwood, J. & Jelinek, H. (2012), Detection of CAN by ensemble classifiers based on Ripple Down Rules, in 'Pacific Rim Knowledge Acquisition Workshop, PKAW2012', Vol. 7457 of *Lecture Notes in Artificial Intelligence*, pp. 147–159.
- Kelarev, A., Kang, B. & Steane, D. (2006), Clustering algorithms for ITS sequence data with alignment metrics, in 'AI 2006: Advances in Artificial Intelligence, 19th Australian Joint Conference on Artificial Intelligence', Vol. 4304 of *Lecture Notes in Artificial Intelligence*, pp. 1027–1031.
- Kelarev, A., Stranieri, A., Yearwood, J. & Jelinek, H. (2012), Empirical study of decision trees and ensemble classifiers for monitoring of diabetes patients in pervasive healthcare, in 'Network-Based Information Systems, NBIS-2012', pp. 441–446.
- Kennedy, P., Simoff, S., Catchpoole, D., Skillicorn, D., Ubaudi, F. & Al-Oqaily, A. (2008), Integrative visual data mining of biomedical data: Investigating cases in chronic fatigue syndrome and acute lymphoblastic leukaemia, in 'Visual Data Mining', Vol. 4404/2008 of *Lecture Notes in Computer Science*, pp. 367–388.
- Khandoker, A., Jelinek, H. & Palaniswami, M. (2009), 'Identifying diabetic patients with cardiac autonomic neuropathy by heart rate complexity analysis', *BioMedical Engineering OnLine* **8**, <http://www.biomedical-engineering-online.com/content/8/1/3>.
- Kohavi, R. (1995), The power of decision tables, in '8th European Conference on Machine Learning', pp. 174–189.
- Kohavi, R. (1996), Scaling up the accuracy of Naive-Bayes classifiers: A Decision-Tree hybrid, in 'Second International Conference on Knowledge Discovery and Data Mining', pp. 202–207.
- Li, J., Fu, A. & Fahey, P. (2009), 'Efficient discovery of risk patterns in medical data', *Artificial Intelligence in Medicine* **45**, 77–89.
- Liang, G. & Zhang, C. (2011), Empirical study of bagging predictors on medical data, in P. Vamplew, A. Stranieri, K.-L. Ong, P. Christen & P. J. Kennedy, eds, 'Australasian Data Mining Conference, AusDM 2011', Vol. 121 of *CRPIT*, ACS, Ballarat, Australia, pp. 31–40.
- Madjarov, G., Gjorgjevikj, D. & Delev, T. (2011), Efficient two stage voting architecture for pairwise multi-label classification, in J. Li, ed., 'AI 2010: Advances in Artificial Intelligence', Vol. 6464 of *Lecture Notes in Artificial Intelligence*, pp. 164–173.
- Melville, P. & Mooney, R. (2005), 'Creating diversity in ensembles using artificial data', *Information Fusion* **6**, 99–111.
- Platt, J. (1998), Fast training of support vector machines using sequential minimal optimization, in 'Advances in Kernel Methods – Support Vector Learning'.
- Quinlan, R. (1993), *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Mateo, CA.
- Seewald, A. & Fuernkranz, J. (2001), An evaluation of grading classifiers advances in intelligent data analysis, in 'Advances in Intelligent Data Analysis', Vol. 2189/2001 of *Lecture Notes in Computer Science*, pp. 115–124.
- Shouman, M., Turner, T. & Stocker, R. (2011), Using decision tree for diagnosing heart disease patients, in P. Vamplew, A. Stranieri, K.-L. Ong, P. Christen & P. Kennedy, eds, 'Australasian Data Mining Conference, AusDM 11', Vol. 121 of *CRPIT*, ACS, Ballarat, Australia, pp. 23–30.
- Sinha, A., Tayebi, H., Krishnaswamy, S., Waluyo, A. & Gaber, M. (2011), Resource-aware ECG analysis on mobile devices, in 'Proceedings of the 2011 ACM Symposium on Applied Computing, SAC11', pp. 1012–1013.
- Sun, B., Zhu, Z., Li, J. & Linghu, B. (2011), 'Combined feature selection and cancer prognosis using support vector machine regression', *IEEE/ACM Transactions on Computational Biology Bioinformatics* **8**, 1671–1677.
- Tayebi, H., Krishnaswamy, S., Waluyo, A., Sinha, A. & Gaber, M. (2011), RA-SAX: Resource-aware symbolic aggregate approximation for mobile ECG analysis, in 'Proceedings of the 12th IEEE International Conference on Mobile Data Management, MDM11', Vol. 1, pp. 289–290.
- Ting, K., Wells, J., Tan, S., Teng, S. & Webb, G. (2009), FaSS: Ensembles for stable learners, in 'MCS 2009: 8th International Workshop on Multiple Classifier Systems', Vol. 5519 of *Lecture Notes in Computer Science*, pp. 364–374.

- Ting, K., Wells, J., Tan, S., Teng, S. & Webb, G. (2011), 'Feature-subspace aggregating: Ensembles for stable and unstable learners', *Machine Learning* **82**(3), 375–397.
- Ting, K. & Witten, I. (1997), Stacking bagged and dagged models, in 'Fourteenth international Conference on Machine Learning', pp. 367–375.
- Van, A., Gay, V., Kennedy, P., Barin, E. & Leijdekkers, P. (2011), Understanding risk factors in cardiac rehabilitation patients with random forests and decision trees, in P. Vamplew, A. Stranieri, K.-L. Ong, P. Christen & P. Kennedy, eds, 'Australasian Data Mining Conference, AusDM 2011', Vol. 121 of *CRPIT*, ACS, Ballarat, Australia, pp. 11–22.
- Webb, G. (2000), 'Multiboosting: A technique for combining boosting and wagging', *Machine Learning* **40**, 159 – 196.
- Webb, G. (2008), Multi-strategy ensemble learning, ensembles of bayesian classifiers, and the problem of false discoveries, in J. Roddick, J. Li, P. Christen & P. Kennedy, eds, 'Data Mining and Analytics 2008, Proceedings of the Seventh Australasian Data Mining Conference, AusDM 2008', Vol. 87 of *CRPIT*, ACS, Glenelg, South Australia, p. 15.
- Webb, G. & Zheng, Z. (2004), 'Multistrategy ensemble learning: Reducing error by combining ensemble learning techniques', *IEEE Transactions on Knowledge and Data Engineering* **16**, 980–991.
- Wickramasinghe, K., Alahakoon, D., Schattner, P. & Georgeff, M. (2011), Self-organizing maps for translating health care knowledge: A case study in diabetes management, in D. Wang & M. Reynolds, eds, 'AI 2011: Advances in Artificial Intelligence', Vol. 7106 of *Lecture Notes in Artificial Intelligence*, pp. 162–171.
- Williams, G. (2009), 'Rattle: a data mining GUI for R', *The R Journal* **1**, 45–55.
- Williams, G. (2011), *Data Mining with Rattle and R: The Art of Excavating Data for Knowledge Discovery (Use R!)*, Springer, New York, Dordrecht, Heidelberg, London.
- Wolpert, D. (1992), 'Stacked generalization', *Neural Networks* **5**, 241–259.
- Wolpert, D. (1996), 'The lack of a priori distinctions between learning algorithms', *Neural Computation* **8**, 1341–1390.
- Yang, Y., Korb, K., Ting, K.-M. & Webb, G. (2005), Ensemble selection for superparent-one-dependence estimators, in 'AI 2005: Advances in Artificial Intelligence, 18th Australasian Joint Conference on Artificial Intelligence', Vol. 3809 of *Lecture Notes in Artificial Intelligence*, pp. 102–111.
- Yearwood, J., Kang, B. & Kelarev, A. (2008), Experimental investigation of classification algorithms for ITS dataset, in 'Pacific Rim Knowledge Acquisition Workshop, PKAW 2008', Hanoi, Vietnam, 15–16 December 2008, pp. 262–272.
- Yearwood, J., Webb, D., Ma, L., Vamplew, P., Ofoghi, B. & Kelarev, A. (2009), Applying clustering and ensemble clustering approaches to phishing profiling, in P. Kennedy, K. Ong & P. Christen, eds, 'Data Mining and Analytics 2009, Proc. 8th Australasian Data Mining Conference, AusDM 2009', Vol. 101 of *CRPIT*, ACS, Melbourne, Australia, pp. 25–34.

Combining Classifiers in Multimodal Affect Detection

M. S. Hussain, Hamed Monkaresi and Rafael A. Calvo

School of Electrical and Information Engineering, The University of Sydney, NSW 2006, Australia

{Sazzad.Hussain, Hamed.Monkaresi, Rafael.Calvo}@sydney.edu.au

Abstract

Affect detection where users' mental states are automatically recognized from facial expressions, speech, physiology and other modalities, requires accurate machine learning and classification techniques. This paper investigates how combined classifiers, and their base classifiers, can be used in affect detection using features from facial video and multichannel physiology. The base classifiers evaluated include function, lazy and decision trees; and the combined where implemented as vote classifiers. Results indicate that the accuracy of affect detection can be improved using the combined classifiers especially by fusing the multimodal features. The base classifiers that are more useful for certain modalities have been identified. Vote classifiers also performed best for most of the individuals compared to the base classifiers.

Keywords: Classifiers, machine learning, affective computing, data fusion.

1 Introduction

Affective computing, mostly useful in the area of human computer interaction (HCI), and particularly affect detection, heavily depends on efficient machine learning techniques (Calvo and D'Mello, 2010). Various modalities such as behavioural signatures and physiological patterns can be indicators of affect, thus pattern recognition techniques applied to a single, but mostly a combination of modalities could lead to affect detection.

A number of techniques have been developed for affect detection and studies tend to use features from audio-visual, speech-text, dialog-posture, face-body-speech, and speech-physiology, face-physiology, and multi-channel physiology (for detailed review see (Calvo and D'Mello, 2010)). Most of these studies have applied single classifiers, such as support vector machines (SVM), k-nearest neighbours (KNN), linear/quadratic discriminant analysis (LDA/QDA), decision trees, Bayesian network etc. with single and multiple modalities (mostly as feature fusion). However, finding a single classifier that works well for all modalities and individuals is difficult. Even though decision level fusion approaches have been proposed for integrating

multimodal information in affect detection, in most studies it could not exceed the performance of feature level fusion (Sebe et al., 2005).

Combining classifiers is thought to provide more accurate and efficient classification results. Instead of just one classifier, a subset of classifiers (aka base classifiers) can be considered along with the best subset of features for the best combination (Kuncheva, 2004). Moreover, a certain base classifier may do well on a certain modality, but it is challenging to generalize one classifier for multiple channels or modalities. There are two important reasons for considering combined classifiers (Utthara et al., 2010): (1) A single classifier can not perform well when the nature of features are different. Using combination of classifiers with a subset of features may provide a better performance. (2) To improve generalization, where a classifier may not perform well for new data beyond that in the training set –generally very small in affective computing applications.

Two main strategies are applied for combining classifiers: fusion and selection. This study investigates the classifier fusion approach. In classifier fusion, each classifier is provided with complete information about the feature space. Combiners such as the average and majority vote are then applied for fusion. Combined classifiers may not necessarily always out-perform one single classifier, but the accuracy will be on an average better than all the base classifiers (Utthara et al., 2010, Kuncheva, 2004).

Combining classifier can be suitable in emotion studies for affect detection or classification where features contribute from multiple modalities (reason 1) and individuals (reason 2) in varying environmental setups. Omar AlZoubi et al. (2011) proposed a classifier ensemble approach using a Winnow algorithm to address the problem of day-variation in physiological signals for affect detection. However, the study used only one type of classifier (four SVM classifiers) for the ensemble. Combined classifiers have been considered in some of our previous studies related to affect detection from multimodal features (Hussain et al., 2012, Hussain et al., 2011b, Hussain et al., 2011a), however the improvements over the base classifiers have not been justified.

In this study we have applied vote classifiers to detect affects, in this context detecting how positive or negative their valence (e.g. happy vs. unhappy) and its intensity (aka arousal or activation) using features from multichannel physiology and facial video. Three types of base classifiers (function, lazy, decision trees) are considered and results are evaluated for the individual base classifiers and the vote classifiers. The study provides empirical justification of using the vote classifiers for affect detection using multimodal features collected from a variety of subjects, during controlled stimulus presentation. The vote classifiers are also briefly

evaluated for affect detection using a separate dataset (Hussain et al., 2012), collected during naturalistic interactions with an Intelligent Tutoring System (ITS).

Section two gives a brief description of the data collection procedure and section three gives the computations model. Section four gives the results with discussions followed by conclusion in section five.

2 Data Collection: Participants, Sensors, and Procedures

The data used for detecting affect (i.e. the arousal and valence dimensions) in this paper was collected in a study where participants viewed emotionally stimulating photos. The purpose of this experiment was to collect physiological signals and facial video in response to emotional stimulus (3-degrees of arousal and valence). Data was collected from 20 students (8 males and 12 females, age ranged from 18 to 30) from University of Sydney. Each session took approximately 15 to 20 minutes for preparation (consent forms, sensor setup and the explanation of experiment protocol) prior to the experiments. Physiological signals and facial video were recorded during the entire session. The participants' electrocardiogram (ECG), skin conductivity (SC), and respiration (Resp) were measured using a BIOPAC MP150 system with AcqKnowledge software. Video was recorded using Logitech Webcam Pro 9000. All videos were recorded in colour at 15 frames per seconds (fps) with pixel resolution of 640×480 pixels.

The experiment was conducted under systematic setup and considered images from the International Affective Picture System (IAPS) (Lang et al., 1997) as part of the emotion stimulation process. The IAPS collection contains set of colour photographs with normative ratings of emotion (valence, arousal, dominance) providing a set of emotional stimuli frequently used in experimental investigations of emotion and attention. The normative ratings in the IAPS collection are the result of many studies with large number of subjects.

Each session lasted approximately 40 minutes where participants viewed the photos from the IAPS collection. A total of 90 images were presented; each image was presented for 10 seconds, followed by a 6 seconds pause showing a blank screen between the images. The experiment was interrupted by short breaks after presenting every 30 images. Images were categorized based on IAPS normative ratings so that the valence and arousal scores for the stimulus spanned a 3×3 space and presented based on the affective circumplex model (Russell, 1980).

3 Computational Model

The computational model for feature extraction, feature selection and classification were implemented in Matlab with the support of in-house and third party toolboxes.

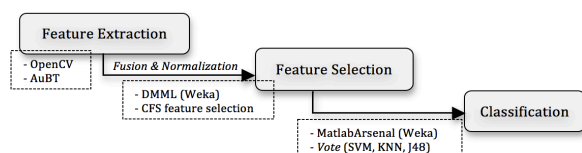


Figure 1: Overview of computational model

Figure 1 gives the overview of the computational model. The following subsections provide detailed description of the three main computational modules followed by the classifier training and testing procedure.

3.1 Feature Extraction, Normalization and Feature Fusion

A total of 287 features were extracted from the facial video and the physiological signals. Feature vectors were calculated from the time window corresponding to the duration of each stimulus presentation (10 seconds). The feature vectors were also labelled with the normative ratings (1-3 degrees of valence/arousal). The feature extraction and normalization process is explained briefly as followed.

Videos were analysed offline using MATLAB and Open Computer Vision library (OpenCV)¹. Two types of image-based features were explored: geometric and chromatic features. Five geometrical data (x and y coordinates, width, height and area) were derived which determined the position of the head in each frame. In addition, each frame was separated into red, green and blue colours in different conditions, due to movement or changing illumination sources. A total of 115 features were extracted from the videos (59 from geometric and 56 from chromatic).

Statistical features were extracted from the different physiological channels using the Augsburg Biosignal toolbox (AuBT) (Wagner et al., 2005) in Matlab. Some features were common for all signals (e.g. mean, median, and standard deviation, range, ratio, minimum, and maximum) whereas other features were related to the characteristics of the signals (e.g. heart rate variability, respiration pulse, frequency). A total of 172 features were extracted from the five physiological signals (84 from ECG, 21 from SC, and 67 for respiration).

All features were merged to achieve the fusion model (*fusion*) for further analysis. All physiological features were considered as the *physio* modality and both geometric and chromatic features were considered as the *face* modality. Hence, *fusion* contained all features of these two modalities. All features were normalized using *z-scores* before classification.

3.2 Feature Selection

The feature selection was implemented in Matlab using the DMML² wrapper for Weka (Hall et al., 2009). Feature selection techniques are used for discarding redundant, noisy features. This study investigates correlation based feature selection (CFS) as a way of choosing the best subset of features. The feature selection was performed separately for all individual modalities, and their fusion. The CFS technique evaluates the worth of a subset of features by considering the individual predictive ability of each feature along with the degree of redundancy between them (Hall, 1999). Equation (1) gives the merit of feature subset S consisting of k features.

¹ OpenCV: opencv.willowgarage.com/wiki/

² DMML: featureselection.asu.edu/software.php

$$Merit_{S_k} = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \quad (1)$$

Where, $\overline{r_{cf}}$ is the average value of all feature-classification correlations, and $\overline{r_{ff}}$ is the average value of all feature-feature correlations. The subset with the highest merit, as measured by Equation (1) found during the search, is used to reduce the dimensionality of both the original training data and the testing data. The CFS is defined by Equation (2). The $\overline{r_{cf_i}}$ and $\overline{r_{fif_j}}$ variables are referred to as correlations.

$$CFS = \frac{max}{S_k} \left[\frac{r_{cf_1} + r_{cf_2} + \dots + r_{cf_k}}{\sqrt{k + 2(r_{f_1f_2} + \dots + r_{f_1f_j} + \dots + r_{f_kf_1})}} \right] \quad (2)$$

3.3 Classification

The classification was performed in Matlab using MatlabArsenal³, a wrapper for the classifiers in Weka (Hall et al., 2009). Three types of base classifiers: lazy, function, and tree are considered.

Firstly, the three types of classifiers are evaluated: decision trees (J48), k-nearest neighbor (KNN), and support vector machine (SVM). In particular, SVM, KNN and decision trees are popular based on their compatibility and performance in many applications (Nguyen et al., 2005). These popular supervised learning algorithms that are simple to implement, span a variety of machine learning theories and techniques (e.g. function, lazy, tree), making them suitable in combined classifiers for addressing the diversity of features and subject variability. The *CVParameterSelection*, a meta-classifier in Weka that performs parameter selection by cross-validation was used to evaluate and determine parameter values for the classifiers with our dataset. The *K* value of one was selected for KNN classification. The exponent value of 1.0 (linear kernel), complexity factor of 1.0 was set for SVM. The C4.5 decision tree was used with confidence factor set to 0.25, and considering the subtree operation when pruning.

Secondly, two types of vote classifiers (as followed) are evaluated for combining classification results from the base classifiers to achieve the final classification decisions.

Average Vote Classifier (AVC): This vote classifier is a meta-classifier that combines the probability distribution of base classifier using the average probability rule. This is categorized as combining probabilistic (soft) outputs (Utthara et al., 2010, Kuncheva, 2004). This Vote classifier determines the class probability distribution computing the mean probability distribution of the base *N* arbitrary classifiers as followed (Seewald, 2003):

$$\overline{pred} = \sum_{i=1}^N \frac{\overline{P_i}}{N} \quad (3)$$

Where, $\overline{P_i}$ refers to the probability given by classifier *i*. The Voting prediction for *j* classes are mapped using $\overline{P'_i}$

instead of $\overline{P_i}$ in Equation (3). $\overline{P'_i}$ is the vector of *p*., for all *j*, where

$$P'_{i,j} = \begin{cases} 1 & \text{if } j = \text{argmax}_j(P_{i,j}), \text{ for given } i \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Weighted Majority Vote (WMV): In this vote classifier more competent base classifiers are given greater power to make the final decision based on the weighted majority vote algorithm (a meta-learning algorithm). This classifier is categorized as combining class labels (crisp outputs) (Utthara et al., 2010, Kuncheva, 2004). The class labels are available from the classifier outputs. The decision by classifier *i* (from *N* arbitrary classifiers) for class *j* is defined as $d_{i,j}$. If the classifier chooses class ω_j then $d_{i,j}=1$, and 0 otherwise. The classifiers whose decisions are combined through weighted majority voting will choose class ω_k if

$$\sum_{i=1}^N b_i d_{i,k} = \max_j \sum_{i=1}^N b_i d_{i,j} \quad (5)$$

Where, b_i is the weight coefficient for classifier *i*.

3.4 Training, Testing and Evaluation

All datasets were initially shuffled and randomized. Then the training and testing was performed separately with 10-fold cross validation. In 10-fold (*k-fold*) cross-validation, each dataset or sample was randomly partitioned into 10 subsamples. Of the 10 subsamples, a single subsample was retained as the validation data for testing the model, and the remaining 9 (*k-1*) subsamples were used as training data. The cross-validation process was then repeated 10 times (the folds), with each of the 10 subsamples used exactly once as the validation data. The 10 results from the folds were then averaged to produce a single estimation.

The *ZeroR* classifier is used for determining the baseline accuracy. The accuracy score is used for reporting the overall classification performance and precision score is used for reporting performance of individual classes.

4 Results and Discussion

In this section we provide results for detecting 3-degrees of valence (negative, neutral, positive) and arousal (low, medium, high) from *physio*, *face* and *fusion* using the vote classifiers (AVC and WMV) and the base classifiers (J48, KNN, SVM). Figures 2 and 3 give the average classification accuracy and the standard deviation (error bars) over all subjects⁴. The baseline classification accuracy is 33% for both valence and arousal.

Firstly, evaluating the overall performance of detecting degrees of valence and arousal from individual modalities (*physio* and *face*) and *fusion* shows that in almost all cases (except J48 and KNN in arousal) *fusion* has higher accuracy and lower standard deviation. Secondly, evaluating the classifiers show that both AVC and WMV in general exhibits similar (compared to individual modalities) or higher (compared to *fusion*) accuracy compared to the base classifiers.

³MatlabArsenal: cs.siu.edu/~qcheng/featureselection/index.html

⁴ Results for 19 subjects due to SC sensor failure in one subject.

Among the base classifiers, KNN exhibits the highest accuracy for *physio* (50%), *face* (60%) and *fusion* (62%) in valence (Figure 2). J48 exhibits the highest accuracy for both *physio* (49%) and *fusion* (56%) with slightly higher accuracy with KNN for *face* (57%) in arousal (Figure 3). SVM shows comparatively low accuracy in *face* for both valence and arousal. For this dataset, the vote classifiers are unable to improve the accuracy of the individual modalities over the base classifiers, except in *physio* for valence (showing 2% and 1% improvement in AVC and WMV respectively). The *fusion* exhibits 2% improvement (both AVC and WMV) in valence and 4% improvement (only AVC) in arousal compared to the accuracy of the best base classifiers. However, the improvements by AVC were statistically significant⁵ only over SVM for *face* in both valence and arousal. The improvement by WMV was also significant over SVM for *face* but only for arousal.

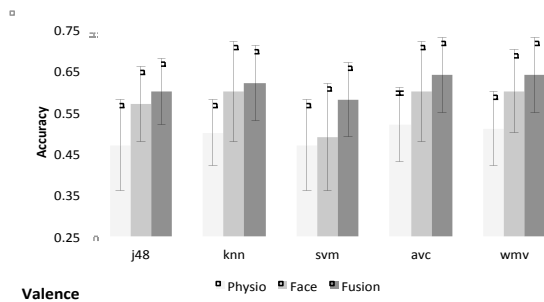


Figure 2: Accuracy (Mean, SD) of classifying 3-degrees of valence from *physio*, *face* and *fusion* using vote classifiers (AVC, WMV) and their base classifiers

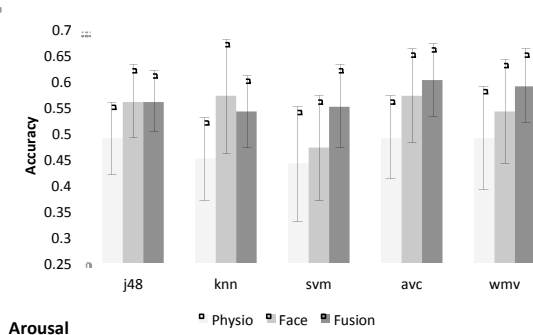


Figure 3: Accuracy (Mean, SD) of classifying 3-degrees of arousal from *physio*, *face* and *fusion* using vote classifiers (AVC, WMV) and their base classifiers

For this dataset, the base classifiers performed better for certain subjects and the vote classifiers for others. This reflects that for some subjects, where the performance of the base classifiers were poor, the vote classifiers achieved improvement. Figures 4 and 5 give the proportion of subjects representing the classifiers that performed with highest accuracy for valence and arousal respectively. The vote classifiers (specially AVC) performed better in most subjects for *fusion* compared to the individual modalities in both valence and arousal. This reflects that the vote classifiers perform best using features from multiple modalities. For *face*, KNN performed better for most subjects in valence and J48 in arousal. SVM in general

was less useful in most subjects except for *physio* in arousal. KNN was also less useful for *physio* and *fusion* in arousal.

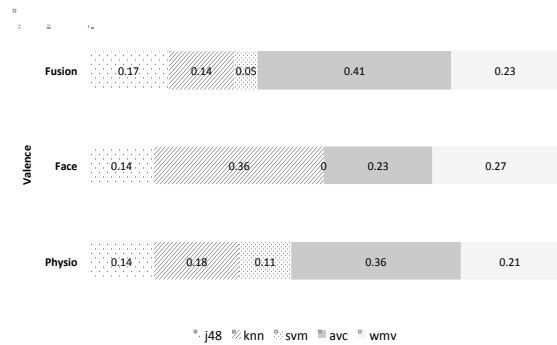


Figure 4: Proportion of subjects representing classifiers that performed with highest accuracy (val.)

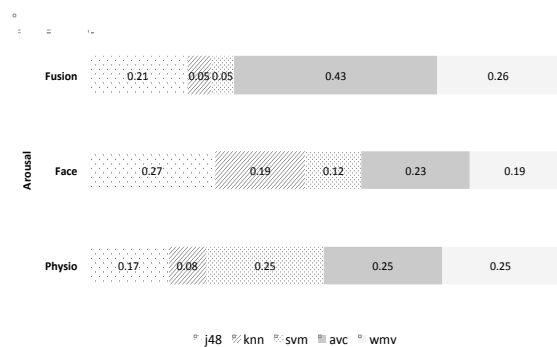


Figure 5: Proportion of subjects representing classifiers that performed with highest accuracy (ar.)

The classification was performed using balanced class distribution; therefore the precision score is used to report the classification accuracies of the individual classes, in this case individual degrees of valence and arousal. Table 1 gives the precision scores (mean and standard deviation) for classifying the individual degrees of valence and arousal from *fusion*. According to table 1, the vote classifiers exhibit higher precision compared to the base classifiers where both AVC and WMV show similar performance. For this dataset, AVC is slightly better at detecting *positive* valence and *medium* arousal whereas; WMV is best at detecting *neutral*, *negative* valence and *high* arousal. This reflects that vote classifiers have improved the accuracy of the individual affective states compared to the base classifiers.

	Valence			Arousal		
	Pos.	Neu.	Neg.	High	Med	Low
j48	0.62 (.13)	0.53 (.10)	0.64 (.12)	0.62 (.09)	0.47 (.14)	0.58 (.12)
knn	0.66 (.13)	0.53 (.14)	0.66 (.10)	0.57 (.10)	0.45 (.14)	0.61 (.12)
svm	0.66 (.17)	0.46 (.15)	0.61 (.19)	0.60 (.11)	0.39 (.18)	0.65 (.15)
avc	0.72 (.14)	0.53 (.13)	0.67 (.13)	0.64 (.10)	0.50 (.16)	0.66 (.11)
wmv	0.70 (.16)	0.55 (.13)	0.69 (.13)	0.65 (.09)	0.46 (.15)	0.66 (.14)

Table 1: Precision scores (Mean, SD) for detecting individual degrees of valence and arousal from *fusion*

⁵ One-way ANOVA and post-hoc test with *bonferroni*

The evaluation of these classifiers with the same computational model can also be presented using another dataset, which consists of similar features (physiological and facial video), collected from participants during naturalistic interactions with an ITS. Hussain et al. (2012) collected this dataset and reported the accuracy of detecting degrees of valence and arousal with AVC from physiological and facial features (see paper for more details about the experiment). Participants had self-reported their affect (3-degrees of valence and arousal) judgment which were synchronized with the physiological and facial video features and used as labels for classification. However, the study by Hussain et al. (2012) did not address detection accuracies of the base classifiers, thus did not quantify if AVC achieved any improvements.

The classifiers selected (base classifiers, AVC, and WMV) in our study in this paper can be evaluated with this dataset that represent affects self-reported from naturalistic interactions compared to normative ratings from a controlled stimulus presentation. Following the study by Hussain et al. (2012), in Figures 6 and 7, we present the overall classification accuracies for detecting degrees of valence and arousal respectively from the base classifiers and the vote classifiers.

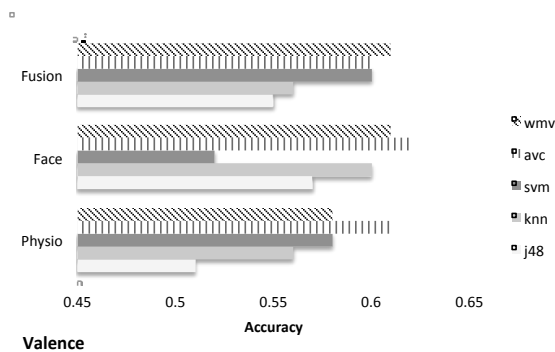


Figure 6: Detecting 3-degrees of valence (ITS dataset) from *physio*, *face* and *fusion* using vote classifiers (AVC, WMV) and their base classifiers

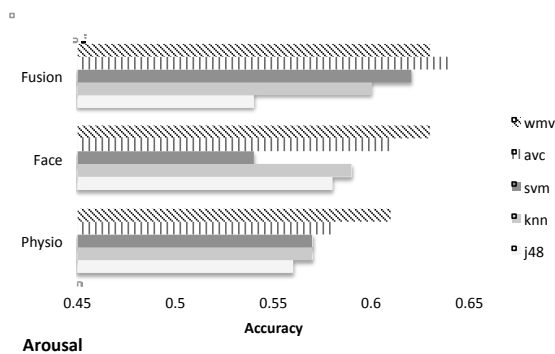


Figure 7: Detecting 3-degrees of arousal (ITS dataset) from *physio*, *face* and *fusion* using vote classifiers (AVC, WMV) and their base classifiers

The vote classifiers are able to improve the classification accuracy in valence and arousal for *face*, *physio*, and *fusion* compared to the base classifiers. In both valence and arousal, J48 was least useful for *physio* and *fusion*, whereas SVM was least useful for *face*.

Similar to the IAPS dataset, KNN proved to be most useful using this dataset for *face* in both valence and arousal. Comparing the vote classifiers, WMV exhibits 1% improvement over AVC for *fusion* in valence and vice-versa in arousal. AVC shows 1% and 3% improvements for *face* and *physio* respectively over WMV in valence. However, WMV shows 2% and 3% improvements for *face* and *physio* respectively over AVC in arousal. The highest accuracy for detecting the degrees of valence is from face with 62% accuracy using AVC (similar trend as in (Hussain et al., 2012)). However, *fusion* has the highest accuracy for detecting the degrees of arousal also using AVC with 64% accuracy.

5 Conclusion

In this study we have evaluated combined classifiers and compared their performances with the base classifiers for detecting degrees of valence and arousal from multimodal features. The vote classifiers considered in this study have showed improvement over the base classifiers (J48, KNN, SVM) using our dataset, especially by fusing the multimodal features. The classifiers that are more important for certain modality have been identified, for example KNN showed to be more useful and SMV least useful for the face modality in both valence and arousal. Even though the improvements of the vote classifiers are not extremely higher than the base classifiers, they are still useful for multimodal features and subject variability in behavioural studies.

As for future work, more base classifiers can be explored to replace less useful ones (for modalities and individuals) to be used for combined classifiers. The classifier selection methods (Kuncheva, 2002) can be applied on these datasets, where every classifier can be an expert in a specific domain (modality) of the feature space for the combined classifier to improve the detection accuracy of affects.

6 References

- Alzoubi, O., Hussain, M. S., D'mello, S. & Calvo, R. Affective modeling from multichannel physiology: analysis of day differences. International Conference on Affective Computing and Intelligent Interaction (ACII2011), 2011 Memphis, USA. Springer LNCS, 4-13.
- Calvo, R. A. & D'mello, S. 2010. Affect detection: An interdisciplinary review of models, methods, and their applications. *IEEE Transactions on Affective Computing*, 1, 18-37.
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. & Witten, I. H. 2009. The WEKA Data Mining Software: An Update. *ACM SIGKDD Explorations Newsletter*, 11, 10-18.
- Hall, M. A. 1999. *Correlation-based feature selection for machine learning*. The University of Waikato.
- Hussain, M. S., Alzoubi, O., Calvo, R. A. & D'mello, S. Affect detection from multichannel physiology during learning sessions with AutoTutor. The 15th International Conference on Artificial Intelligence in Education (AIED), 28 June - 02 July 2011a Auckland, New Zealand. Springer LNAI, 131-138.

- Hussain, M. S., Calvo, R. A. & Aghaei Pour, P. Hybrid fusion approach for detecting affects from multichannel physiology. International Conference on Affective Computing and Intelligent Interaction (ACII2011), October 2011b Memphis, Tennessee, USA. Springer LNCS, 568-577.
- Hussain, M. S., Hamed, M. & A., C. R. Categorical vs. dimensional representations in multimodal affect detecting during learning. 11th International Conference on Intelligent Tutoring Systems, 2012 Chania, Greece. Springer LNCS, 78-83.
- Kuncheva, L. I. 2002. Switching between Selection and Fusion in Combining Classifiers: An Experiment. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 32, 146-156.
- Kuncheva, L. I. 2004. *Combining pattern classifiers: Methods and algorithms*, Wiley-Interscience.
- Lang, P. J., Bradley, M. M. & Cuthbert, B. N. 1997. International affective picture system (IAPS): Technical manual and affective ratings. *Gainesville, FL: The Center for Research in Psychophysiology, University of Florida*.
- Nguyen, T., Li, M., Bass, I. & Sethi, I. K. Investigation of Combining SVM and Decision Tree for Emotion Classification. Seventh IEEE International Symposium on Multimedia, 2005 Irvine, California, USA. 540-544.
- Russell, J. A. 1980. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39, 1161-1178.
- Sebe, N., Cohen, I., Gevers, T. & Huang, T. S. 2005. Multimodal Approaches for Emotion Recognition: A Survey. *Proc. SPIE*, 5670, 56-67.
- Seewald, A. K. Towards a theoretical framework for ensemble classification. Proceedings of the 18th Int. Joint Conference on Artificial Intelligence (IJCAI-03), 2003. Morgan Kaufmann, 1443-1444.
- Utthara, M., Suranjana, S., Sukhendu, D. & Pinaki, C. 2010. A Survey of Decision Fusion and Feature Fusion Strategies for Pattern Classification. *IETE Technical Review*, 27, 293-307.
- Wagner, J., Kim, J. & Andre, E. From physiological signals to emotions: Implementing and comparing selected methods for feature extraction and classification. IEEE International Conference on Multimedia and Expo 2005, 6 July 2005 Amsterdam, The Netherlands. 940-943.

Application of Tree-structured Data Mining for Analysis of Process Logs in XML format

Dang Bach Bui

Fedja Hadzic

Michael Hecker

Department of Computing
School of Electrical Engineering and Computer Science
Curtin University of Technology,
Email: dang.buibach@postgrad.curtin.edu.au, f.hadzic@curtin.edu.au,
Michael.Hecker@cbs.curtin.edu.au

Abstract

Process logs are increasingly being represented using XML based templates such as *MXML* and *XES*. Popular XML data mining techniques have had limited application to directly mine such data. The majority of work in the process mining field focuses on process discovery and conformance checking tasks often utilizing visualization and simulation based techniques. In this paper, an approach is proposed within which a wider range of data mining methods can be directly applied on tree-structured process log data. Clustering, classification and frequent pattern mining are used as a case in point and experiments are performed on publicly available real-world and synthetic data. The results indicate the great potential of the proposed approach in adding to the available set of methods for process log analysis. It presents an alternative where process model discovery is not the pre-requisite and a variety of methods can be directly applied.

Keywords: process/event log analysis, clustering, frequent subtree mining, classification, XML/MXML/XES mining

1 Introduction and Related Works

A business process is a set of related activities following some logical order whose objective is to create a complete product or service for a customer or a market (Aguilar-Savén 2004). In process-aware information systems, when a business process is executed it leaves traces in an event log (also called process log) of the system (van der Aalst 2011). Cook et al. (Cook & Wolf 1995) was the first to introduce the concept of process mining in the software engineering domain, while Agrawal (Agrawal et al. 1998) generalized the concept to workflow management systems. The purpose is to give valuable information about the processes captured by business information systems. It significantly increases productivity and saves on cost by providing an insight view on business process by different simulation, modeling, analysis and data mining techniques (van der Aalst 2011). The process mining techniques have been applied in many domains e.g. software development, health care, public administration, etc. (van der Aalst 2011)

The aim of process discovery task is to find a model that best describes the workflow of a business process.

Many types of models are graphical-based and describe a variety of control flow constructs e.g. sequence, loop, choice, parallel, synchronization. Conformance checking aims to find any difference between the log and the model. Non-conformance may indicate either that the model does not reflect the reality, or deviating process instances which require corrective actions (van der Aalst 2011). Process enhancement analyzes other dimensions of the process log, e.g. actors and activities (work distribution), actors and actors (social network analysis), actor behaviors and decision mining (Rozinat & van der Aalst 2006), to optimize the business process. The majority of research has focused on the process model discovery task or on methods utilizing the process model as the base for analysis. For example, sequential event logs are often analyzed to discover the underlying process model (Günther & Van Der Aalst 2007, Weijters & van der Aalst 2003, van der Aalst et al. 2004, Maruster et al. 2002). Clustering of sequential event logs was used for the pre-processing step in process model discovery in (Bose & van der Aalst 2009, De Medeiros et al. 2008, Greco et al. 2006). Classification learning to predict the next possible event was studied in (Goedertier et al. 2008). The problem of mining frequent patterns of workflow schema executions was introduced in (Greco et al. 2005). The authors presented specialized graph mining algorithms to deal with structural constraints imposed by the workflow schemas and their instances. Please refer to (van der Aalst 2011, Tiwari et al. 2008) for a complete overview of process mining algorithms and techniques.

There is a recent momentum in representing event logs in XML format. The first XML standard created for event log is *MXML* (Günther & van der Aalst 2006) and more recently, *XES* standard was proposed in (Verbeek 2011). Other attempts in using XML to store event logs are presented in (Kim 2006) and (Gonçalves et al. 2002). Semi-structured documents such as XML are known for their ability to represent the contextual information among different data items in a domain specific way. Due to their hierarchical nature XML documents are commonly represented as rooted ordered labeled trees (Hadzic et al. 2011b, Zaki 2005). While sequence mining techniques have been applied in the process mining field, to our best knowledge, no tree-structured data mining techniques have been specifically explored for mining of *MXML*/*XES* event logs. For example, the frequent subtree mining methods are the basis for discovering interesting associations among tree-structured data objects in XML data, but their utilization in the process mining field is still to be explored. Same holds for other methods that take structural aspects into account during tasks such as XML clustering (Kutty et

al. 2011, Hadzic et al. 2011a) and classification (Kim et al. 2010). Note that a synthetic process log dataset was used in (Hadzic et al. 2011a), but the main purpose was to compare the time performance and quality of clustering solution between algorithms. The work in (Greco et al. 2005) demonstrated the benefit of applying frequent pattern mining that incorporates workflow schema structure during the analysis of workflow execution. Similarly, the process mining field would benefit by the application of tree-structured data mining techniques, as they not only preserve the order of events but also their context and structural organization within the workflow. In the case of web logs it has already been shown in (Hadzic et al. 2011b, Zaki 2005) that a subtree-based pattern is more informative than an itemset or a sequential pattern as it captures the structural properties as well as the navigational behavior over the web site structure. Hence in the case of event/process logs similar reasoning would apply, and a subtree pattern would capture the workflow execution pattern over the overall structure of the business process at hand. The subtree patterns also preserve the context of the events and event attributes within a trace.

The process log is often characterized by repetition of events within a trace and the underlying tree-structures of an XES document can grow quite large and complex which can pose a problem to the performance of tree-structured data analysis. To alleviate the complexity associated with mining complex structures and to enable a wider range of data analysis/mining techniques to be directly applied on tree-structured data, a structure-preserving flat data format of tree-structured data has been recently proposed in (Hadzic et al. 2011a, Hadzic 2011). An interesting implication of the method is that the exact positions of nodes/attributes are taken into account during the knowledge discovery process. This property can be useful in process mining as events/actions are distinguished based on their context or exact occurrence within a trace of events. Using this technique as a basis, a general approach is proposed capable of encompassing a broad range of business process aspects during the analysis phase. Decision tree learning, frequent pattern mining and clustering methods are used as case in point and applied to publicly available synthetic and real world data. The results indicate the capability of the approach in discovering interesting descriptive and discriminating characteristics of workflows. A variety of data mining techniques can be utilized within the approach, and applied directly to MXML/XES data to satisfy different application needs. This work extends the available pool of process mining techniques and does not require business process models as the basis for discovery.

The paper is organized as follow. An illustrative scenario that motivates the direction of this work is presented in Section 2. Section 3 describes the proposed approach components of which are tested on real world and synthetic data in Section 4. Section 5 concludes the paper and discusses our future work.

2 Motivation Scenario

In the data mining field an XML document is modeled as a rooted ordered labeled tree, which can be denoted as $T = (v_0, V, L, E)$, where (1) $v_0 \in V$ is the root vertex; (2) V is the set of vertices or nodes; (3) L is a labelling function that assigns a label $L(v)$ to every vertex $v \in V$; (4) $E = (v_1, v_2) \mid v_1, v_2 \in V \text{ AND } v_1 \neq v_2$ is the set of edges in the tree, and (4) for each internal node the children are ordered from left to right.

Frequent subtree mining (Hadzic et al. 2011b, Zaki 2005) is important to enable the discovery of association among tree-structured data object, and several algorithms have been developed that mine different subtree types (Hadzic et al. 2011b, Zaki 2005, Tan et al. 2008, Hadzic et al. 2008, Chi et al. 2005, 2004). A closed subtree is a subtree for which none of its proper supertrees has the same support, while for a maximal subtrees, no supertrees exist that are frequent. For a detailed overview of the frequent subtree mining field, please refer to (Hadzic et al. 2011b, Chi et al. 2004).

Fig. 1 shows a tree representation of a simplified example extracted from Dutch hospital process log (Bose & Aalst 2012) originally stored in a XES file. No matter what representation of the data is, process mining algorithms such as alpha (van der Aalst et al. 2004), heuristic miners (Weijters & van der Aalst 2003), fuzzy miner (Günther & Van Der Aalst 2007), etc. understand them as a set of sequences of events, and from that discover the generative process model. An example of process model discovered by heuristic algorithm is shown in Fig. 2a. It is noticeable that the process model discovered does not include any information about the attribute values of each event e.g. department information of where the activities were administered. In order to incorporate the contextual information of each event into the mining process, we consider the set of traces as a set of trees rather than as a set of sequences as is commonly done. Using the frequent closed subtree mining (Chi et al. 2005) on the tree database gives us different subtree patterns, each of which occur at different traces. An example of closed subtrees at support = 2 is given Fig. 2b. The subtree at the top occurs in trace (1) and (3) of Fig. 1. These subtree patterns preserve the order of events (according to pre-order traversal of the subtree) and thus represent the frequent path of executions. The frequent paths of execution (*consult* \Rightarrow *administration*) shown in Fig. 2b do not contain the event *phone consult*, *blood test* and *cytologic* because the minimum support value is set to 2; if we lower that parameter longer paths of executions can be detected. The benefit of using subtree mining is that the activities and their contextual information are preserved in the pattern e.g. the path of execution information is enriched with their administering departments e.g. *Radio therapy*, *Obstetrics Lab*.

The order of events within a frequent path of execution is consistent among the matched instances in the database. However, the positional information of those individual events is not indicated in Fig. 2b and in reality there could be other events that occur in between the events belonging to the frequent path execution. The latter case can be observed from our example that the subtree pattern in the lower part of Fig. 2b does not match with trace (4) of Fig. 1 i.e. there is an extra *phone consult* event between the *consult* and *administration* events in the process log. In reality, this result can make us believe that under no circumstances an additional *phone consult* happens after the first *consult* which could cause misjudgments of the process. In general, the position of the captured patterns could be an important lead to many discoveries. Furthermore, process logs in XML format can be quite complex with many nodes in the underlying tree repeating throughout the traces. This can cause combinatorial problems and hinder the analysis due to the large volume of irrelevant patterns captured through frequent subtree. These characteristics of the data and the desired analysis at lower level of detail motivate us to explore the technique recently proposed in (Hadzic 2011) and utilized in (Hadzic et al. 2011a) for clustering, which forms the basis of the

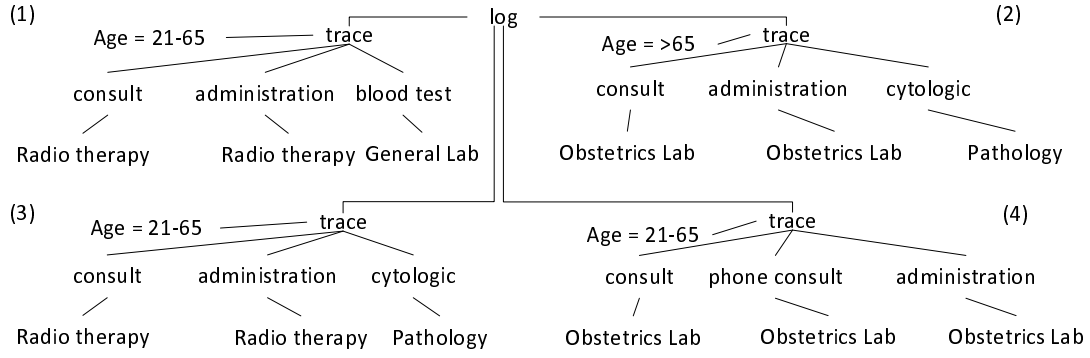


Figure 1: Traces and their elements presented as hierarchical structure

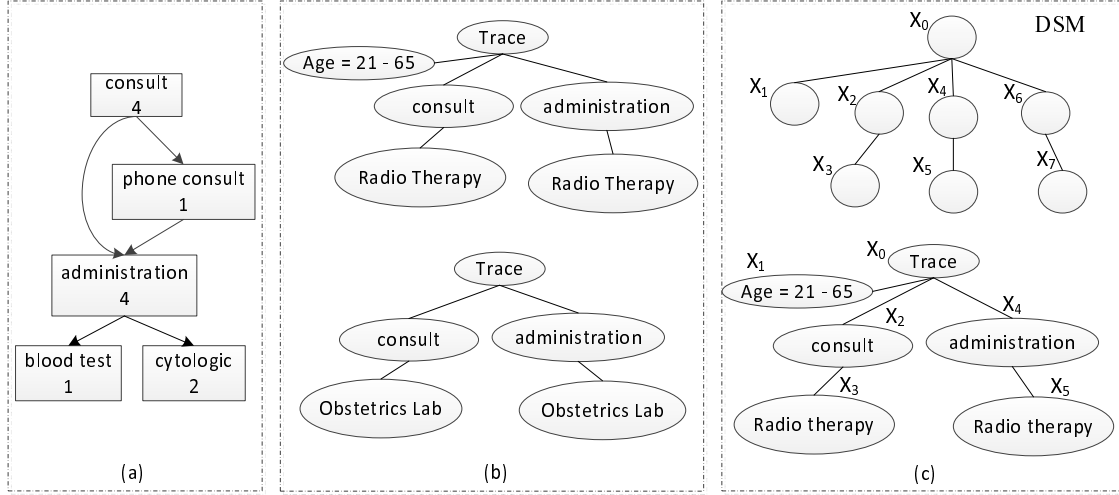


Figure 2: Patterns learned from (a) Heuristic Miner (b) Frequent subtree mining (c) DSM based frequent subtree mining

proposed approach described in the next section.

3 Proposed approach

The proposed approach shown in Fig. 3 consists of four main phases, pre-processing, database structure model (*DSM*) extraction and tree to flat conversion (Hadzic 2011), knowledge discovery and interpretation. Depending on the specific purpose of the process mining task, different pre-processing techniques could be used e.g. grouping/removal, discretization, filtering etc. If the MXML/XES data is not ready, extract transform and load methods can be used (van der Aalst 2011). This semi-structured file is modeled as a set of rooted ordered labeled trees and represented in a pre-order string encoding (Zaki 2005). A pre-order string encoding lists the node labels in the sequence of the pre-order traversal of a tree, and uses a special symbol (e.g. -1), when backtracking up the tree. For example, the pre-order string encoding of the trace (1) in Fig. 1 is 'trace', 'Age = 21-65', -1 , 'consult', 'Radio therapy', -1 , -1 , 'administration', 'Radio therapy', -1 , -1 , 'blood test', 'General Lab', -1 , -1 .

The technique (Hadzic 2011) utilized in this approach converts tree-structured data into a flat data structure format (henceforth referred as table) while preserving both structural and attribute-value information. The approach starts by first extracting the *DSM* (Hadzic et al. 2011a, Hadzic 2011) of which each tree instance is a valid subtree. This *DSM* contains the most general structure where every instance

from the tree database can be matched to. The *DSM* tree is shown on top of Fig 2c. The pre-order string encoding of *DSM* will become the first row of the table with nodes X_i (i corresponds to the pre-order position of the node in the tree) and backtracks b_j (j corresponds to the backtrack number) are used as the attribute names. For each record, when a label is encountered, it is placed to the matching column under the matching node X_i in the *DSM* structure. When a backtrack (-1) is encountered, a valued 1 is placed to the matching backtrack b_j . Remaining entries are assigned a value of 0 (non-existence). The resulting table is called *DSM-Flat*. Table 1 shows the flat representation of the traces in Fig. 1. This conversion process enables the application of frequent pattern mining, clustering, classification and prediction techniques originally developed for vectorial data directly to tree structured process data (knowledge discovery phase in Fig. 3). The discovered knowledge patterns can be re-mapped to tree structure by using the *DSM*. For example, a frequent subtree mining task can be done by first converting Table 1 into itemset format as shown in Table 2 and then performing the frequent itemset mining. One of the frequent itemsets found at support=2 is $(X_0)'trace'$, $(X_1)'Age=21-65'$, $(X_2)'consult'$, $(X_3)'Radio therapy'$, $(X_4)'administration'$, $(X_5)'Radio therapy'$, which can be mapped into a tree as shown at the bottom of Fig. 2c using the method described in (Hadzic et al. 2011a). In comparison to the respective traditional subtree displayed at the top of Fig. 2b, one can see that using the *DSM* approach, the positional

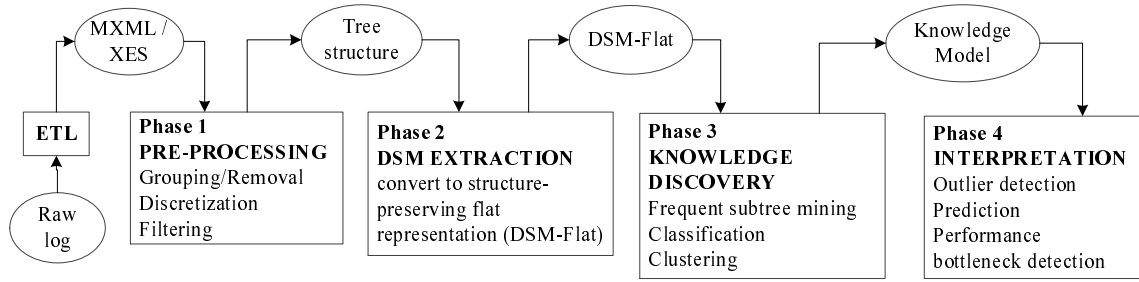


Figure 3: The proposed approach

X_0	X_1	b_0	X_2	X_3	b_1	b_2	X_4	X_5	b_3	b_4	X_6	X_7	b_5	b_6
trace	Age21-65	1	consult	Radio therapy	1	1	administration	Radio therapy	1	1	blood test	General Lab	1	1
trace	Age>65	1	consult	Obstetrics Lab	1	1	administration	Obstetrics Lab	1	1	cytologic	Pathology	1	1
trace	Age21-65	1	consult	Radio therapy	1	1	administration	Radio therapy	1	1	cytologic	Pathology	1	1
trace	Age21-65	1	consult	Obstetrics Lab	1	1	phone consult	Obstetrics Lab	1	1	administration	Obstetrics Lab	1	1

Table 1: Flat representation of tree database

information of each node is indicated. In process mining this interprets to an exact occurrence of an event (or any other aspect of the process) within a trace. This characteristic of distinguishing subtree based upon their exact occurrences would cause the DSM method not to detect the subtree displayed at the bottom of Fig. 2b at support = 2. This is because the right hand side of the subtree (i.e. node *administration* and *Obstetrics Lab*), occurs at different position within trace (2) and (4) of Fig. 1. Hence, the DSM approach would not consider groups of events similar if additional/different events occur within the group (e.g. *phone consult* in trace (4) of Fig. 1). This is important as the analyst can be certain that the DSM based subtree pattern reflects exact similarity of groups of events across traces, where no additional or different events occur in between. Moreover, using the DSM approach, structural complexity associated with mining of complex process logs is avoided by flat representation while structural characteristics of the data are still preserved. For a detailed description of the DSM approach please refer to (Hadzic et al. 2011a, Hadzic 2011).

Besides indicating the structural characteristics of the knowledge patterns discovered, in the interpretation phase of the approach, the patterns will be evaluated for their specific use in a given application. For example in outlier/exception detection, the low occurring frequent subtree patterns can indicate characteristics of outlying or exceptional cases. The difference to the more frequently occurring patterns reflecting the norm will be investigated. The instances characterized by such rare patterns may correspond to outliers. In context of clustering, clusters covering only small percentage of instances are suspect of being outliers. The cluster characteristics will be compared to see what the difference is between the outliers and clusters with many instances. The classification methods can be useful for outlier prediction purposes. Hence, once the outlying instances are detected, the instances themselves will be labeled as outlying and others as norm and classification techniques are run to discover a classification model. This knowledge model can then be used to predict the outlying behavior, when a set of preconditions during business process execution path become true. Generally speaking, the classification methods are useful when the process log can be labeled with respect to a particular business aspect (e.g. duration, performance bottleneck, known cases of exceptions/fraud). This will be demonstrated in the next section when applied on

data where labels can be logically assigned.

In the cases where no label can be logically assigned to the process log, we adopt a different approach to enable one to discover different variations in process executions, and learn about descriptive characteristic of each variation, as well as discriminating characteristics with respect to others. This approach is illustrated as a whole in Fig. 4. It starts by performing clustering on the process logs to group process instances which have similar process execution paths. To discover the descriptive characteristics of each cluster we apply frequent pattern mining to detect the common subtree pattern(s) (paths of execution) among instances (step 2a). To detect discriminating characteristics among the clusters, process instances are assigned a virtual cluster label so that classification models can be discovered (step 2b). The results of this process when applied on process log from a Dutch Hospital (Bose & Aalst 2012) are discussed in the next section.

4 Approach Discussion

The approach proposed in the previous section can be better justified when it is compared with other approaches or applied in real environment. However, there are no comparable methods at this moment and we are waiting to apply this approach in practice in order to get feedbacks from domain experts. At this phase, the approach design is motivated as follows.

- Phase 1 (Pre-processing): the process log is originally in semi-structured format. This data should be converted into a tree-structured form in order to enable the direct mining of the contextual and chronological information of the events. This maximizes the potential of the knowledge discovered.
- Phase (DSM extraction): recent tree mining methods are not able to mine the tree database without losing all or part of the position information of the subtree patterns. Furthermore, the complexity of the frequent subtree pattern searching is high. The DSM extraction phase transform the tree database into a flat representation that enables the direct application of traditional data mining techniques.
- Phase 3 (Knowledge discovery): depending the goals one wants to achieve, different data mining techniques can be utilized at this phase. For

	Itemsets
1	(X ₀)trace, (X ₁)Age21-65, (X ₂)consult, (X ₃)Radio therapy, (X ₄)administration, (X ₅)Radio therapy, (X ₆)blood test, (X ₇)General Lab
2	(X ₀)trace, (X ₁)Age>65, (X ₂)consult, (X ₃)Obstetrics Lab, (X ₄)administration, (X ₅)Obstetrics Lab, (X ₆)cytologic, (X ₇)Pathology
3	(X ₀)trace, (X ₁)Age21-65, (X ₂)consult, (X ₃)Radio therapy, (X ₄)administration, (X ₅)Radio therapy, (X ₆)cytologic, (X ₇)Pathology
4	(X ₀)trace, (X ₁)Age21-65, (X ₂)consult, (X ₃)Obstetrics Lab, (X ₄)phone consult, (X ₅)Obstetrics Lab, (X ₆)administration, (X ₇)Obstetrics Lab

Table 2: Itemset format

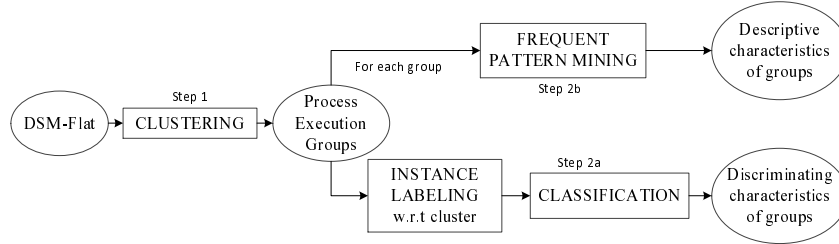


Figure 4: Proposed method for unsupervised process log analysis

example, the process instances can be labeled according to different business requirements and then be used to train classifiers for prediction purposes. One common application is that in case the process analyst is not familiar with the domain of the process log, clustering methods may help identify groups of similar process instances. Frequent pattern mining can then be applied to identify the descriptive characteristics of each group. Classification techniques are able to identify the discriminative characteristics among the groups.

- Phase 4 (Interpretation): in this phase, the results obtained from previous phases should be analyzed and interpreted in a way is understandable and actionable to the domain experts.

5 Experiments and Discussion

The proposed approach is tested on a real life hospital dataset and two synthetic datasets regarding insurance claim and telephone repair. For all experiments we follow the generally proposed four-phase approach Fig. 3). Note that in Phase 2, we do not separate the XML elements, attributes and their values into separate nodes. Therefore in our tree representation of process logs a node label can be a representative of both the element and the element value, while the hierarchical properties of the document are adhered to. The tree mining and frequent item set mining algorithms were run on a Linux machine, Intel Xeon E5345 at 2.33 GHz, 8 GB RAM and 4MB Cache Open SUSE 10.2 64bit. The classification, clustering and DSM extraction/DSM-flat conversion tasks were executed on Windows Server 2008 64-bit machine with 128GB of RAM, quad socket quad core Xeon E7330 (2.4 GHz).

5.1 Hospital dataset

The process log used in this experiment is taken from a Dutch Hospital. Each trace in the input file has a set of attributes e.g. *diagnosis*, *diagnosis code*, *treatment*, *treatment code*, *start time*, *end time*, *age* etc. Each event in a trace includes a set of nine attributes such as: *org:group* (the department in which the activity occurred), *concept:name* (the name of the activity), *time:timestamp* etc. The dataset describes a large variety of processes characterizing different treatments/diagnosis, which themselves are overlapping. These properties of the dataset were described in (Bose & Aalst 2012), with a clear indication that

k	Correlation	Euclidean	Jaccard
3	0.517(0.153)	0.388(0.001)	0.306(0)
4	0.574(0.197)	0.552(0.001)	0.287(0.001)
5	0.586(0.209)	0.467(0.012)	0.452(0.001)
6	0.594(0.213)	0.573(0.01)	0.41(0.002)
7	0.61(0.223)	0.607(0.028)	0.471(0.006)
9	0.653(0.25)	0.637(0.04)	0.5(0.002)

Table 3: Average internal similarity and external similarity of different clustering solutions

one needs to focus on only one aspect/segment of the data, for example a single diagnosis or treatment code. Hence, in this experiment we focus on the set of processes where the describing attribute is a single diagnosis code or treatment code. A subset of data that contains traces of patients who are diagnosed with the code *M13* (most frequent diagnosis) is selected for analysis. For traces that have multiple treatment codes, the codes are concatenated into a single treatment code, while preserving the order of the concatenation across records with same treatment code combination. Other trace attributes such as *specialism code*, *treatment code combination ID*, etc. are removed as they are either unique for each trace or contained elsewhere in the trace. Event attributes *lifecyle:transition* and *number of executions* are removed as their values are the same across all logs. The structural properties of the subset of the dataset reflecting diagnosis code *M13* are as follows: |transactions|= 252; avg. encoding length = 934.5; max. tree size = 2796; avg. tree height = 2; avg. tree fan-out = 7.4; avg. tree size = 467.7; max tree height = 2; max. tree fan-out = 352. Unlike the approach described in (Bose & Aalst 2012) where the original dataset is preprocessed and examined according to one perspective e.g. time perspective, organizational perspective, urgent and non-urgent case perspective, our approach as described in Fig. 3 and Fig. 4 gears toward a more holistic solution which is more beneficial for users that are not familiar to the domain. In what follows we describe how our method is applied to the hospital dataset.

Discovering process execution groups: it is easier to perform the pattern analysis if the hospital dataset can be split into different homogeneous groups of instances. This is best done in unsupervised way as we do not have specific knowledge about the domain. The hospital data is clustered into different process execution groups. We utilized the DSM based clustering technique proposed in (Hadzic et al. 2011a). It converts the data to a structure-preserving flat representation and uses the CLUTO clustering

Cluster	Size	ISim	ESim
0	13	0.999	0.003
1	28	0.88	0.002
2	58	0.552	0.001
3	5	0.488	0
4	37	0.2	0.001
5	74	0.193	0

Table 4: Clustering result at $k=4$ and Euclidean distance measure is used

toolkit (Karypis 2003) to form clusters. The only difference is in the representation as in (Hadzic et al. 2011a) each XML entity (element, attribute and their values) is represented as an individual node, while in this work the attributes and/or values of an XML element are mapped to the same node as the element itself. Please note that in these experiments we do not provide comparisons of the method with other clustering methods for tree-structured data, as extensive comparison on data of varied complexity including a complex synthetic process log, is already provided in (Hadzic 2011). The number of clusters (k) is trial with different values and Euclidean, Jaccard and correlation distance measures are used. Table 3 shows the average internal similarity and external similarity values of all clustering solutions with the latter shown in parentheses. The clustering solution that has the smallest number of k and the largest gap between the average of internal and external similarity is selected for further analysis. The best result is achieved with parameter $k=4$ using Euclidean distance measure, which produced a clustering solution having six clusters. The detailed internal and external similarity measures (abbreviated as ISim and ESIm, respectively) of each cluster are presented in Table 4. It took 9.25s to run the whole process including DSM extraction, DSM-flat conversion and clustering. It can be seen from the table that the top two clusters outperform the remaining clusters in term of the difference between the internal similarity and external similarity.

Discovering discriminating characteristics of groups: groups of similar process instances are identified in the first step in the proposed method. It is useful for a process analyst if he/she can identify the key differences in characteristics of instances among different groups. This can be done by applying classification algorithm on each group; due to their high number of instances and cluster quality, the first three clusters are selected for the classification task. For each cluster, 70% of instances are reserved for the training set and the remainder for the test set. Because the number of instances for each class is different, we apply the oversampling method to balance the examples of each class. The *C4.5* tree induction algorithm (Quinlan 1993) and the *Rapidminer* software (Mierswa et al. 2006) with default parameter settings are used in all classification tasks of this paper. The resulting decision tree is shown in Fig. 5 and it has accuracy of 83.3%. The whole tree induction, applying model and performance evaluation was accomplished in under 1 second. The left most branch of the decision tree shows a rule that if $X_2 = \text{Age} = \text{retired}$, $X_{28} = \text{NO}$ and $X_{19} = \text{duration} = 0$ then the instance is classified as belonging to cluster 0. Due to the property of the DSM approach of representing tree-structured data, the position of an attribute X_n in a DSM tree can always be inferred. In the hospital dataset, each event has seven attributes, thus if X_4 stores the node *event* of the 1st event of a trace then the node *event* of the 3rd event (if it exists)

is located at X_{28} in the DSM tree. With this knowledge in mind, we can interpret the above rule as if the patient is > 65 years old and there are no more than 3 events (e.g. consultations) in the process (the whole treatment) and the duration of the 2nd event is less than 1 day, this patient treatment process belongs to cluster 0.

Discovering descriptive characteristics of groups: each group identified by the clustering algorithm in the first step of the proposed method contains process instances that shares similar characteristics. A frequent pattern mining algorithm can reveal the characteristics that are prevalent among its instances. This can be done either by a frequent closed subtree mining algorithm *CMTreeMiner* or a frequent closed itemset mining algorithm *LCM* (Uno et al. 2004) applied on the flat representation of the tree acquired using DSM method (Hadzic 2011) (abbreviated as LCM-DSM). Both frequent pattern mining minimum supports are set at 90% and the results are shown in Fig. 6 and Fig. 7, respectively. Note that since in our scenario the aim is to discover the patterns that reflect the characteristics of majority of process execution instances within a group, a rather large support threshold is used. For different application aims one could choose lower support thresholds, and this would typically result in more specific patterns characterising smaller subsets of process execution instances within a group.

The subtrees of Fig. 6 indicate some distinguishing characteristics of cluster 0 and 1. For example, cluster 0 is characterized by 2 events, each having producer code of *SGNA* where the 2nd event has attribute *name* = *administratief* (administration). On the other hand, cluster 1 is characterized by 3 events, the first one having attributes *name* = *vevolgconsult* (follow up consultation) and *activity code* = *411100*. Further, cluster 0 is characterized by *AgeGroup* = *retired*, and cluster 1 by *AgeGroup* = *working*. Note however, that from these subtrees one cannot be certain whether any other events occurred between detected common events among the instances and/or whether the additional events differed. On the other hand the subtrees detected using LCM-DSM in Fig. 7 confirms that no additional events occurred between the common events of cluster 0, and that they were the first executed events within the traces. In fact, the DSM based subtrees also indicate how many events in total occurred in the traces of the instances (90% in this case) of the cluster, i.e. 3 events for cluster 0 (X_4 , X_{12} and X_{20}) and 4 events for cluster 1 (X_4 , X_{12} , X_{20} and X_{28}). Furthermore the attribute *name* = *administratief* in the 2nd event of traditional subtree of cluster 0 in Fig. 6 did not occur in DSM-based subtree of cluster 0 in Fig. 7, which also indicates that this attribute was not frequent in the 2nd event but was frequent when its occurrence in the 2nd and 3rd event was counted together. We have confirmed this with the instances of cluster 0 and have found that the attribute *name* = *administratief* occurs 7 times in the 2nd event and 6 times in the 3rd event. Another difference is that the attributes *name* = *vevolgconsult* and *activity code* = *411100* did not occur in the DSM based subtree of cluster 1. The DSM subtree of cluster 1 indicates that there was another event (at X_4) before the event at node X_{12} (i.e. the 2nd event in the traces). We have inspected the instances of cluster 1 and have found that the association of *name* = *vevolgconsult* and *activity code* = *411100* occurs 18 times in 1st event and 7 times in 2nd event.

These differences indicate the benefit of DSM in cases when we would like to find the exact location of each repeated or outlying values, or know the exact

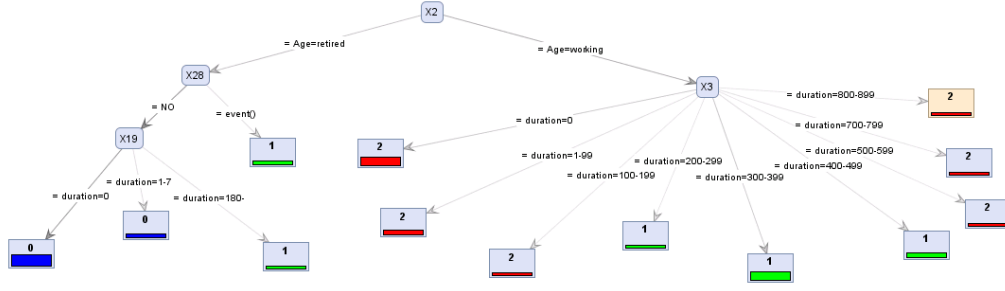


Figure 5: Decision tree learned from three clusters of process instances

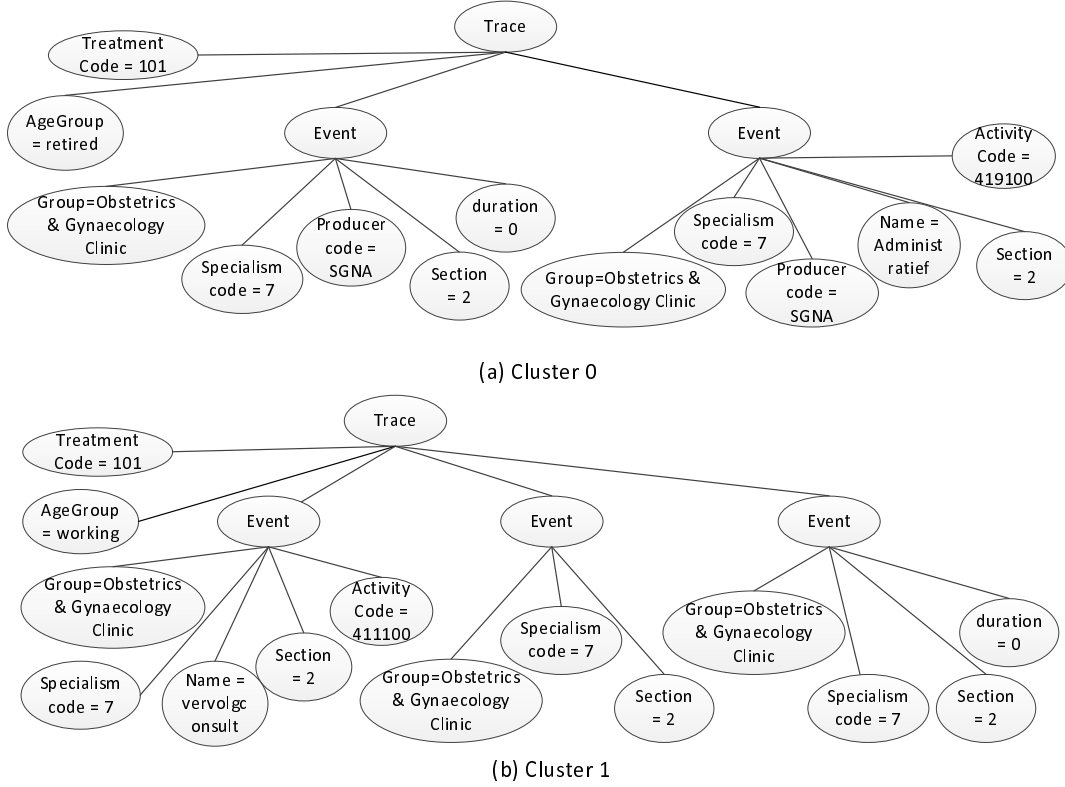


Figure 6: Frequent subtrees identified by CMTreeMiner

occurrence of an event and its characteristics within the trace as a whole. However, we do not claim that the traditional subtree mining would not be useful, but that the different approaches each have its own useful characteristics. For example the traditional subtrees reflect the occurrences of events and their characteristics no matter in which part of the trace they occurred, while the DSM based subtrees provide further detail of their exact location within a trace, and both could be used in a complementary way to obtain a more comprehensive analysis.

5.2 Insurance dataset

The synthetic insurance process log describes the handling of claims in an insurance company. The log contains 46138 events related to 3512 cases (claims). One typical process is that the customers file the claims, the center checks information, registers to the system, the claim is then quickly checked by a claim handler, after that it is fully examined; the officer advises the claimant and starts payment; finally the claim is closed. The purpose of this experiment is to

build a classification model to identify four possible outcomes of a claim such as *processed*, *rejected*, *insufficient information* or *not liable*. Each trace is first labeled from one of the four values as described above. The structural properties of the tree database are as follows $|\text{transactions}| = 3512$; avg. length of encoding = 114.1; max. tree size = 73; avg. height of trees = 2; avg. fan-out of trees = 4; avg. size of trees = 57.5; max height of trees = 2; max. fan-out of trees = 19. The decision tree model discovered from the balanced dataset is displayed in Fig. 8. The running time is 2 seconds and has accuracy of 100% evaluated using ten-fold cross-validation.

Fig. 8 shows that if activity *end* happens at the 7th event (at position X_{35}), the claim is identified as *not liable*; otherwise the claim would be *insufficient information*. Furthermore, at position X_{53} , we know that if an eleventh event is available the claim is *processed*, otherwise it is *rejected*. This indicates another useful property of the DSM-Flat representation, as specific points of difference between events of traces of different class can be directly detected rather than manually searching for the differences within the often

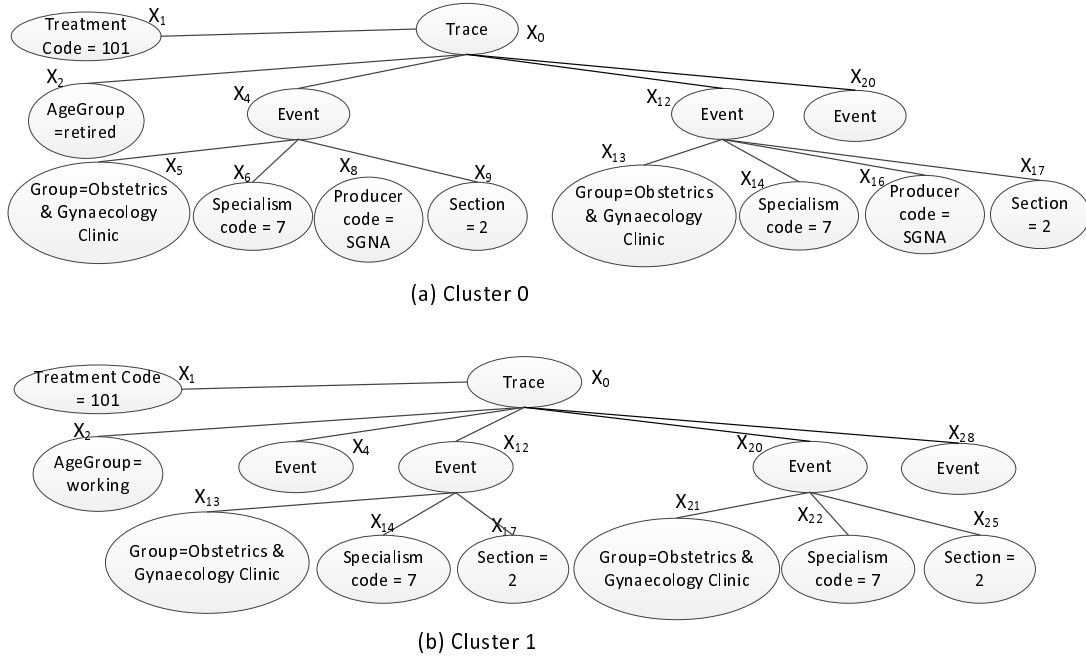


Figure 7: Frequent subtrees identified by DSM-LCM method

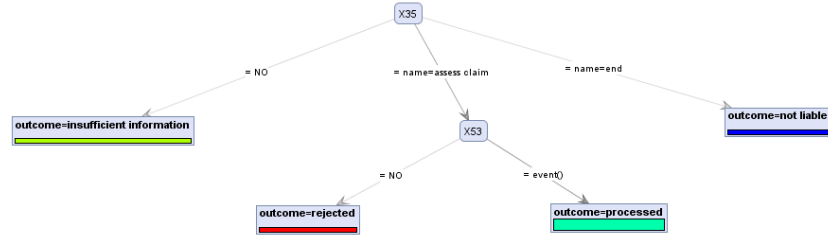


Figure 8: Decision tree for outcome of a process instance

large frequent pattern set in case of frequent subtree mining application.

5.3 Telephone repair dataset

The synthetic telephone dataset describes artificial logs of telephone repair processes. One example of a process instance starts by a customer registering a telephone for repair; the telephone is analysed; then transferred to either simple repair team or complex repair team; at the same time the customer is informed of the condition of their device; once the telephone is repaired it is tested; if not fixed it is then sent back for repair; the case is archived after the telephone is fixed. In this dataset we try to predict (1) the time needed to repair each telephone (class values were 0, 1 and 2 hours) and (2) the complexity of the repair which can be *simple*, *complex* or *both*. Note that the number of attributes in each event of this dataset was the same but some of them could differ in order. Hence, in this case they were first sorted in the same order to ensure that columns in the flat representation contain values of the same attribute. The structural properties of the tree database are as follows: |transactions|= 1104; avg. encoding length = 106; max. tree size = 112; avg. tree height = 2; avg. tree fan-out = 4.5; avg. tree size = 53.5; max tree height = 2; max. tree fan-out = 27.

Prediction of duration: The resulting C4.5 decision tree model (took 6s to build and evaluate) has the size of 69 with 58 leaf nodes and accuracy

of 82.7% evaluated using ten-fold cross-validation. Due to its large size, the decision tree is not shown here. Inspecting the rules of the decision tree shows that the duration of a process is classified as 0 if $X_{90} = X_{67} = X_{41} = \text{NO}$. Note that X_{90} contains the 4th attribute (*number of repair*) of the 8th event, X_{67} contains the 5th attribute (*defect fixed*) of the 13th event and X_{41} contains the 5th attribute (*defect fixed*) of the 8th event node. X_{27} , which corresponds to the 1st attribute (*resource*) of the 6th event, names the officer responsible for the repair at that stage. From the decision model, we observe that officer *SolverC3* performed poorly as most of his/her tasks are completed in one hour while others finish in less than one hour. While this is a synthetic dataset, this kind of analysis is also useful for detecting performance bottlenecks and investigating into resource optimization.

Prediction of complex process: a trace is labeled *simple* (or *complex*) if it contains any *simple* (or *complex*) repair, as part of the descriptive attributes of actions within an event. If both *simple* and *complex* repairs exist then the trace is labeled *both*. The resulting decision tree for the three-class classification evaluated using ten-fold cross-validation is displayed in Fig. 9. It took 2 seconds to build and evaluate the model and the classification accuracy is at 92.94

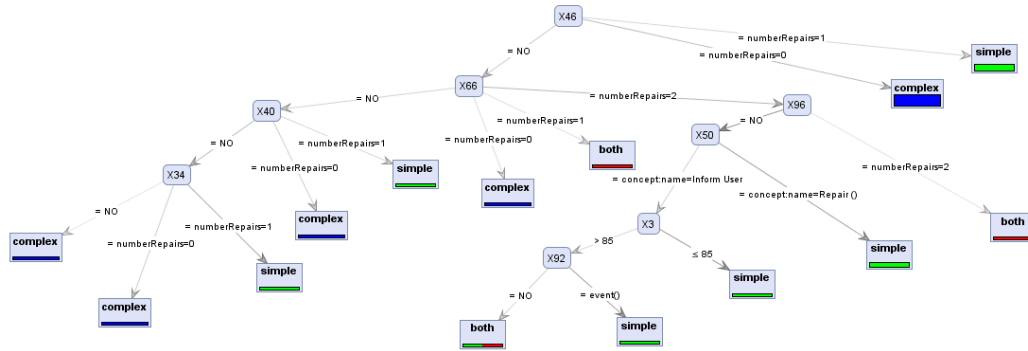


Figure 9: Decision tree for complexity prediction of a process instance

6 Conclusion and Future Work

Providing that process logs are increasingly available in XML format, this paper introduces an approach for direct application of a wide range of data mining/analysis methods to tree structured process logs. In the experiments using two synthetic XES datasets, decision tree learning is applied to directly detect all discriminating criteria. For the experiment on real world hospital process log, a combination of clustering, frequent pattern mining and classification techniques is used. We demonstrate how using the proposed approach one can detect and characterize different groups representing different process executions within the business, as well as detect the discriminating characteristics between the different groups. We have also illustrated some important differences and implications for process log analysis between subtrees extracted using the traditional subtree mining approach and the DSM approach which takes the node position into account. In our future work we will explore the use of the proposed approach for exception detection/analysis/prediction, model conformance checking, and in general discovery of patterns encompassing broad aspects of processes.

References

- Agrawal, R., Gunopulos, D. & Leymann, F. (1998), Mining process models from workflow logs, in 'Proceedings of the 6th International Conference on Extending Database Technology: Advances in Database Technology', EDBT '98, Springer-Verlag, London, UK, UK, pp. 469–483.
- Aguilar-Savén, R. S. (2004), 'Business process modelling: Review and framework', *International Journal of Production Economics* **90**(2), 129 – 149.
- Bose, R. P. J. C. & Aalst, W. M. P. (2012), Analysis of patient treatment procedures, in F. Daniel, K. Barkaoui, S. Dustdar, W. Aalst, J. Mylopoulos, M. Rosemann, M. J. Shaw & C. Szyperski, eds, 'Business Process Management Workshops', Vol. 99 of *Lecture Notes in Business Information Processing*, Springer Berlin Heidelberg, pp. 165–166.
- Bose, R. P. J. C. & van der Aalst, W. M. P. (2009), Context aware trace clustering: Towards improving process mining results., in 'SDM', SIAM, pp. 401–412.
- Chi, Y., Muntz, R. R., Nijssen, S. & Kok, J. N. (2004), 'Frequent subtree mining - an overview', *Fundam. Inf.* **66**(1-2), 161–198.
- Chi, Y., Xia, Y., Yang, Y. & R. Muntz, R. (2005), 'Mining closed and maximal frequent subtrees from databases of labeled rooted trees', *IEEE Trans. on Knowl. and Data Eng.* **17**(2), 190–202.
- Cook, J. E. & Wolf, A. L. (1995), Automating process discovery through event-data analysis, in 'Proceedings of the 17th international conference on Software engineering', ICSE '95, ACM, New York, NY, USA, pp. 73–82.
- De Medeiros, A. K. A., Guzzo, A., Greco, G., Van Der Aalst, W. M. P., Weijters, A. J. M. M., Van Dongen, B. F. & Saccà, D. (2008), Process mining based on clustering: a quest for precision, in 'Proceedings of the 2007 international conference on Business process management', BPM'07, Springer-Verlag, Berlin, Heidelberg, pp. 17–29.
- Goedertier, S., Martens, D., Baesens, B., Haesen, R. & Vanthienen, J. (2008), Process mining as first-order classification learning on logs with negative events, in 'Proceedings of the 2007 international conference on Business process management', BPM'07, Springer-Verlag, Berlin, Heidelberg, pp. 42–53.
- Gonçalves, M. A., Luo, M., Shen, R., Ali, M. F. & Fox, E. A. (2002), An xml log standard and tool for digital library logging analysis, in 'Proceedings of the 6th European Conference on Research and Advanced Technology for Digital Libraries', ECDL '02, Springer-Verlag, London, UK, UK, pp. 129–143.
- Greco, G., Guzzo, A., Manco, G. & Sacca, D. (2005), 'Mining and reasoning on workflows', *IEEE Trans. on Knowl. and Data Eng.* **17**(4), 519–534.
- Greco, G., Guzzo, A., Pontieri, L. & Sacca, D. (2006), 'Discovering expressive process models by clustering log traces', *IEEE Trans. on Knowl. and Data Eng.* **18**(8), 1010–1027.
- Günther, C. W. & van der Aalst, W. M. P. (2006), A generic import framework for process event logs, in 'Proceedings of the 2006 international conference on Business Process Management Workshops', BPM'06, Springer-Verlag, Berlin, Heidelberg, pp. 81–92.
- Günther, C. W. & Van Der Aalst, W. M. P. (2007), Fuzzy mining: adaptive process simplification based on multi-perspective metrics, in 'Proceedings of the 5th international conference on Business process management', BPM'07, Springer-Verlag, Berlin, Heidelberg, pp. 328–343.

- Hadzic, F. (2011), A structure preserving flat data format representation for tree-structured data, in 'PAKDD QIMIE Workshop', Shenzhen, China, pp. 221–233.
- Hadzic, F., Hecker, M. & Tagarelli, A. (2011a), Xml document clustering using structure-preserving flat representation of xml content and structure, in 'Proceedings of the 7th international conference on Advanced Data Mining and Applications - Volume Part II', ADMA'11, Springer-Verlag, Berlin, Heidelberg, pp. 403–416.
- Hadzic, F., Tan, H. & Dillon, S. T. (2011b), *Mining of Data with Complex Structures*, Springer.
- Hadzic, F., Tan, H. & Dillon, T. (2008), Mining unordered distance-constrained embedded subtrees, in 'Proceedings of the 11th International Conference on Discovery Science', DS '08, Springer-Verlag, Berlin, Heidelberg, pp. 272–283.
- Karypis, G. (2003), Cluto: A clustering toolkit, Technical report, University of Minnesota.
- Kim, H., Kim, S., Weninger, T., Han, J. & Abdelzaher, T. (2010), Ndpmine: efficiently mining discriminative numerical features for pattern-based classification, in 'Proceedings of the 2010 European conference on Machine learning and knowledge discovery in databases: Part II', ECML PKDD'10, Springer-Verlag, Berlin, Heidelberg, pp. 35–50.
- Kim, K. (2006), A xml-based workflow event logging mechanism for workflow mining, in 'Proceedings of the 2006 international conference on Advanced Web and Network Technologies, and Applications', APWeb'06, Springer-Verlag, Berlin, Heidelberg, pp. 132–136.
- Kutty, S., Nayak, R. & Li, Y. (2011), Xml documents clustering using a tensor space model, in 'Proceedings of the 15th Pacific-Asia conference on Advances in knowledge discovery and data mining - Volume Part I', PAKDD'11, Springer-Verlag, Berlin, Heidelberg, pp. 488–499.
- Maruster, L., Weijters, A. J. M. M., Aalst, W. M. P. v. d. & Bosch, A. v. d. (2002), Process mining: Discovering direct successors in process logs, in 'Proceedings of the 5th International Conference on Discovery Science', DS '02, Springer-Verlag, London, UK, UK, pp. 364–373.
- Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M. & Euler, T. (2006), Yale: rapid prototyping for complex data mining tasks, in 'Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining', KDD '06, ACM, New York, NY, USA, pp. 935–940.
- Quinlan, J. (1993), *Programs for Machine Learning*, Morgan Kaufmann, San Mateo, CA.
- Rozinat, A. & van der Aalst, W. M. P. (2006), Decision mining in prom, in 'Proceedings of the 4th international conference on Business Process Management', BPM'06, Springer-Verlag, Berlin, Heidelberg, pp. 420–425.
- Tan, H., Hadzic, F., Dillon, T. S., Chang, E. & Feng, L. (2008), 'Tree model guided candidate generation for mining frequent subtrees from xml documents', *ACM Trans. Knowl. Discov. Data* **2**(2), 9:1–9:43.
- Tiwari, A., Turner, C. J. & Majeed, B. (2008), 'A review of business process mining: state-of-the-art and future trends', *Business Process Management Journal* **14**, 5–22.
- Uno, T., Kiyomi, M. & Arimura, H. (2004), LCM ver.2: Efficient mining algorithms for Frequent/Closed/maximal itemsets, in 'Proc. 1st Int'l workshop on open source data mining: frequent pattern mining implementations'.
- van der Aalst, W. M. P. (2011), *Process Mining - Discovery, Conformance and Enhancement of Business Processes*, Springer.
- van der Aalst, W., Weijters, T. & Maruster, L. (2004), 'Workflow mining: Discovering process models from event logs', *IEEE Trans. on Knowl. and Data Eng.* **16**(9), 1128–1142.
- Verbeek, H.M.W., B. J. D. B. v. . A. W. v. d. (2011), Xes, xesame, and prom 6, in E. E. Soffer, P. & Proper, ed., 'Information Systems Evolution (CAiSE Forum 2010)', Vol. 72, Springer Berlin, pp. 60–75.
- Weijters, A. J. M. M. & van der Aalst, W. M. P. (2003), 'Rediscovering workflow models from event-based data using little thumb', *Integr. Comput.-Aided Eng.* **10**(2), 151–162.
- Zaki, M. J. (2005), 'Efficiently mining frequent trees in a forest: Algorithms and applications', in *IEEE Transaction on Knowledge and Data Engineering* **17**, 1021–1035.

Associative Classification using a Bio-Inspired Algorithm

Omar S. Soliman¹

Roba Bahgat²

Amr Adly³

^{1,2,3} Faculty of Computers and Information, Cairo University,
5 Ahmed Zewal Street, Orman, Giza, Egypt,
e-mail: {Dr.omar.soliman@gmail.com}

Abstract

This paper proposes an ambitious bio-inspired algorithm for associative classification (AC) based on Quantum-Inspired Artificial Immune system (QAIS) for building an efficient classifier by searching association rules to find the best subset of rules for all possible association rules. It integrates concepts of quantum computing (QC) and artificial immune system (AIS) as a bio natural inspired algorithm. It employs a mutation operator with a quantum-based rotation gate to control and maintain diversity, and guides the search process. The proposed QAIS is implemented and evaluated using benchmark datasets (Blake & Merz 1998) including Adult, Nursery, Iris and Breast-Cancer datasets. The obtained results are analysed and compared with experimental implementation results of AIS-AC algorithm (Do et al 2009). The experimental results showed that the proposed algorithm is performed well with large search space and has higher accuracy, and maintained diversity.

Keywords: Associative classification, Q-gate operator, QC, AIS, Bio-inspired optimization algorithms.

1 Introduction

Associative classification (AC) has shown a great dominance over many classification techniques. Associative classification uses association rule mining for rules discovery process to identify data class labels. Associative classification also integrates the rule discovery and classification process to build the classifier that supports in decision making process. The main advantages of the associative classification approaches is to discover high quality association rules in a very large space of candidate rules and integrate these rules with the classification process efficiently.

Bio-inspired optimization algorithms (BIOAs) represent a set of computational intelligence paradigms in machine learning, computer science and some engineering disciplines, which model various natural phenomena like the concept of evolution and the behavioral pattern displayed by various species. BIOAs serve as an attractive alternative for solving complex problems which can't be solved by the usual techniques. These BIOAs include Genetic Algorithms

(GAs), Particle Swarm Optimization (PSO), and Differential Evolution (DE), Artificial neural networks (ANN), Artificial Immune system (AIS), Fuzzy logic, Rough computing and quantum computing have been applied in various applications domain including decision support systems, data mining and knowledge discovery.

Artificial Immune Systems (AIS) have emerged during the last decade, Artificial immune systems can be defined as a computational system that is inspired by theoretical immunology, observed immune principles and mechanisms. The AIS uses the population-based search model of evolutionary computation algorithms that it is regarded as a suitable way for dealing with complex search space.

Quantum-Inspired Artificial Immune System (QAIS) is firstly introduced based on clonal selection algorithm and some concepts of quantum computing and proved that it is more effective than the immune operation. In the last decade, we could notice that there is a great interest in studying biologically inspired systems as artificial neural networks, evolutionary computation, DNA computation, and recently artificial immune systems (AIS). An immune system is biological system within an organism that protects it against disease by detecting and killing pathogens. It consists of a complex of cells, molecules and organs and It has the ability to distinguish antigen and antibody. It has three immunological principles the immune network theory, negative selection mechanism, and clonal selection principle. In this paper we focus in clonal selection principle and mutation operator using quantum theory.

Nowadays, Immune system applications spread in many fields as data mining, production,... etc since it has some features like learning, memory acquisition, pattern recognition, diversity generation, noise tolerance, detection and optimization. Associative classification uses association rules to predict data class label. The main issue with the associative classification approach is the high quality association rules discovery in a very large space of candidate rules and incorporating these rules in the classification process by an efficient way so applying QAIS for associative classification will be useful because we will get the benefit of immune system features and quantum computing contribution.

So, the main aim of this paper is to develop a bio-inspired algorithm for associative classification for building an efficient classifier by searching association rules to find the best subset of rules for all possible association rules. The rest of this paper is organized as follows. Section 2 presents the related work of the QAIS and AC with evolutionary algorithms besides problems and issues. The proposed algorithm is pre-

sented in section 3. The experiments setup and results are presented in section 4, where the last section is devoted to conclusions and further researches.

2 Problem Background and Related Works

Quantum-Inspired Immune system is introduced by Yangyang and Licheng, they proposed a new immune clonal algorithm, called a quantum inspired immune clonal algorithm (QICA), based on the concept and principles of quantum computing, such as a quantum bit and superposition of states. QICA uses a quantum bit, the smallest unit of information (Li & Jiao 2005). A multiuser detection application is proposed using Quantum Immune system by Yangyang et al (Li et al 2006). Research in Quantum Immune and its applications has been increased in the last years, Qun et al proposed a quantum-inspired immune algorithm (QIA) for Hybrid flow shop problems (HFSP) to minimize Makespan which have been proved to be NP-hard in when the objective is to minimize the Makespan (NiU et al 2009). Soliman and Adly proposed an ambitious algorithm based on Quantum-Inspired Immune system (QAIS) for building an efficient classifier by searching association rules to find the best subset of rules for all possible association rules (Soliman & Adly 2012).

Researchers also apply the Quantum Immune algorithm in the Multi-objective optimization area, Gao et al, proposed a novel quantum-inspired artificial immune system (MOQAIS) is presented for solving the multi-objective 0-1 knapsack problem (MKP), their algorithm is composed of a quantum-inspired artificial immune algorithm (QAIS) and an artificial immune system based on binary encoding (BAIS) (Gao et al 2010).

Another quantum immune algorithm is introduced for finding Pareto-optimal solutions to multiobjective optimization problems based on quantum computing and immune system. Experimental results showed that the MOQAIS algorithm is able to find a much better spread of solutions and has better convergence near the true Pareto-optimal front compared to the vector immune algorithm (VIS) and the elitist non-dominated sorting genetic system (NSGA-II) (Gao & Wang 2011). Qiaoyu et al introduced a new kind of quantum immune clonal algorithm for continuous space optimization. They updated quantum bits using quantum rotation gate to accelerate convergence and mutation is performed by quantum non-gate to avoid hasty convergence (Qiaoyu et al 2010). Lian et al proposed an immune-inspired quantum genetic optimization algorithm (IQGOA) based on clonal selection algorithm. Their Experimental results have shown that it is superior to clonal selection algorithm and Genetic Algorithm (GA) on performance (Lian 2011). Wang et al present a load balancing strategy based on Quantum Immune Evolutionary algorithm to optimize loading distribution by quantum coding and quantum evolution operator. It ensures the diversity of population by using immune operator vaccinations and immune selection when quantum is into the local optimum (Su & Wang 2011). The first Associative Classification approach was introduced with the classification based on associations (CBAs) algorithm. He integrated the two mining techniques classification and association rule mining. The integration is done by focusing on a specific subset of association rules whose right-hand-side are restricted to the classification class attribute and they refer to this subset of rules as the class association rules (CARs) (Ma 1998). Based on U-Apriori algo-

rithm and CBA algorithm, propose an associative classifier for uncertain data, uCBA (uncertain Classification Based on Associative), which can classify both certain and uncertain data. Their algorithm redefines the support, confidence, rule pruning process and classification strategy of CBA (Qin et al 2010). Mamta et al proposed a new model (associative classifier) based on weightage and utility for useful mining of substantial class association rules. This model uses the CBA-RG algorithm to produce a set of class association rules from a database and as well as exploits the downward closure property of the a priori algorithm (Punjabi et al 2011). A new associative classification method called CMAR, classification based on Multiple Association Rules. The method extends an efficient frequent pattern mining method, FP-growth, constructs a class distribution-associated FP-tree, and mines large database efficiently (Li et al 2001). Classification based on Predictive Association Rules, (CPAR), is developed by Yin and Han at 2003. CPAR depends on a greedy algorithm to generate rules directly from training data. Instead of generating a large number of candidate selection rules (Han 2003). Some predictive rule mining techniques such as CPAR, PRM and FOIL with statistical and Laplace as rule evaluation measures for predicting Tuberculosis. CPAR and PRM were better than FOIL and also statistical measure results in less generation time compared to Laplace measure (Asha et al 2011). An efficient algorithm to solve a specific problem called (the SSR-CARM problem) in binomial time $O(k^2n^2)$ which avoids selecting all k significant rules in a one-by-one manner (Wang et al 2005).

Decision trees are proposed to summarize associative classification rules. The proposed classification model benefit from the advantages of associative classification and decision trees (Chen & Hung 2009). They proposed a novel associative classification model, which first mines multi-class classification information from need-rating data, then constructs a rating classifier, and finally predicts customers' ratings for products (Jiang et al 2010). A new associative classification algorithm based on weighted voting (ACWV). It takes into account both the quality and number of rules instead of relying on only several high-quality rules (Zhu et al 2010). An associative classifier algorithm using demand-driven, so that the corresponding algorithm achieves high classification performance even in the case of limited labelling efforts (Velooso & Meira 2011). With an effective approach to building compact and accurate associative classification – Gain-based Association Rule Classification (Chen, Liu, Yu, Wei, & Zhang, 2006) in forms of association rules, they explore a way of fuzzy extension to GARC in dealing with the problem caused by crisp partitions for continuous attribute domains in data (Chen et al 2011). A fuzzy associative classification model based on variant apriori and multi-objective evolutionary algorithm NSGA-II (MOEA-FACM) is proposed. MOEA-FACM adopts fuzzy confirmation measure based on probabilistic dependence to assess fuzzy associative rule in order to generate good quality rule set. Then a small number of fuzzy associative rules are selected from the pre-screened candidate rule set using NSGA-II (Weigang & Xiuli 2011).

Also, Mangalampalli proposed a fuzzy associative classification algorithm for object class detection in images using interest points which relies only on the positive class for training (Mangalampalli et al 2010). Dixit studied and optimized an artificial immune system based classification system. They evaluated the

performance of the AIS based classification system by computing accuracy at different clonal factors and varying number of generations. They used three standard datasets to compute the accuracy. They found that the system gives highest accuracy with clonal factor 0.4 (DIXIT & CHANDEL 2011). The clonal selection algorithm for Associative Classification (AC) is investigated and proposed a new approach known as AIS-AC for mining association rules effectively for classification with treating the rule mining process as an optimization problem of finding an optimal set of association rules according to some predefined constraints. AIS-AC approach is efficient in dealing with the complexity problem on the large search space of rules and It avoided searching greedily for all possible association rules, so it could find an effective set of associative rules for classification (Do et al 2005, 2009).

Association Classification Rules Mining problem is treated as a multi-objective problem rather than a single objective one. They developed a binary multi-objective particle swarm optimization model to optimize the measures like coverage and confidence rules for rule discovery then a small number of rules are targeted from the extracted rules to design an accurate and compact classifier which can maximize the accuracy of the rule sets and minimize their complexity simultaneously (Das et al 2011). Shahzad presented a hybrid classification algorithm called ACO-AC, combining the idea of association rules mining and supervised classification using Ant Colony Optimization (ACO). ACO is used to mine only effective subset of class association rules instead of searching for all possible rules in large search space. The mining process stops when the discovered rules achieves a minimum coverage threshold (Shahzad 2010).

3 Proposed algorithm

In our proposed algorithm we deal with the associative algorithm process as an optimization problem to find the optimal (best) classification association rules (CARs) that will build the classifier. Rules discovery process differs from the rule discovery in the basic association rule mining algorithm. We search classification association rules using quantum-inspired Immune system. We considered each CAR as an immune cell and each generation is a set of class association rules.

Rule Selection process implicitly consists of two parts rule discovery (generation) and rule evaluation (selection). Rule discovery come from the testing dataset starting from initial population then memory population increases through generations. We search for rules with the highest confidence values and confidence measure as the affinity in immune system so we select rules with high affinity and the selected rules should satisfy the support constraint to filter specific rules from the population before the selection process. We terminate this process when generation count equals the number of generations and classification process applied after getting the CARs from memory pool. We build the classifier and apply it on benchmark datasets then evaluate its accuracy.

3.1 Algorithm Steps

The main steps of the proposed algorithm are described in Algorithm 1. Where the detailed description of these steps are introduced in the following subsections.

3.2 Selection process

In this process, rules with the support less than min-Support threshold are eliminated; where *selection-Number of* rules with the highest confidence are selected. The selection criteria is based on the best fitness.

Algorithm 1 QAIS for Associative Classification

```

1: Initialize a random population of rules  $P_0$  of size  $n$ , set the memoryset  $M = \Phi$  and  $gCount = 0$ 
2: while ( $gCount < NoOfGenerations$ ) do
3:   for each rule  $R$  in  $p$  do
4:     if  $Support(R) < minSupport$  then
5:       Remove Rule  $R$  from  $P$ 
6:     end if
7:   end for
8:   Sort rules in  $P$  according to affinity value in descending order.
9:   Select the first selectionNo affinity rules and insert them into  $P$  instead other rules.
10:  Clone  $P$  by cloning best  $nClones$  rules.
11:  Mutate  $P$  by algorithm 2.
12:  Prune rules inside  $P$ .
13:  for each rule  $R$  in  $p$  do
14:    if  $conf(R) > minConfidence$  then
15:      Insert  $R$  into  $M$ 
16:    end if
17:  end for
18:  Reselect randomly new population ( $P_{new}$ ) from  $M$  and make  $P = P_{new}$ 
19:  Increment  $gCount$  by one.
20: end while
    
```

3.3 Cloning Process

The cloning process is performed by using the clonal rate of a rule is directly related to the affinity value (confidence) of the. We denote clonal rate of a rule as CR.

Clonal rate of a rule is directly proportional to the affinity of the rule so the clones directly depend on affinity value so we pick proportion of the population $nClones$ to be cloned. $nClones$ is calculated as follows:

Given *selectionNo* is the best affinity selected rules and the dataset size N then calculate clonal factor (cf) the proportion of *selectionNo* in the dataset.

$$cf = \frac{selectionNo}{N} \quad (1)$$

and $nClones$ get by the following equation:

$$nClones = cf * selectionNo. \quad (2)$$

Finally we make clones of the best $nClones$ rules.

3.4 Mutation process

In this process, each cell is mutated using quantum-based rotation Q-gate operator, a high probability is given for each low affinity cell to be mutated more than high affinity cell and then a new offspring is produced.

3.4.1 Mutation Operator

As mentioned above, the used mutation operator is a quantum-based rotation Q-gate. By selecting a subset $C_s \subseteq C$ where C is the population rules and

assume there exists a parent cell

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \dots & \alpha_n \\ \beta_1 & \beta_2 & \beta_3 & \dots & \beta_n \end{bmatrix} \quad (3)$$

where:

$$|\alpha_{i2}|^2 + |\beta_{i2}|^2 = 1, i = 1, 2, \dots, n. \quad (4)$$

The item j is selected randomly from cell i such that

$$\begin{bmatrix} \alpha'_j \\ \beta'_j \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \alpha_j \\ \beta_j \end{bmatrix} \quad (5)$$

where:

$$\theta_j = \theta_{j-1} + p_j * \delta \quad (6)$$

$$\delta = \begin{cases} U(0, \theta_j) & , p_j = -1 \\ U(\theta_j, \frac{\pi}{2}) & , p_j = +1 \end{cases} \quad (7)$$

θ_j is the rotated angle

The rotated angle is calculated as follows:

$$\theta_j = \arctan\left(\frac{\beta_j}{\alpha_j}\right). \quad (8)$$

then recalculate support and confidence of mutated solutions. The main steps of the quantum-based rotation Q-gate mutation are described in algorithm 2.

3.5 Pruning Process

In this process, covered rules by the memory set rules are pruned to insure that those covered rules will not be exist in the coming generations and reduce complexity by eliminating redundancy.

3.5.1 Pruning Criteria:

Rule $R^* : itemset^* \Rightarrow c$ is covered by rule $R : itemset \Rightarrow c$ if the following condition is satisfied:

1. $itemset \subset itemset^*$ and
2. Confidence value of R is Greater than or equal R^* .

3.6 Reselection process

We reselect from memory set randomly to form the new population .The memory set contains the best uncovered rules with highest affinity value so the next generation will converge to the optimal solutions.

4 Experimental Results and Discussion

Data preprocessing is the initial step for any data mining algorithm. Data preprocessing is performed to convert the data in a specific format which can be easily dealt by the algorithm. The proposed algorithm is implemented and evaluated using benchmark datasets Adult, Nursery, Iris and Breast-Cancer from the UCI Machine Learning Repository (Blake & Merz 1998). Each record regards as an immune cell and each item has a predetermind possible values stored in itemsets population. The obtained results

Algorithm 2 Mutation process

- 1: **for** each cell $i \in C_s$ **do**
- 2: Given itemset S_j of possible values:

$$S_j = \{item_1, item_2, \dots, item_L\} \quad (9)$$

where: $L = size(S_j)$

- 3: **while** $L > 1$ **do**
- 4: $r = U(0, 1)$
- 5: **if** $r < (\alpha'_j)^2$ **then**
- 6: $S_j \leftarrow S_j\{1 : \lfloor L/2 \rfloor\}$
- 7: **else**
- 8: $S_j \leftarrow S_j\{\lfloor L/2 \rfloor + 1 : L\}$
- 9: **end if**
- 10: $L = size(S_j)$
- 11: **end while**
- 12: $selected_j = S_j\{1\}$
- 13: **end for**

DataSets	Avg Affinity of all Generations
Iris	0.988
B-Cancer	0.853
Nursery	0.926
Adult	0.768

Table 1: Average Affinity values of analysed Datasets

are compared with experiment implementation result of AIS-AC algorithm (Do et al 2009).

In mutation process, a point (item,value) are picked to be mutated. We get the "item" which is equal to the itemset population name and get a possible value from its itemsets population. The search criterion is the high value of affinity which is regarded as the algorithm compass. AIS is essentially based on the mutation operator so it can achieve a diverse number of local optima. All experiments are performed within 70 % minimum confidence (minConfidence) and minimum support values (minSupport) are %10,%5,%2.5,%0.6.

4.1 Affinity Analysis

The performance of the proposed algorithm is evaluated using affinity analysis for all datasets. The affinity values are recorded, averaged and visualized of each generation for all datasets including Adult, Iris, Nursery and Breast-Cancer. The obtained averaged affinity values for each dataset overall generations are reported in table 1.

Where the obtained average affinity values of each generation at various support value are visualized as shown in Figures 1 & 2 for all datasets. As shown in figure 1 the average affinity growth rate is increased through generations.

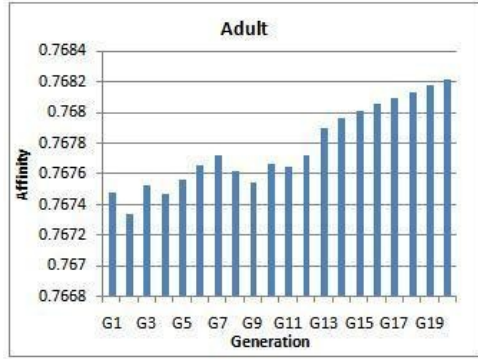
For Adult dataset with minConfidence and minSupport values, figure 1(a) showed that the affinity value is between 0.6 and 0.98 and it increases through generations with its growth rate which different for various support values.

For Iris dataset, as shown in figure 1(b) the affinity value is improved through runs with different support values and the average affinity value is between 0.9 and 0.99. The algorithm in Iris dataset explores the search space faster than any other dataset since the small size of data so it is clear that the lowest value of average affinity is 0.9.

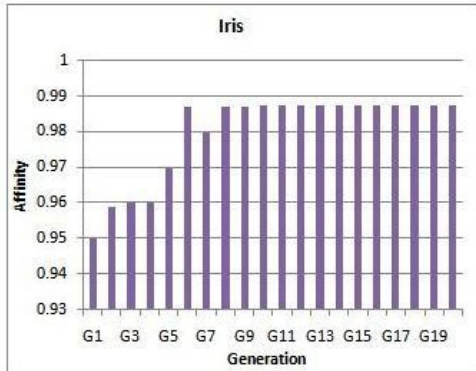
For Nursery dataset the affinity analysis is performed within minConfidence value %50 and minSupport values. We run the model with confidence equals

% 70 but we found that the model overfit the data, so we run with $\text{minConf} = 0.5$. As shown in figure 2(c) the affinity value is improved through runs and achieve good evolution and average affinity is between 0.91 and 0.98 so we have a small interval and small growth rate, but when the support value decreased the growth rate is increased since the lower support value will get small confidence value so the growth will be clear.

Finally, for Breast-Cancer dataset, figure 2(d) showed that the affinity value is improved through runs with different support values and the average affinity value is between 0.82 and 0.91. The affinity analysis showed the ability of the proposed algorithm to obtain higher affinity values and increased through generations for all datasets.



(a)



(b)

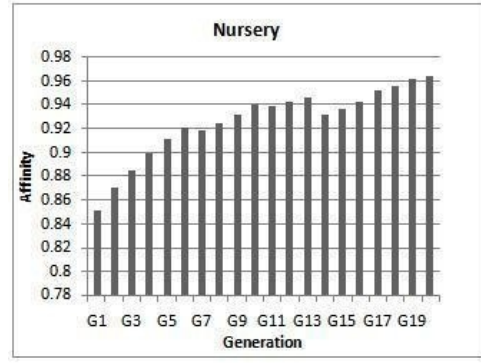
Figure 1: Affinity values for Adult & Iris datasets

4.2 Accuracy Analysis

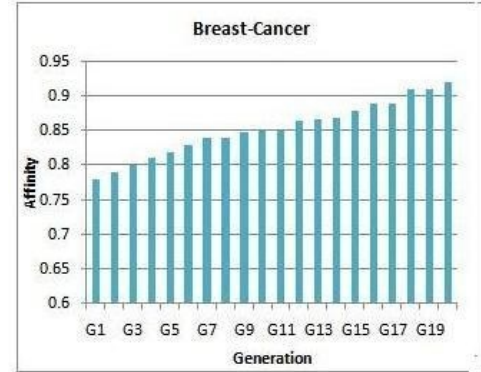
Suppose the accuracy when a rule $R : \text{itemset} \Rightarrow c$ is used to predict the class label of a transaction T is the probability of which c is the class label of the transaction that contains the itemset iset (Do et al 2009). Accuracy can be calculated by:

$$\text{accuracy}(R) = p(c \mid \text{itemset})$$

Now, the probability of which c is the class label of T that contains the itemset iset should be calculated. Each transaction T which contains iset can be regarded as a “trial”. If a trial belongs to c , then the outcome is “success,” otherwise the outcome is “false”. The accuracy of proposed algorithm is calculated using the probability of success on a random trial (Do et al 2009).



(c)



(d)

Figure 2: Affinity values for Nursery & Breast Cancer datasets

Support		0.1	0.05	0.025	0.006	0.003
Iris	AIS-AC	0.92	0.92	0.91	0.91	0.92
	QAIS	0.92	0.92	0.92	0.91	0.92
B-Cancer	AIS-AC	0.68	0.67	0.64	0.66	0.68
	QAIS	0.71	0.64	0.67	0.69	0.70
Nursery	AIS-AC	0.89	0.91	0.92	0.97	0.98
	QAIS	0.91	0.8	0.54	0.53	0.53
Adult	AIS-AC	0.74	0.77	0.77	0.7	0.72
	QAIS	0.77	0.79	0.73	0.7	0.71

Table 2: Accuracy values of analysed Datasets

The accuracy of the proposed algorithm is calculated and compared with AIS-AC (Do et al 2009) for all datasets as reported in table 2 with different minimum support values, and visualized as shown in figures 3 and 4. As reported 2 and showed in figures 3 and 4 the accuracy of QAIS is decreasing when the support value is decreasing that mean larger number of rules.

For example, when the support value equals 0.1, for Iris dataset figure 3(a), the accuracy of QAIS is 0.93 and AIS-AC is 0.91 then if we jump to the support value of 0.025, the accuracy will be 0.91 and 0.9 respectively. For Breast-Cancer figure 3(b) dataset the accuracy will be 0.7 and 0.67 respectively with support value equals 0.01 then if we jump to the support value of 0.025 the accuracy value will be 0.67 and 0.63.

For Nursery dataset, as shown in figure 4(c) when the support value equals 0.1, the accuracy of QAIS is 0.96 and AIS-AC is 0.91 then when we jump to the support value of 0.006, the accuracy will be 0.92 and 0.53 respectively. In Adult dataset as shown in figure 4(d) the accuracy values of QAIS are better than AIS-AC when the support values equal 0.01 and

0.1, and the accuracy of both algorithms are equal when support value equal 0.006.

The analysis of obtained accuracy results showed that the accuracy of QAIS is better than AIS-AC over all data sets, and it is decreasing for both algorithms when support value is decreasing but QAIS is decreasing with lower rate.

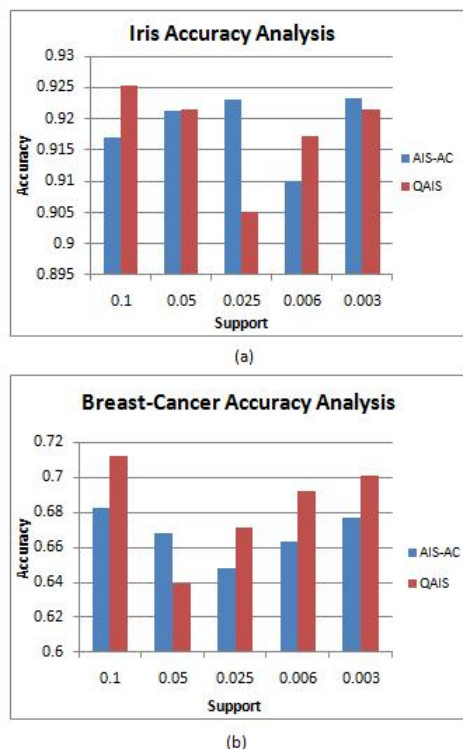


Figure 3: Accuracy values for Iris & Breast Cancer datasets

5 Conclusion

In this paper, a bio-inspired algorithm for associative classification is proposed. It is based on the clonal selection theory and quantum theory. The proposed algorithm generates association rules efficiently for classification process in a large search space. The Q-gate mutation operator is employed to control diversity of immune cells in the search space and guide the search process. The proposed algorithm able to deal with complex search space of association rules. The algorithm is implemented and evaluated for benchmark dataset. The obtained results are compared with results of AIS-AC and showed that the proposed algorithm is performed well and has significant accuracy and average affinity values. It evaluates discovered rules after each generation and eliminates bad rules from memory set.

For further research, quantum-inspired immune system can be enhanced by applying quantum cloning operator in addition to mutation operator, as well as more experiments.

References

Asha, T. et al. (2011), A Study of Associative Classifiers with Different Rule Evaluation Measures for Tuberculosis Prediction. *International Journal of Computer Applications*, (3), pp. 15-20. pp. 207-216.

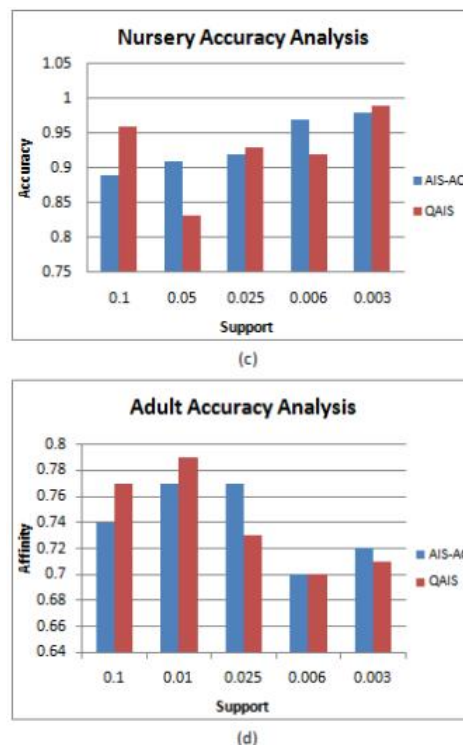


Figure 4: Accuracy values for Nursery & Adult datasets

Blake, C. & Merz, C.J.(1998), UCI Repository of machine learning databases.

Chen, G., Xiong, Y. & Wei, Q. (2011), A Fuzzy Extension to Compact and Accurate Associative Classification. *35 Years of Fuzzy Set Theory*, pp 171-193.

Chen, Y.L. & Hung, L.T.H.(2009), Using decision trees to summarize associative classification rules. *Expert Systems with Applications*, 36(2), pp 2338-2351.

Das, M., Roy, R., Dehuri, S. & Cho, S.B. (2011), A New Approach to Associative Classification Based on Binary Multi-Objective Particle Swarm Optimization. *International Journal of Applied Metaheuristic Computing (IJAMC)*, 2(2), pp 51-73.

DIXIT, S. & CHANDEL, G.S. (2011), Optimizing Mining Association Rules for Artificial Immune System based Classification. *International Journal of Engineering Science*, 3, pp 6732-6738.

Do, T., Hui, S. & Fong, A. (2005), Artificial immune system for associative classification. *Advances in Natural Computation*, pp 428-428.

Do, T.D. , Hui, S.C. , Fong, ACM & Fong, B. (2009), Associative classification with artificial immune system. *Evolutionary Computation*, *IEEE Transactions on*, 13(2), pp 217-228.

Gao, J. & Wang, J. (2011), A hybrid quantum-inspired immune algorithm for multiobjective optimization. *Applied Mathematics and Computation*, 217(9), pp 4754-4770.

Gao, J., Fang, L. & He, G.(2010), A quantum-inspired artificial immune system for multiobjective 0-1 knapsack problems. *Advances in Neural Networks-ISBN 2010*, pp 161-168.

Han, J. (2003), CPAR: Classification based on predictive association rules. In *Proceedings of the Third SIAM International Conference on Data Mining*. pp 331-335.

Jiang, Y., Shang, J. & Liu, Y.(2010), Maximizing customer satisfaction through an online recommendation system: A novel associative classification model. *Decision Support Systems*, 48(3), pp 470-479.

- Li, W., Han, J. & Pei, J. (2001), CMAR: Accurate and efficient classification based on multiple class-association rules. In Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on. pp 369-376.
- Li, Y. & Jiao, L. (2005), Quantum-inspired immune clonal algorithm. Artificial Immune Systems, pp 304-317.
- Li, Y., Jiao, L. & Gou, S. (2006), Quantum-inspired immune clonal algorithm for multiuser detection in DS-CDMA systems. Simulated Evolution and Learning, pp 80-87.
- Lian, Z. (2011), Immune-Inspired Quantum Genetic Optimization Algorithm and its Application. Advanced Materials Research, 143, pp 547-551.
- Ma, B.L.W.H.Y. (1998), Integrating classification and association rule mining. In Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, pp 80-86
- Mangalampalli, A., Chaoji, V. & Sanyal, S. (2011), I-FAC: Efficient fuzzy associative classifier for object classes in images. In Proceedings of the 2010 20th International Conference on Pattern Recognition. pp 4388-4391.
- NiU, Q., Zhou, T. & Ma, S. (2009), A quantum-inspired immune algorithm for hybrid flow shop with makespan criterion. Journal of Universal Computer Science, 15(4), pp 765-785.
- Punjabi, M., Kushwaha, V. & Ranjan, R. (2011), Exploring Associative Classification, Technique Using Weighted Utility Association Rules for Predictive Analytics. High Performance Architecture and Grid Computing, pp 169-178.
- Qiaoyu, Y., Weili, L. & Junci, C. (2010), Continuous quantum immune clonal optimization and its application to calculation and analysis of electromagnetic in induction motor. In Intelligent Computing and Intelligent Systems (ICIS), 2010 IEEE International Conference on. pp 364-368.
- Qin, X. et al. (2010), Associative classifier for uncertain data. Web-Age Information Management, pp 692-703.
- Shahzad, W. (2010), Classification and Associative Classification Rule Discovery Using Ant Colony Optimization. National University.
- Soliman, O.S. & Adly, A., (2012), Bio-inspired algorithm for classification association rules. In proceeding of the 8th International Conference on Informatics and Systems (INFOS2012), pp 154-160.
- Su, R.N. & Wang, Y. (2011), Load Balancing Strategy Based on Immune Quantum Evolutionary Algorithm. Jisuanji Gongcheng/ Computer Engineering, 37(2),pp 154-156..
- Veloso, A. & Meira, W. (2011), Self-Training Associative Classification. Demand-Driven Associative Classification, pp 87-95.
- Wang, Y.J., Xin, Q. & Coenen, F.(2005), Selection of Significant Rules in Classification Association Rule Mining. Foundation of Semantic Oriented Data and Web Mining, p 106-108.
- Weigang, H. & Xiuli, S.(2011), A Fuzzy Associative Classification Method Based on Multi-Objective Evolutionary Algorithm. Journal of Computer Research and Development, 48(4), pp 567-575.
- Zhu, X., Song, Q. & Jia, Z. (2010), A weighted voting-based associative classification algorithm. The Computer Journal, 53(6), pp 786-801.

An Iterative Two-Party Protocol for Scalable Privacy-Preserving Record Linkage

Dinusha Vatsalan and Peter Christen

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

Abstract

Record linkage is the process of identifying which records in different databases refer to the same real-world entities. When personal details of individuals, such as names and addresses, are used to link databases across different organisations, then privacy becomes a major concern. Often it is not permissible to exchange identifying data among organisations. Linking databases in situations where no private or confidential information can be revealed is known as ‘privacy-preserving record linkage’ (PPRL). We propose a novel protocol for scalable and approximate PPRL based on Bloom filters in a scenario where no third party is available to conduct a linkage.

While two-party protocols are more secure because there is no possibility of collusion between one of the database owners and the third party, these protocols generally require more complex and expensive techniques to ensure that a database owner cannot infer any sensitive information about the other party’s data during the linkage process. Our two-party protocol uses an efficient privacy technique called Bloom filters, and conducts an iterative classification of record pairs into matches and non-matches, as selected bits of the Bloom filters are revealed. Experiments conducted on real-world databases that contain nearly two million records, show that our protocol is scalable to large databases while providing sufficient privacy characteristics and achieving high linkage quality.

Keywords: Data matching, entity resolution, privacy, approximate matching, scalability, Bloom filter.

1 Introduction

Privacy-preserving record linkage (PPRL) is the problem of how to efficiently link different databases to identify records that correspond to the same real-world entities without revealing their identities to any party involved in the process, or to any external party or adversary. The three main challenges that a PPRL solution in a real-world context needs to address are (1) scalability to large databases by efficiently conducting the linkage; (2) achieving high quality of the linkage results through the use of approximate (string) matching and effective classification of compared record pairs into matches (two records that are assumed to correspond to the same entity) and non-

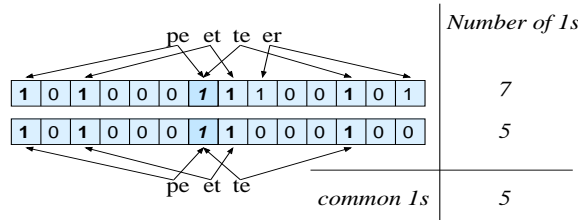
matches (two records that are assumed to correspond to two different entities); and (3) provision of sufficient privacy guarantees such that the interested parties only learn the records that were reconciled as referring to the same real-world entities (Christen 2012, Clifton et al. 2004, Hall & Fienberg 2010).

One example real-world PPRL application would be where a research team aims to study the correlations between types of car accidents and resulting injuries. Such an analysis requires the linkage of databases from hospitals, health insurance companies and the police (Christen 2012). Another example from the health domain is a health surveillance system that continuously links data from human health data, animal health data, and drugs data to monitor outbreaks of contagious diseases that could lead to epidemics or even pandemics (Clifton et al. 2004). Another application of current interest is where a national security agency needs to collect and link records from a diverse set of databases (such as communication providers, banks, airlines, immigration, and social security) to identify potential terrorism threats (Christen 2006, 2012, Clifton et al. 2004). These example scenarios illustrate that commonly data from different organizations need to be linked, but privacy and confidentiality issues often arise which might prevent such record linkage applications.

Several approaches have been proposed to deal with PPRL over the past two decades (Trepetin 2008, Verykios et al. 2009, Karakasidis & Verykios 2010, Durham et al. 2011, Vatsalan et al. 2013). These approaches can be classified into ‘three-party protocols’ and ‘two-party protocols’. Three-party protocols require a third party for performing the linkage while two-party protocols don’t (Christen 2006, 2009, Verykios et al. 2009). The main advantages of two-party protocols over three-party protocols are that they are more secure because there is no possibility of collusion between one of the database owners and the third party, and often they have lower communication costs. However, two-party protocols generally apply more complex techniques, such as Secure Multi-party Computation (SMC) (Clifton et al. 2002, Goldreich 2004, Lindell & Pinkas 2009), to ensure that the two database owners cannot infer any sensitive information from each other during the linkage process. The use of complex techniques, which are computationally intensive, makes PPRL solutions not scalable to large databases and thus are not applicable in real-world contexts.

Among several different privacy techniques that are applied in PPRL solutions, Bloom filters (Bloom 1970) are one efficient technique that can provide adequate privacy guarantees if effectively used. A Bloom filter is a bit string data structure of length l bits,

Copyright ©2012, Australian Computer Society, Inc. This paper appeared at the 10th Australasian Data Mining Conference (AusDM 2012), Sydney, Australia, December 2012. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 134, Yanchang Zhao, Jiuyong Li, Paul Kennedy, and Peter Christen, Ed. Reproduction for academic, not-for-profit purposes permitted provided this text is included.



$$\text{Dice-coefficient}(\text{peter}, \text{pete}) = 2 \times 5 / (5 + 7) = 0.83$$

Figure 1: Mapping of strings into Bloom filters and calculating their Dice coefficient similarity.

where all the bits are initially set to 0. k independent hash functions, h_1, h_2, \dots, h_k , each with range $1, \dots, l$, are used to map the elements of a set into the Bloom filter by setting the corresponding bit positions to 1. Bloom filters have previously been used in several three-party or multi-party PPRL solutions.

Schnell et al. (Schnell et al. 2009) were the first to propose a method for approximate matching in PPRL using Bloom filters. In their work, the attribute values of each record in the databases to be linked are concatenated into one string, and the q -grams (sub-strings of length q) of these strings are mapped into Bloom filters using k independent hash functions. These Bloom filters are sent to a third party and the Dice coefficient (Christen 2012) is used to calculate the similarity of two Bloom filters:

$$\text{sim}(b^A, b^B) = \frac{2c}{x^A + x^B} \quad (1)$$

where c is the number of common bit positions that are set to 1 in both Bloom filters b^A and b^B (common 1-bits), x^A is the number of bit positions that are set to 1 in b^A , and x^B is the number of bit positions that are set to 1 in b^B . The Dice coefficient is used since it is insensitive to many matching zeros in long Bloom filters (Schnell et al. 2009). For example, mapping the bigrams ($q = 2$) of the two string values ‘peter’ and ‘pete’ into $l = 14$ bits long Bloom filters using $k = 2$ hash functions and calculating the Dice coefficient similarity of the two Bloom filters are illustrated in Figure 1.

This approach requires a third party to perform the linkage, since each of the two database owners could mount a dictionary attack on the Bloom filters of the other party because they know the hash functions $h_1 \dots h_k$ and the length of the Bloom filters l . The approach is efficient because of the use of Bloom filters and it supports approximate matching of values as well, rendering it applicable to real-world conditions. However, as with other three-party protocols, collusion between the parties is a major security drawback of this approach (Schnell et al. 2009). Recent research in PPRL has analysed the weaknesses of Bloom filters in three-party settings using constraint satisfaction cryptanalysis (Kuzu et al. 2011), and novel solutions based on random sampling of bits from attribute level Bloom filters have been proposed (Durham 2012).

Our aim is to develop a two-party protocol for PPRL using Bloom filters. We propose a method that eliminates the need of a third party by iteratively revealing selected bits in the Bloom filters between two database owners, and classifying record pairs into matches and non-matches in an iterative way to reduce the number of record pairs with unknown match status at each iteration without compromising privacy.

Our paper contributes (1) a solution for PPRL in a two-party framework using Bloom filters that is feasible in real-world PPRL applications by addressing the three main challenges that a practical PPRL application poses; (2) an analysis of the proposed solution in terms of complexity, accuracy, and privacy; and (3) an empirical evaluation of the protocol using a large real-world Australian telephone database.

The remainder of the paper is structured as follows. In the following section, we provide an overview of related work in PPRL. In Section 3 we describe the steps of the protocol illustrated with two small sets of Bloom filters. In Sections 4 and 5 we analyse the protocol with regard to performance and privacy, and in Section 6 we validate these analyses through an experimental study. Finally we summarize our findings and discuss future research directions in Section 7.

2 Related Work

Various techniques for privately and efficiently calculating approximate similarities in PPRL have been proposed (Trepetin 2008, Verykios et al. 2009, Karakasidis & Verykios 2010, Durham et al. 2011, Vatsalan et al. 2013). There has been a variety of privacy techniques employed to facilitate PPRL. They include secure hash encoding (Dusserre et al. 1995, Van Eycken et al. 2000, Weber et al. 2012), generalization techniques (Kantarcioglu et al. 2008, Inan et al. 2008, Vatsalan et al. 2011, Mohammed et al. 2011, Karakasidis & Verykios 2012), SMC techniques (Song et al. 2000, Atallah et al. 2003, Ravikumar et al. 2004, Al-Lawati et al. 2005, Inan et al. 2008, 2010, Yakout et al. 2012), differential privacy (Inan et al. 2010), pseudo random functions (Song et al. 2000, O’Keefe et al. 2004, Freedman et al. 2005), Bloom filters (Lai et al. 2006, Schnell et al. 2009, Durham et al. 2010, Karakasidis & Verykios 2011, Durham 2012), reference values (Scannapieco et al. 2007, Pang et al. 2009, Vatsalan et al. 2011, Yakout et al. 2012), phonetic encoding (Karakasidis & Verykios 2011, Karakasidis et al. 2011), and random records (Kargupta et al. 2003, Karakasidis et al. 2011).

Most of the two-party solutions use SMC techniques for the private comparison. Atallah et al. (2003) proposed a two-party protocol where the edit distance algorithm is modified to provide privacy using SMC techniques. Ravikumar et al. (2004) used SMC techniques for the secure computation of several distance functions. The approach of Song et al. (2000) in a two-party context calculates enciphered permutations of values using pseudo random functions and SMC techniques for approximate matching of documents.

Inan et al. (2010) and Yakout et al. (2012) proposed two phase solutions where the first phase is the blocking phase that aims to reduce the number of candidate record pairs by removing pairs that are unlikely to be matches. The remaining candidate pairs are then compared in detail using SMC techniques in the second phase. Inan et al. used differential privacy to partition the perturbed datasets through statistical queries and then generate candidate record pairs from the records in the same partitions. Yakout et al. mapped all the records into a complex plane and then used a slab of a certain width to generate candidate record pairs.

Bloom filters were proposed by Bloom (1970) for efficiently checking set membership. Initially, Bloom filters have been used to support membership queries (Broder et al. 2002). More recently, they have also been used for computing similarities. Several ap-

Table 1: Notation used in this paper.

$\mathbf{D}^A, \mathbf{D}^B$	Databases held by database owners Alice and Bob, respectively
$\mathbf{S}^A, \mathbf{S}^B$	Lists of values of linkage attributes A for each record in \mathbf{D}^A and \mathbf{D}^B , respectively
b^A, b^B	A Bloom filter of each record in \mathbf{S}^A or \mathbf{S}^B , respectively
$\mathbf{O}^A, \mathbf{O}^B$	Lists of record IDs and number of 1-bits for each record in \mathbf{D}^A and \mathbf{D}^B , respectively
\mathbf{C}, \mathbf{C}_i	List of candidate record pairs, list of candidate record pairs at iteration i
s_t, s_l, s_r	Minimum similarity threshold value to classify a record pair as a match, minimum acceptable similarity threshold value to add random bits, and minimum similarity threshold value to reveal bits in an iteration
l	Length of Bloom filters
$h_1 \dots h_k, k$	Hash functions used to map a set of elements into a Bloom filter, number of hash functions
q	Number of characters that make a q -gram
i	Iteration i , $i > 0$
r, r_i	Number of bit positions revealed, number of bit positions revealed in iteration i
t_i	Total number of bit positions revealed so far up to iteration i , $t_i = \sum_i r_i$
x, x^A, x^B	Number of 1-bits, number of 1-bits in b^A or b^B , respectively
x_i, x_i^A, x_i^B	Total number of 1-bits revealed so far up to iteration i , total number of 1-bits revealed in b^A or b^B so far up to iteration i , respectively
$r_{min}, r_{max}, z_{max}$	Minimum number of bits that can be revealed in an iteration, maximum number of total bits to be revealed, maximum number of random bits that can be added
c_{min}, c_i	Minimum number of common 1-bits required in both Bloom filters b^A and b^B , total number of common 1-bits revealed from Bloom filters b^A and b^B so far up to iteration i
d, d_{max}	Difference between x^A and x^B , maximum difference between x^A and x^B to be classified as a ‘match’
$sim(\cdot, \cdot)$	Function used to calculate similarities between two Bloom filters b^A and b^B (Dice coefficient)

proaches have been suggested for similarity calculation in PPRL by using Bloom filters (Lai et al. 2006, Schnell et al. 2009, Durham et al. 2010, Karakasidis & Verykios 2011, Durham 2012).

Lai et al. (2006) proposed a multi-party approach that uses Bloom filters for private matching. In their approach, each party partitions its Bloom filters and sends a segment to the other party. The received segments are computed with a logical conjunction (and) and the partial resulting segments are exchanged between the parties. Each party checks its own full Bloom filters with the results and if the membership test is successful then it is considered to be a match. Though the cost of this approach is low since the computation is totally distributed between the parties and the creation and processing of Bloom filters are very fast, the approach is very sensitive to low quality data and is unable to perform approximate matching.

Schnell et al. (2009)’s three-party approach takes into consideration the problem of approximate matching based on a combination of q -grams and Bloom filters as described in Section 1.

Recently, Durham (2012) proposed a three-party framework for PPRL using Bloom filters. In her work, she suggested record level Bloom filter encoding to overcome the problem of cryptanalysis associated with field (or attribute) level encoding (Kuzu et al. 2011), and she used locality-sensitive hash functions for private blocking to reduce the computational complexity. Empirical studies conducted on real datasets show that this approach outperforms existing Bloom filter based approaches.

3 Protocol Description

Two database owners, *Alice* and *Bob*, with databases \mathbf{D}^A and \mathbf{D}^B , participate in the protocol. We divide the steps of our protocol into three main phases, which are the preparation phase, the length filtering phase, and the iterative classification phase. The notation we use is summarized in Table 1. Figures 2 to 7 illustrate the steps of the protocol.

3.1 Preparation Phase

In the initial preparation phase the database owners prepare their data to be used in the iterative protocol. The steps of this phase are:

1. Alice and Bob agree upon a bit array length l ; k hashing functions $h_1 \dots h_k$; the length (in characters) of grams q ; the similarity measure $sim(b^A, b^B)$ to measure the similarity of two Bloom filters b^A and b^B ; a minimum similarity threshold value s_t , above which two records are classified as a match; the maximum number of bit positions they are willing to reveal to each other r_{max} ($r_{max} \leq l$); and a set of attributes A (linkage attributes) that are used to link the records.
2. Alice and Bob each stores the values of their linkage attributes in a list, \mathbf{S}^A and \mathbf{S}^B , respectively, for each of the records in their databases.
3. For every attribute string s in \mathbf{S}^A , Alice performs the following steps:
 - (a) Alice converts string s into a set of q -grams.
 - (b) Alice converts these q -gram sets into a Bloom filter b^A (of that record) of length l using the hash functions $h_1 \dots h_k$. All the attributes of a record are mapped to one single Bloom filter.
4. Alice also counts for each Bloom filter the number of bit positions that are set to 1 (1-bits), x^A , and stores this number along with the identifier of the record into its list \mathbf{O}^A , as is illustrated in Figure 2 for the example Bloom filters.
5. For every attribute string s in \mathbf{S}^B , Bob performs steps 3 and 4.

3.2 Length Filtering Phase

The second phase of our protocol aims to remove non-matching record pairs using a length filtering method on the Bloom filters. At the end of this phase, candidate record pairs are generated with their corresponding value for the minimum number of common 1-bits they require (c_{min}) to be classified as a match. We use the Dice-coefficient (Equation 1) as the similarity function $sim(\cdot, \cdot)$ to compare two Bloom filters, as it is insensitive to many zeros in Bloom filters. However, any q -gram based similarity function can be used (Christen 2012). Algorithm 1 shows the main steps involved in this phase.

Alice's Bloom Filters

RecID	Bloom Filters	Num 1s (x^A)
RA1	1 1 1 1 0 1 0 1 0 0	6
RA2	0 1 0 0 0 0 1 0 0 1	3
RA3	0 1 1 1 0 1 1 0 0 1	6
RA4	1 1 0 0 1 0 1 1 0 0	5

Bob's Bloom Filters

RecID	Bloom Filters	Num 1s (x^B)
RB1	1 1 1 1 0 1 0 1 1 0	7
RB2	1 1 1 1 0 1 1 0 0 1	7
RB3	1 1 0 1 1 0 1 0 0 0	5

Figure 2: Example Bloom filters held by Alice and Bob for the records in their databases (\mathbf{D}^A) and (\mathbf{D}^B), respectively, and the number of 1-bits in each of the Bloom filters.

Record Pairs

A	B	x^A	x^B	Length Filter
RA1	RB1	6	7	$(6-7 \leq 6/2)$ Yes
RA1	RB2	6	7	$(6-7 \leq 6/2)$ Yes
RA1	RB3	6	5	$(6-5 \leq 6/2)$ Yes
RA2	RB1	3	7	$(3-7 \leq 3/2)$ No
RA2	RB2	3	7	$(3-7 \leq 3/2)$ No
RA2	RB3	3	5	$(3-5 \leq 3/2)$ No
RA3	RB1	6	7	$(6-7 \leq 6/2)$ Yes
RA3	RB2	6	7	$(6-7 \leq 6/2)$ Yes
RA3	RB3	6	5	$(6-5 \leq 5/2)$ Yes
RA4	RB1	5	7	$(5-7 \leq 5/2)$ Yes
RA4	RB2	5	7	$(5-7 \leq 5/2)$ Yes
RA4	RB3	5	5	$(5-5 \leq 5/2)$ Yes

Candidate Record Pairs

A	B	x^A	x^B	c_{min}
RA1	RB1	6	7	6
RA1	RB2	6	7	6
RA1	RB3	6	5	5
RA3	RB1	6	7	6
RA3	RB2	6	7	6
RA3	RB3	6	5	5
RA4	RB1	5	7	5
RA4	RB2	5	7	5
RA4	RB3	5	5	4

Figure 3: Pruning record pairs that are non-matches (length filtering) according to the number of 1-bits, x^A and x^B , using Equation 2 (left), and candidate record pairs after the length filtering phase, with the minimum number of common 1-bits required to be classified as a match, c_{min} , according to the values of x^A and x^B , calculated using Equation 3 (right). s_t is set to 0.8. The minimum value of all c_{min} , $\min(c_{min})$, is 4 which will be used as the value for r_1 in the first iteration ($i = 1$).**Algorithm 1:** Length Filtering**Input:**

- \mathbf{O}^A : List of record IDs and num of 1-bits (r^A, x^A) from Alice
- \mathbf{O}^B : List of record IDs and num of 1-bits (r^B, x^B) from Bob
- Minimum similarity threshold s_t

Output:

- List of candidate record pairs with their minimum number of common 1-bits required (c_{min}): \mathbf{C}

```

1:  $\mathbf{C} = []$ 
2: for  $(r_i^A, x_i^A) \in \mathbf{O}^A$  do
3:   for  $(r_i^B, x_i^B) \in \mathbf{O}^B$  do
4:      $x_{min} = \min(x_i^A, x_i^B)$ 
5:      $d = |x_i^A - x_i^B|$ 
6:      $d_{max} = \frac{2x_{min}(1-s_t)}{s_t}$ 
7:     if  $d \leq d_{max}$  then
8:        $c_{min} = \lfloor \frac{s_t(x_i^A + x_i^B)}{2} \rfloor$ 
9:       Append  $([r_i^A, x_i^A], [r_i^B, x_i^B], c_{min})$  to  $\mathbf{C}$ 

```

1. Alice and Bob exchange the number of 1-bits in each of their Bloom filters along with their record identifiers or randomly generated unique ID numbers (lists \mathbf{O}^A and \mathbf{O}^B , respectively). They then generate all the record pairs ($|\mathbf{D}^A| \times |\mathbf{D}^B|$) if no blocking function is applied, see Section 4 for how this can be improved) along with the number of 1-bits as is illustrated in Figure 3.
2. The difference between the number of 1-bits in two Bloom filters $d = |x^A - x^B|$, should be less than the maximum bit difference d_{max} , in order to consider the pair as a possible match. Assume $x^A \leq x^B$ and all the bit positions set to 1 in b^A are also set to 1 in b^B ($c = x^A$). This assumption gives the lower bound of the similarity coefficient and the upper bound of bit difference d_{max} . The value for d_{max} can be calculated given the minimum similarity coefficient threshold s_t and number of 1-bits in the Bloom filters, x^A and x^B , as shown in Equation 2.

All the pairs that have a larger 1-bit difference than d_{max} can be removed without proceeding

further since they cannot be matches.

$$\begin{aligned}
sim(b^A, b^B) &= \frac{2c}{x^A + x^B} \geq s_t \\
\frac{2 \min(x^A, x^B)}{\min(x^A, x^B) + (\min(x^A, x^B) + d)} &\geq s_t \\
\frac{2x^A}{x^A + x^A + d} &\geq s_t \\
d &\leq \frac{2x^A(1-s_t)}{s_t} \\
d_{max} &= \frac{2x^A(1-s_t)}{s_t}. \quad (2)
\end{aligned}$$

In order to classify a record pair as a match (similarity value above the threshold value s_t), the record pair must have less than or equal to d_{max} number of differences between 1-bits in their Bloom filters. Alice and Bob store only the record pairs that have $|x^A - x^B| \leq d_{max}$, as is illustrated in Figure 3.

For example, if s_t is set to 0.8, then the difference between 1-bits in two Bloom filters must be at maximum half the value of the smaller value for the 1-bits in the two Bloom filters ($0.5 \times \min(x^A, x^B)$) in order to be classified as a match, following $sim(b^A, b^B) \geq 0.8 \Rightarrow \frac{2c}{x_1^A + x_1^B} \geq$

$$\frac{8}{10} \Rightarrow \frac{2x_1^A}{x_1^A + (x_1^A + d)} \geq \frac{8}{10} \Rightarrow d \leq 0.5x_1^A.$$

3. Alice and Bob now calculate the minimum number of common 1-bits required for a record pair to be classified as a match, c_{min} , for each pair of the remaining candidate records, as is illustrated in Figure 3. This is calculated for each pair using the values for x^A , x^B and s_t as shown in Equation 3, where $\lfloor \cdot \rfloor$ denotes the rounding to the next lowest integer value. The resulting candidate record pairs with the values for x^A , x^B , and c_{min} are stored in the Candidates Index data structure, \mathbf{C} , which will be used as an input to

the next phase of the protocol, the iterative classification phase.

$$\begin{aligned} \text{sim}(b^A, b^B) &= \frac{2c}{x^A + x^B} \geq s_t \\ \frac{2c_{\min}}{x^A + x^B} &= s_t \\ c_{\min} &= \left\lfloor \frac{s_t(x^A + x^B)}{2} \right\rfloor \end{aligned} \quad (3)$$

3.3 Iterative Classification Phase

The main task of a record linkage process is the classification of record pairs (Christen 2012). The iterative classification phase is where we classify record pairs into matches, non-matches, and possible matches. This classification needs to be done in such a way that no information about the values that were mapped into Bloom filters is being revealed to the two database owners.

Alice and Bob are prepared to reveal $(l - r_{\max})$ bit positions to each other in an iterative way without compromising the sensitive values in their Bloom filters. The number of bits to be revealed in each iteration, r_i , is a crucial parameter to be set as it provides a trade-off between privacy and computational efficiency of the protocol. There are two possible extreme cases.

1. Revealing all the $(l - r_{\max})$ bits in one iteration, which is very fast but is not secure since the bit positions are revealed for all the Bloom filter pairs including non-matches as well.
2. Revealing the $(l - r_{\max})$ bits in $(l - r_{\max})$ iterations where only 1 bit position is revealed in each iteration. This would be the best case for preserving privacy as it removes the non-matches in an iterative way before revealing the rest of the bit positions. This approach is however not scalable to large databases, especially with long Bloom filters, as each iteration requires communication between the database owners.

Hence, a method to reveal an optimal number of bits, r_i , in each iteration is required. We propose a method to calculate this optimal number by finding the smallest value of the minimum number of additional common 1-bits required to classify a pair as a match in each iteration among all the record pairs. The record pair that requires the smallest number of additional common 1-bits among all the other pairs has a security risk if more bit positions are revealed than the minimum number of common 1-bits it requires.

Assume c_i is the total number of common 1-bits revealed so far up to iteration i . The value for $\min(c_{\min} - c_{i-1})$ ($i > 0$) is calculated to be used as the value for r_i in the i^{th} iteration. For example, in the first iteration ($i = 1$), $\min(c_{\min})$ ($c_0 = 0$) will be used as the value for the number of bit positions to be revealed, r_1 . After r_1 bit positions are revealed in the first iteration, the value for $(c_{\min} - c_1)$ will be calculated for each of the remaining record pairs to calculate the value for $r_2 = \min(c_{\min} - c_1)$ in the second iteration, and then $\min(c_{\min} - c_2)$ will be used as the value for r_3 in the third iteration, and so on.

The iterative classification phase is done as follows (Algorithm 2 provides an overview of these steps):

1. Among all the $(c_{\min} - c_{i-1})$ values for all the unclassified pairs of records, the minimum value,

Algorithm 2: Iterative Classification Phase

Input:

- **C**: Candidate record pairs from length filtering phase

Output:

- **M**: Set of record pairs classified as matches

- **N**: Set of record pairs classified as non-matches

- **P**: Set of record pairs classified as possible matches

```

1: M = [], N = [], P = C
2: while P ≠ [] do
3:    $i = 1, t = 0$ 
4:   while  $r \leq r_{\max}$  do
5:      $r_i = \min(c_{\min} - c_{i-1})$ 
6:      $t = t + r_i$ 
7:     for  $(b^A, b^B) \in P$  do
8:        $x^A = \text{num\_1-bits\_in\_}b^A$ 
9:        $x^B = \text{num\_1-bits\_in\_}b^B$ 
10:       $c_{\min} = \text{num\_common\_1-bits\_in\_}b^A \text{ and } b^B$ 
11:       $\text{reveal\_bits}(r)$ 
12:       $x_i^A = \text{total\_num\_1-bits\_revealed\_in\_}b^A$ 
13:       $x_i^B = \text{total\_num\_1-bits\_revealed\_in\_}b^B$ 
14:       $c_i = \text{total\_num\_common\_1-bits\_revealed\_in\_}b^A \text{ and } b^B$ 
15:      if  $c_i \geq c_{\min}$  then // Case C1
16:        Append  $(b^A, b^B)$  to M
17:        Delete  $(b^A, b^B)$  from P
18:      else if  $c_i < c_{\min}$  and  $(c_{\min} - c_i) > (l - t)$  then // C2
19:        Append  $(b^A, b^B)$  to N
20:        Delete  $(b^A, b^B)$  from P
21:      else if  $c_i < c_{\min}$  and  $(c_{\min} - c_i) \leq (l - t)$  then // C3
22:        if  $((x^A - x_i^A) < (c_{\min} - c_i))$  or
23:           $((x^B - x_i^B) < (c_{\min} - c_i))$  then // C4
24:          Append  $(b^A, b^B)$  to N
25:          Delete  $(b^A, b^B)$  from P
26:       $i = i + 1$ 
27:   for  $(b^A, b^B) \in P$  do
28:     Do_rehash()
    
```

$\min(c_{\min} - c_{i-1})$, is taken as the lower bound of the number of bits to be revealed in the next iteration. Alice and Bob both will exchange $r_i = \min(c_{\min} - c_{i-1})$ same bit positions from each of their Bloom filters. For example, if $r_1 = \min(c_{\min} - c_0) = \min(c_{\min}) = 4$, then the first 4 bit positions are exchanged in the first iteration, as shown in Figure 4. The total number of bit positions revealed so far up to an iteration i is $t_i = \sum_i r_i$.

From the exchange of t_i bit positions, three possible cases can occur with each record pair.

- Case 1 (C1 in Algorithm 2): Record pairs which have c_{\min} or more than c_{\min} out of t_i bit positions in both Bloom filters (b^A and b^B) set to 1 ($c_i \geq c_{\min}$). These pairs are classified as matches.
- Case 2 (C2 in Algorithm 2): Record pairs which have some or none of the t_i bit positions set to 1 in both Bloom filters b^A and b^B ($c_i < c_{\min}$) and the number of additional common 1-bits required ($c_{\min} - c_i$) is greater than the number of remaining unrevealed bit positions ($c_i < c_{\min}$ and $(c_{\min} - c_i) > (l - t_i)$). These pairs are classified as non-matches.
- Case 3 (C3 in Algorithm 2): Record pairs which have some or none of the t_i bit positions set to 1 in both Bloom filters b^A and b^B ($c_i < c_{\min}$) and the number of additional common 1-bits required ($c_{\min} - c_i$) is less than or equal to the number of remaining unrevealed bit positions ($c_i < c_{\min}$ and $(c_{\min} - c_i) \leq (l - t_i)$). These record pairs are classified as possible matches.

Candidate Record Pairs – Iteration 1



















A	B	c_{\min}	Alice's BF	Bob's BF	$c_{\min} - c_1$	$x^A - x_1^A$	$x^B - x_1^B$	Class
RA1(6)	RB1(7)	6			2	RA1(2)	RB1(3)	Pos Match
RA1(6)	RB2(7)	6			2	RA1(2)	RB2(3)	Pos Match
RA1(6)	RB3(5)	5			2	RA1(2)	RB3(2)	Pos Match
RA3(6)	RB1(7)	6			3	RA3(3)	RB1(3)	Pos Match
RA3(6)	RB2(7)	6			3	RA3(3)	RB2(3)	Pos Match
RA3(6)	RB3(5)	5			3	RA3(3)	RB3(2)	Non Match
RA4(5)	RB1(7)	5			3	RA4(3)	RB1(3)	Pos Match
RA4(5)	RB2(7)	5			3	RA4(3)	RB2(3)	Pos Match
RA4(5)	RB3(5)	4			2	RA4(3)	RB3(2)	Pos Match

Figure 4: Bloom Filters of Alice and Bob with $t_1 = 4$ ($r_1 = \min(c_{\min}) = 4$) bits revealed after the first iteration. The calculated values for c_1 are used to calculate the value for r_2 for the next iteration, $r_2 = \min(c_{\min} - c_1) = 2$.

Candidate Record Pairs – Iteration 2

A	B	$c_{\min} - c_1$	Alice's BF	Bob's BF	$c_{\min} - c_2$	$x^A - x_i^A$	$x^B - x_i^B$	Class
RA1(2)	RB1(3)	2			1	RA1(1)	RB1(2)	Pos Match
RA1(2)	RB2(3)	2			1	RA1(1)	RB2(2)	Pos Match
RA1(2)	RB3(2)	2			2	RA1(1)	RB2(1)	Non Match
RA3(3)	RB1(3)	3			2	RA3(2)	RB1(2)	Pos Match
RA3(3)	RB2(3)	3			2	RA3(2)	RB2(2)	Pos Match
RA4(3)	RB1(3)	3			3	RA4(2)	RB1(2)	Non Match
RA4(3)	RB2(3)	3			3	RA4(2)	RB2(2)	Non Match
RA4(3)	RB3(2)	2			1	RA4(2)	RB3(1)	Pos Match

Figure 5: Bloom Filters of Alice and Bob with $t_2 = 6$ ($r_2 = 2$) bits revealed after the second iteration. The calculated values for c_2 are used to calculate the value for r_3 for the next iteration, $r_3 = \min(c_{\min} - c_2) = 1$.

Candidate Record Pairs – Iteration 3

A	B	$c_{\min} - c_2$	Alice's BF	Bob's BF	$c_{\min} - c_3$	$x^L - x_3^L$	$x^R - x_3^R$	Class																						
RA1(1)	RB1(2)	1	<table><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>×</td><td>×</td><td>×</td><td>×</td></tr></table>	1	1	1	1	0	1	0	×	×	×	×	<table><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>×</td><td>×</td><td>×</td><td>×</td></tr></table>	1	1	1	1	0	1	0	×	×	×	×	1	RA1(1)	RB1(2)	Pos Match
1	1	1	1	0	1	0	×	×	×	×																				
1	1	1	1	0	1	0	×	×	×	×																				
RA1(1)	RB2(2)	1	<table><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>×</td><td>×</td><td>×</td><td>×</td></tr></table>	1	1	1	1	0	1	0	×	×	×	×	<table><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>×</td><td>×</td><td>×</td><td>×</td></tr></table>	1	1	1	1	0	1	1	×	×	×	×	1	RA1(1)	RB2(1)	Pos Match
1	1	1	1	0	1	0	×	×	×	×																				
1	1	1	1	0	1	1	×	×	×	×																				
RA3(2)	RB1(2)	2	<table><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>×</td><td>×</td><td>×</td><td>×</td></tr></table>	0	1	1	1	0	1	1	×	×	×	×	<table><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>×</td><td>×</td><td>×</td><td>×</td></tr></table>	1	1	1	1	0	1	0	×	×	×	×	2	RA3(1)	RB1(2)	Pos Match
0	1	1	1	0	1	1	×	×	×	×																				
1	1	1	1	0	1	0	×	×	×	×																				
RA3(2)	RB2(2)	2	<table><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>×</td><td>×</td><td>×</td><td>×</td></tr></table>	0	1	1	1	0	1	1	×	×	×	×	<table><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>×</td><td>×</td><td>×</td><td>×</td></tr></table>	1	1	1	1	0	1	1	×	×	×	×	1	RA3(1)	RB2(1)	Pos Match
0	1	1	1	0	1	1	×	×	×	×																				
1	1	1	1	0	1	1	×	×	×	×																				
RA4(2)	RB3(1)	1	<table><tr><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>×</td><td>×</td><td>×</td><td>×</td></tr></table>	1	1	0	0	1	0	1	×	×	×	×	<table><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>×</td><td>×</td><td>×</td><td>×</td></tr></table>	1	1	0	1	1	0	1	×	×	×	×	0	RA4(1)	RB3(0)	Match
1	1	0	0	1	0	1	×	×	×	×																				
1	1	0	1	1	0	1	×	×	×	×																				

Figure 6: Bloom Filters of Alice and Bob with $t_3 = 7$ ($r_3 = 1$) bits revealed after the third iteration. The calculated values for c_3 are used to calculate the value for r_4 for the next iteration, $r_4 = \min(c_{\min} - c_3) = 1$.

Candidate Record Pairs – Iteration 4

A	B	$c_{\min} - c_3$	Alice's BF	Bob's BF	$c_{\min} - c_4$	$x^A - x^A_4$	$x^B - x^B_4$	Class
RA1(1)	RB1(2)	1	$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$	0	RA1(0)	RB1(1)	Match
RA1(1)	RB2(1)	1	$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$	0	RA1(0)	RB2(1)	Non Match
RA3(1)	RB1(2)	2	$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$	2	RA3(1)	RB1(1)	Non Match
RA3(1)	RB2(1)	1	$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$	1	RA3(1)	RB2(1)	Pos Match

Figure 7: Bloom Filters of Alice and Bob with $t_4 = 8$ ($r_4 = 1$) bits revealed after the fourth iteration. The pair that is still classified as possible match will need to be re-processed with different hash functions.

2. After having t_i bit positions revealed in iteration i , all the pairs that are classified as matches and non-matches (cases C1 and C2) can be removed from the set of candidate record pairs \mathbf{C} . Only pairs that are classified as possible matches (case C3) will be taken to the next iteration.

Based on the revealed bit positions, Alice and Bob calculate the new values for c_i , x_i^A , and x_i^B . Moreover, the values for x_i^A and x_i^B can also be used to prune more non-matches from the pairs of records that were classified as possible matches. Record pairs which have $(c_{min} - c_i) < (x^A - x_i^A)$ or $(c_{min} - c_i) < (x^B - x_i^B)$ can be classified as non-matches and pruned (case C4 in Algorithm 2). For example, if 2 more 1-bits are left unrevealed in b^B ($x^B - x_i^B = 2$) after revealing 4 bit positions in the first iteration, and

($c_{min} - c_i$) is 3 which means at least 3 more common 1-bits are required for the record pair to be classified as a match from only 2 1-bits in b^B (which is impossible), then this record pair can be removed at this iteration without taking into the next iteration and revealing more bits for this non-matching pair. Record pair RA3 and RB3 in Figure 4 is such a case.

3. For the pairs that are classified as possible matches (case 3), Alice and Bob repeat the steps until r_{max} bit positions are exchanged in an iterative method (r_{max} is set to 8 in our example).
4. The record pairs that are still classified as possible matches in the last step, after r_{max} bit positions have been revealed, need to be re-hashed into new Bloom filters with different hash functions k (lines 26 and 27 in Algorithm 2).

3.4 Computational Complexity

Assuming the number of candidate record pairs is n , the average number of q -grams in each record is Q , the number of hash functions used to map q -grams into a Bloom filter is k , and the length of Bloom filters is l , then the computation cost of this protocol is $O(n*Q*k)$ hash operations and $O((n*l)^2)$ bit comparisons, while the communication cost is $O(n*l)$. The communication complexity of this protocol is therefore linear in the size of the databases.

4 Improving Efficiency

In the length filtering phase, we remove record pairs that have a difference between the number of 1-bits larger than a certain value, depending on the minimum similarity threshold value s_t before starting the iterations as explained in Section 3.2. This reduces the number of candidate record pairs to be processed in the iterative classification phase.

Indexing techniques (Christen 2011) can be applied before performing the linkage based on phonetic encodings (Christen 2012) such that similar records are grouped together. This further reduces the number of candidate record pairs, because only the record pairs that are in the same blocks will be considered as candidate record pairs. Alice and Bob each independently applies an indexing function to their databases and groups records that have the same blocking key value (Christen 2012). The indexing function, for example, can be applied on another set of quasi-identifier attributes or part of the linkage attributes. Secure set intersection protocols (Agrawal et al. 2003, Kissner & Song 2004) can be used to securely identify the list of common blocks in both databases (Vatsalan et al. 2011).

Locality sensitive hashing (LSH) can also be applied to reduce the number of candidate record pairs (Gionis et al. 1999). The LSH method originally addresses the approximate nearest neighbor problem by hashing values such that similar records are put into the same buckets with high probability. Secure set intersection or binning (Vatsalan et al. 2011) can then be used to find the list of common blocks in both databases.

The iterative pruning of candidate record pairs using Bloom filters allows removing pairs that have higher probability of being non-matches before exchanging more bit positions. The aim of our iterative method is to prune the record pairs that are classified as non-matches and matches and thereby reduce the number of pairs of possible matches in each iteration as much as possible. We proposed to reveal $\min(c_{min} - c_{i-1})$ number of bits in each iteration. Experiments conducted on a real-world database (see Section 6) show that though many bits are being revealed in the first few iterations, only a very few bits are being revealed in the later iterations which takes many iterations to run and thus makes the process not scalable to large databases.

To overcome this problem, we propose a method for revealing more bits when the number of bits to be revealed becomes very small, without compromising privacy. Assume r_i bits have been revealed in iteration i , among which c_i number of common 1-bits have been found in a record pair which needs $c_{min} - c_i$ more common 1-bits in both Bloom filters in order to classify the pair as a match. If $c_{min} - c_i$ is very small and we still can classify the pair as a match even if no more common 1-bits are found in the later iterations, then this pair will not be at a security risk if more bits

are revealed in the next iteration, because it has already been considered as a match. The question now arises what is the maximum value for $c_{non} = c_{min} - c_i$ that can be ignored to classify the pairs as matches without accuracy loss. We introduce another similarity threshold value, s_r , to calculate the value for the minimum number of bits that can be revealed for each pair in an iteration, r_{min} , as shown in Equation 4. This basically expands the calculation of value r_i in step 5 of Algorithm 2 as below. Among the values for c_{non} for all the pairs, the smallest value is taken to be used as the value for the minimum number of bits that can be revealed in all the pairs of Bloom filters in an iteration, $r_{min} = \min(c_{non})$.

$$\begin{aligned} s_t - s_r &= \frac{2(c_{non})}{x^A + x^B} \\ r_{min} &= \min(c_{non}) \\ r_i &= \min(r_i, r_{min}) \end{aligned} \quad (4)$$

If r_i becomes less than r_{min} in an iteration, especially in later iterations, then r_{min} bits will be revealed. It is important to note that the similarity threshold to reveal, s_r , is only used to calculate the value for r_{min} while the similarity threshold s_t is used to classify the pairs. This approach reduces the complexity of the protocol significantly without compromising the privacy of the non-matched record pairs. This is empirically evaluated in Section 6.

5 Privacy Analysis

The amount of privacy provided by this protocol depends on the number of hash functions used (k) and the length of the Bloom filter (l) (Schnell et al. 2009, Kuzu et al. 2011). The values for k and l have to be carefully chosen as these values provide a trade-off between accuracy of the classification and privacy. The higher the value for k/l , the higher the privacy and the lower the accuracy, because the number of q -grams mapped to one single bit increases, which results in less accurate linkage results but makes it harder for an attacker to infer the possible combinations.

Assume the minimum number of bits required to perform a dictionary attack using an external database to infer a bit pattern is t_a . The privacy characteristics provided by our protocol are:

1. More bits are revealed for pairs that are more likely to be matches (Figure 16).
2. Non-matching record pairs are removed in the earlier iterations when only a small number of bits have been revealed ($t_i < t_a$), which therefore cannot be used to infer records using a dictionary attack (Figures 13 and 16).
3. When a sufficient number of bits t_a are revealed for a dictionary attack (iteration i), the remaining unclassified pairs have a minimum similarity that is close enough ($\text{sim}_{min}(\mathbf{C}_i) \approx s_t$) to be considered as matches (Figures 13 and 14).

Pruning candidate record pairs that have higher probability of being classified as non-matches at early iterations improves the privacy of the protocol, since the non-matches are removed without revealing more bits in the next iterations. We evaluate the probability of a dictionary attack when different percentage of bits are revealed (see Section 6). We expect the probability to increase with the percentage of bits

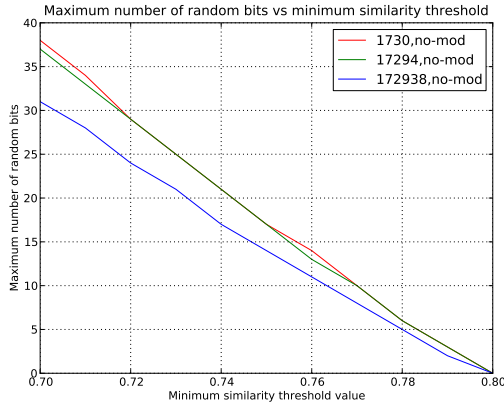


Figure 8: Maximum number of random bits z_{max} that can be added against the minimum lower bound s_l of the similarity threshold value, with $s_t = 0.8$.

revealed, as the more bits revealed the smaller the number of possible attribute values one could infer, having access to an external data source (such as a large telephone directory) containing most possible attribute values, which results in a reduced security of the protocol. Hence, only the pairs that have a higher possibility of becoming matches are having a higher probability of a dictionary attack when the percentage of revealed bits increases. Since we hash-map several attribute values from each record into one combined Bloom filter, it is harder for an attacker to infer individual attribute values that correspond to a revealed bit pattern (Durham 2012).

The security parameter which is the maximum number of bits to be revealed in the Bloom filters r_{max} is agreed upon by the two database owners. This determines the privacy of the protocol. A larger value of r_{max} results in less privacy but more record pairs being classified, while a smaller value allows only a smaller number of pairs being classified with higher privacy.

Depending on the data and the distribution of 1-bit patterns, another security issue to be considered with our protocol is that revealing some bits (that have comparatively high sensitive information due to a small number of q -grams that are mapped to those bits) are susceptible to dictionary attacks. We propose two methods for overcoming the problem of revealing the rare bits in Bloom filters that can be attacked with higher probability.

1. **Adding random bits:** Random bits can be added to Bloom filters individually by the database owners in the preparation phase in order to perturb the dataset. The question is how many random bits need to be added to increase the security without compromising accuracy and complexity. When adding random bits three cases can occur. One is when the bits added by the two database owners lead to the same number of additional matching 1-bits at the same positions (common 1-bits) which results in almost the same similarity value. The second case is where some of the added bits are matching and thus the number of additional common 1-bits introduced by the addition of random bits is less than the number of random bits added by the database owners. The third case occurs where the added bits do not match with any bit positions and thus no additional common 1-bits are introduced by adding random bits.

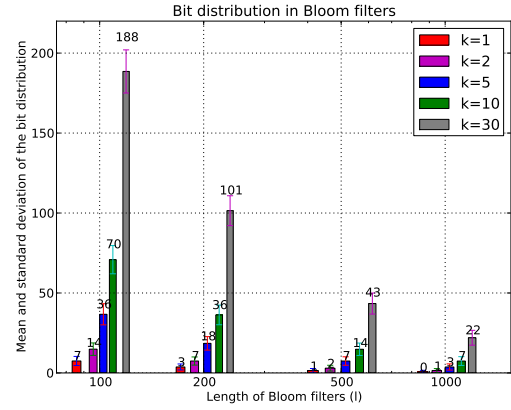


Figure 9: Bit distribution in Bloom filters. Numbers shown are mean values and the error bars are standard deviations.

In both the second and third cases, the new similarity value decreases because of the random bits. However, the third case is the worst case and needs to be considered in determining the similarity threshold value. The database owners must agree on a minimum acceptable lower bound of the similarity threshold, s_l , with $s_l < s_t$. If the values for x_{min}^A , x_{min}^B , s_t , and s_l are known, then the maximum number of random bits that can be added by the database owners, z_{max} , can be estimated using Equation 5.

$$\begin{aligned}
 s_t &= \frac{2 \times c_{min}}{x_{min}^A + x_{min}^B} \\
 s_l &= \frac{2 \times c_{min}}{(x_{min}^A + z_{max}) + (x_{min}^B + z_{max})} \\
 s_l &= \frac{s_t \times (x_{min}^A + x_{min}^B)}{(x_{min}^A + z_{max}) + (x_{min}^B + z_{max})} \\
 z_{max} &= \frac{(s_t - s_l) \times (x_{min}^A + x_{min}^B)}{2 \times s_l} \quad (5)
 \end{aligned}$$

Figure 8 shows the maximum number of random bits (z_{max}) that can be individually added to each Bloom filter by the database owners to perturb the bit distribution in Bloom filters against the minimum similarity threshold value that is acceptable without much accuracy loss in the classification results. The maximum number of random bits linearly increases when the minimum similarity threshold decreases.

2. **Simulation attack:** The database owners can individually simulate the protocol and attack their own databases before exchanging the values in order to identify if there exist any bits that map only to a small number of q -grams. Based on that, they can either change the values for k , l , and q , or they can agree on an appropriate value for the security parameter r_{max} . The bit distribution in Bloom filters in a real-world Australian online telephone database with 17,294 records shows that an average of 22 q -grams and a minimum of 14 q -grams are mapped to one single bit when $k = 30$, $q = 2$ and $l = 1000$ (as shown in Figure 9). As can be seen from this figure, the number of q -grams mapped to one bit decreases with l while increasing with k .

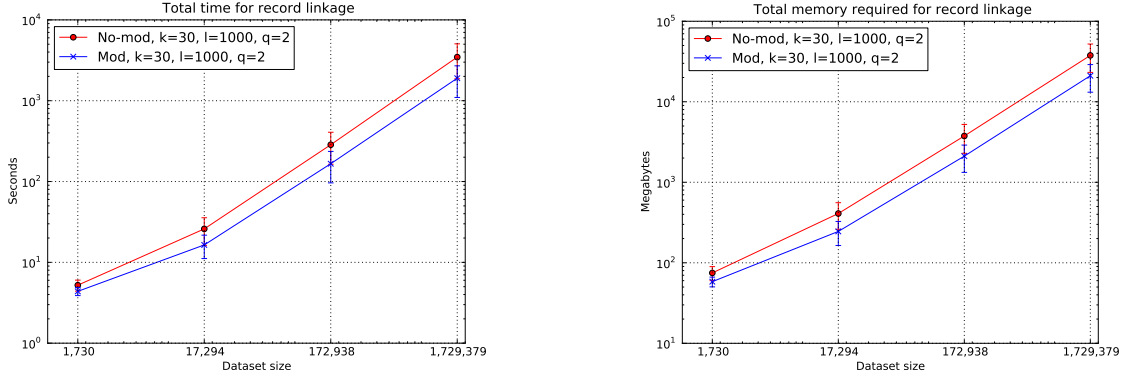


Figure 10: Total run time (left plot) and memory usage (right plot) of the linkage averaged over the results of both database owners over all variations of each dataset. The error bars shown are the standard deviations.

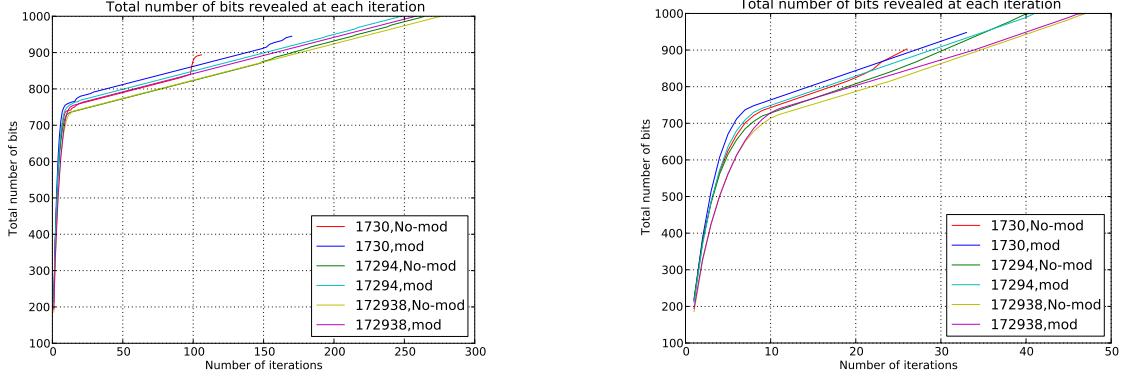


Figure 11: Total number of bits revealed at each iteration without s_r (left plot) and after introducing s_r (right plot). The values for the similarity thresholds were set as $s_t = 0.8$ and $s_r = 0.77$.

Table 2: The number of records in the datasets used for experiments, and the number of records that overlap (i.e. occur in both datasets of a pair). This is considered as the number of true matches.

Dataset sizes	25% overlap	50% overlap	75% overlap
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	864,487	1,297,029

6 Experimental Evaluation

We conducted experiments using a real Australian telephone directory 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). To generate datasets of different sizes, we sampled 0.1%, 1%, 10% and 100% of records in the full database twice each, and stored them into a pair of files such that 25%, 50% or 75% of records appeared in both files of a pair. Table 2 provides an overview of the datasets generated.

The record pairs that occur in both datasets are exact matches (these datasets are labelled as ‘No-mod’ in the results figures). To investigate the performance of our protocol in the context of ‘dirty data’ (where attribute values contain errors and variations), we generated another series of datasets (labelled as ‘Mod’) where we modified each attribute value by applying two randomly selected character edit op-

erations (insert, delete, substitute or transposition). This leads to a much reduced number of exact matching record pairs and allows us to evaluate the accuracy of approximate matching of our protocol.

Following previous work (Schnell et al. 2009), we set the values for the Bloom filter parameters as $l = 1000$, $k = 30$, and $q = 2$. The minimum similarity threshold to classify was set to $s_t = 0.8$ and threshold to reveal bits was set to $s_r = 0.77$. All four attributes were used as the linkage attributes.

We prototyped the protocol using the Python programming language (version 2.7.1). We also implemented an attacker program to evaluate the privacy characteristics of this protocol. We used the attribute values in the full Australian telephone directory as the attacker’s reference set of values, and we calculated the probability of a dictionary attack as the number of unique possible values that can be inferred with the bits revealed for every bit pattern in a dataset.

All tests were run on an otherwise idle computer with a 64 bit Intel Xeon (2.4 GHz), 128 GBytes of main memory and running Ubuntu 11.04. The prototype and test datasets are available from the authors.

6.1 Scalability

Figure 10 shows the scalability of our protocol. Computation complexity is assessed as the total run time and memory usage required for the linkage. All variations of the datasets were used. The results of both the exact and the approximate matching (‘No-mod’ and ‘Mod’) are shown in the figures. The result figures exhibit a linear complexity trend in the size of the databases which makes the protocol scalable to large databases.

As discussed in Section 4, the number of bits revealed in the later iterations is very small and therefore it takes more iterations to classify the record pairs (see the left plot in Figure 11). With the proposed method of using a second similarity threshold, $s_r = 0.77$, this has been significantly improved, as shown in the right plot in Figure 11. The total number of iterations required to classify all the record pairs is reduced 6-fold (from 300 to 50 iterations for the largest dataset) with the proposed approach using a second threshold $s_r = 0.77$.

The reduction ratio (Christen 2012) of record pairs with unknown match status after classifying record pairs as ‘matches’ and ‘non-matches’ at each iteration is shown in Figure 12. As can be seen from the figure, the protocol shows a high increment rate in the reduction ratio after the first few iterations.

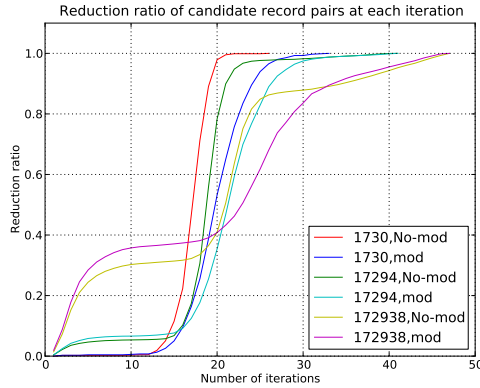


Figure 12: Reduction ratio (Christen 2012) of possible matches at each iteration, using different dataset sizes.

6.2 Privacy

The privacy characteristics of this protocol (as discussed in Section 5) are empirically evaluated in this section assuming that an adversary has access to an external database. The empirical evaluation of probability of a dictionary attack on a dataset consisting of 17,294 records using a public Australian telephone directory as an external database is shown in Figure 13.

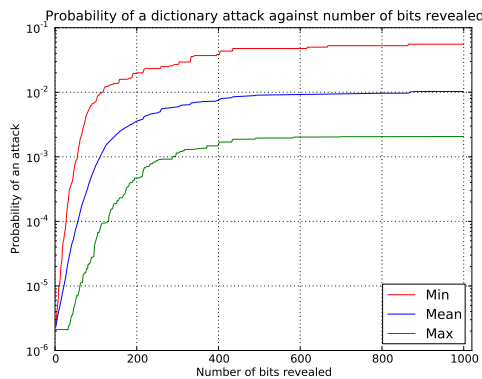


Figure 13: Probability of a dictionary attack from bits revealed (dataset with 17,294 records, reference dataset Australian telephone directory).

This study practically validates that the probability of a dictionary attack increases with the number of bits revealed, and the minimum probability of an attack becomes greater than 0.05 (i.e. the number

of values that can be inferred becomes less than 20) only after 800 bits being revealed. When 800 bits are revealed, most of the non-matching record pairs have already been removed (as can be seen from Figure 16), and the minimum similarity value of the remaining record pairs is nearly 0.7 (illustrated in Figure 14), which assures that the privacy of non-matches with similarity below 0.6 is not compromised with this iterative pruning approach.

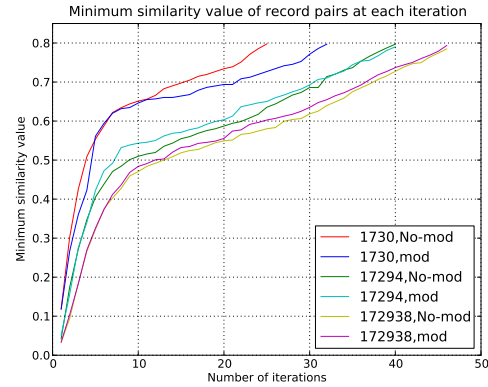


Figure 14: Minimum similarity value of unclassified record pairs at each iteration.

The results of this empirical evaluation of a dictionary attack validates that the privacy of the records corresponding to non-matches is not compromised by this iterative classification approach. The value for security parameter r_{max} can be set after conducting such a simulation attack as described above. For example, in this setting r_{max} can be agreed upon by the database owners to be set as 800.

6.3 Linkage quality

As can be seen from Figure 16, many non-matches are being classified in the first few iterations and then matches are classified more towards the middle to last iterations. The overall reduction ratio (Christen 2012) of candidate record pairs is thus high (Figure 12), while the recall ratio of matches being classified is also high (Figure 15).

The recall ratio is almost 1.0 for the datasets with no modifications (‘No-mod’). It is higher (nearly 0.8) with modified datasets as well (a total of 8 edits per record that results in almost 50% modifications in the corresponding q -grams), which explains the aspect of fault-tolerance to data errors by performing approximate matching.

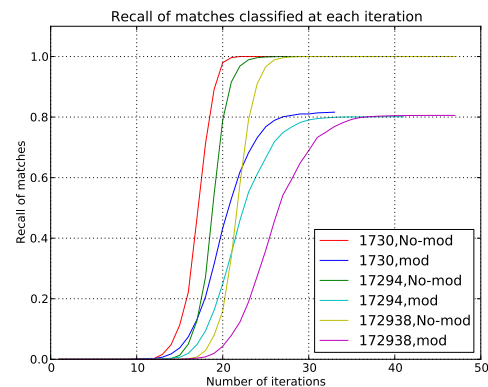


Figure 15: Recall of matches at each iteration, using different dataset sizes.

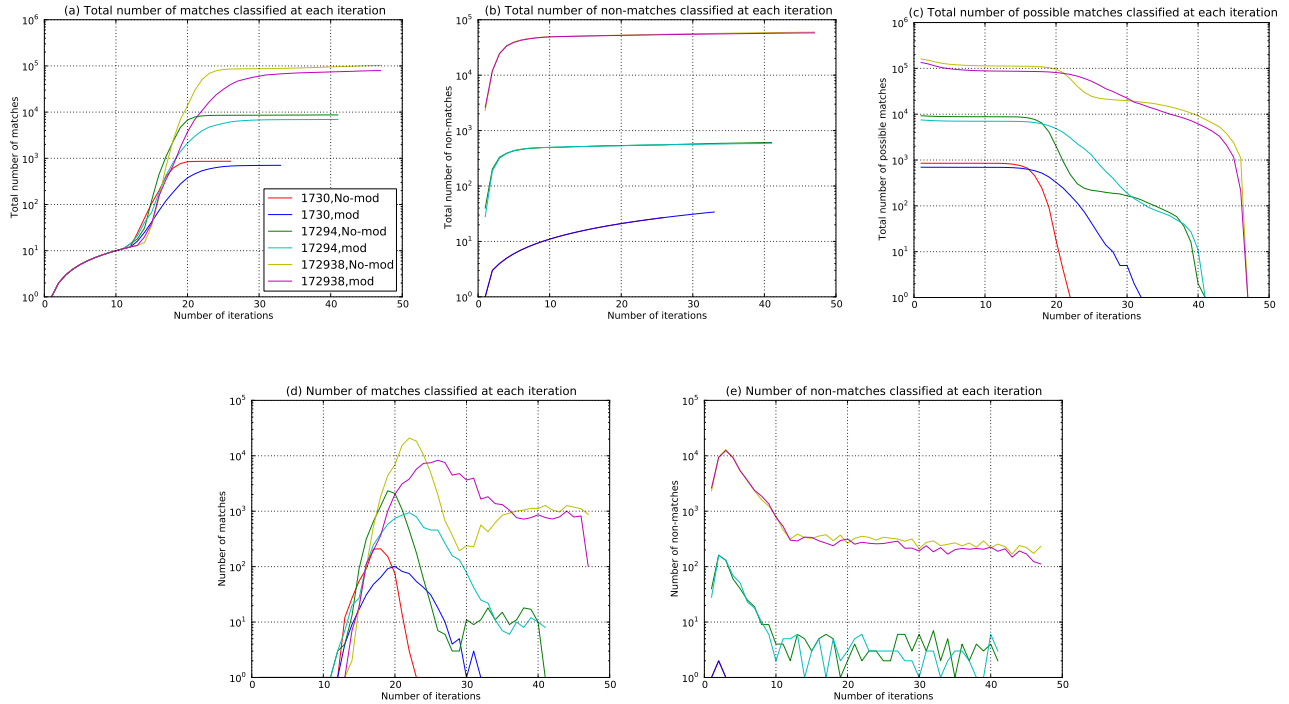


Figure 16: (a) Total number of record pairs classified as matches, (b) non-matches, (c) possible matches, and (d) number of record pairs classified as matches and (e) non-matches at each iteration.

The results of this empirical study validate that this parametric solution performs well by addressing the three main challenges of PPRL which are scalability, privacy, and linkage quality in the current parameter settings. The values for the Bloom filter related parameters l , k , and q play a major role in determining the balancing of these three factors as they provide a trade-off among the three factors.

7 Conclusion

In this paper we proposed a practical two-party protocol for privacy-preserving record linkage by addressing the three main challenges, which are scalability to large databases, high linkage quality results, and sufficient privacy characteristics. With the appropriate determination of values for the parameters, the experimental studies on a real-world database show that our proposed two-party PPRL protocol can perform efficient linkage with high linkage quality while providing adequate privacy characteristics.

In future work, we aim to find the best optimal values for the parameters by theoretically modelling the privacy, accuracy, and complexity of the protocol with different parameter values. Comparing our protocol with other two-party protocols in terms of the three factors is also an interesting research avenue. Another direction would be to study how effectively parallelism can be applied into this protocol.

Learning the values of the parameters such that all three main factors of PPRL are balanced will allow this protocol to be employed in real-world PPRL applications.

References

- Agrawal, R., Evfimievski, A. & Srikant, R. (2003), Information sharing across private databases, in 'ACM SIGMOD', San Diego, pp. 86–97.
- Al-Lawati, A., Lee, D. & McDaniel, P. (2005), Blocking-aware private record linkage, in 'Journal of Data and Information Quality', 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.
- Bloom, B. (1970), 'Space/time trade-offs in hash coding with allowable errors', *Communications of the ACM* **13**(7), 422–426.
- Broder, A., Mitzenmacher, M. & Mitzenmacher, A. (2002), Network applications of Bloom filters: A survey, in 'Internet Mathematics'.
- Christen, P. (2006), Privacy-preserving data linkage and geocoding: Current approaches and research directions, in 'IEEE ICDM Workshop on Privacy Aspects of Data Mining', Hong Kong.
- Christen, P. (2009), Geocode matching and privacy preservation, in 'Workshop on Privacy, Security, and Trust in KDD', Springer, pp. 7–24.
- Christen, P. (2011), 'A survey of indexing techniques for scalable record linkage and deduplication', *IEEE Transactions on Knowledge and Data Engineering*.
- Christen, P. (2012), *Data Matching*, Data-Centric Systems and Applications, Springer.
- Clifton, C., Kantarcioglu, M., Doan, A., Schadow, G., Vaidya, J., Elmagarmid, A. & Suci, D. (2004), Privacy-preserving data integration and sharing, in 'ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery', pp. 19–26.
- Clifton, C., Kantarcioglu, M., Vaidya, J., Lin, X. & Zhu, M. (2002), 'Tools for privacy preserving distributed data mining', *SIGKDD Explorations* **4**(2), 28–34.

- Durham, E. (2012), A framework for accurate, efficient private record linkage, PhD thesis, Vanderbilt University.
- Durham, E., Xue, Y., Kantarcioglu, M. & Malin, B. (2010), Private medical record linkage with approximate matching, in 'AMIA Annual Symposium Proceedings', p. 182.
- 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*.
- Dusserre, L., Quantin, C. & Bouzelat, H. (1995), 'A one way public key cryptosystem for the linkage of nominal files in epidemiological studies', *Medinfo* 8, 644-647.
- Freedman, M., Ishai, Y., Pinkas, B. & Reingold, O. (2005), 'Keyword search and oblivious pseudorandom functions', *Theory of Cryptography*.
- Gionis, A., Indyk, P. & Motwani, R. (1999), Similarity search in high dimensions via hashing, in 'Proceedings of the International Conference on Very Large Data Bases', pp. 518-529.
- Goldreich, O. (2004), *Foundations of cryptography: Basic applications*, Vol. 2, Cambridge Univ Press.
- Hall, R. & Fienberg, S. (2010), Privacy-preserving record linkage, in 'Privacy in Statistical Databases, Springer LNCS 6344', Corfu, Greece, pp. 269-283.
- Inan, A., Kantarcioglu, M., Bertino, E. & Scannapieco, M. (2008), A hybrid approach to private record linkage, in 'IEEE ICDE', Cancun, Mexico, pp. 496-505.
- Inan, A., Kantarcioglu, M., Ghinita, G. & Bertino, E. (2010), Private record matching using differential privacy, in 'EDBT'.
- Kantarcioglu, M., Jiang, W. & Malin, B. (2008), A privacy-preserving framework for integrating person-specific databases, in 'Privacy in Statistical Databases', Springer, pp. 298-314.
- 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.
- Karakasidis, A. & Verykios, V. (2011), 'Secure blocking+secure matching = secure record linkage', *Journal of Computing Science and Engineering* 5, 223-235.
- Karakasidis, A. & Verykios, V. (2012), Reference table based k-anonymous private blocking, in 'ACM Symposium on Applied Computing', Riva del Garda, Italy.
- Karakasidis, A., Verykios, V. & Christen, P. (2011), Fake injection strategies for private phonetic matching, in 'International Workshop on Data Privacy Management', Leuven, Belgium.
- Kargupta, H., Datta, S., Wang, Q. & Sivakumar, K. (2003), On the privacy preserving properties of random data perturbation techniques, in 'ICDM', IEEE, pp. 99-106.
- Kissner, L. & Song, D. (2004), Private and threshold set-intersection, in 'Technical Report', Carnegie Mellon University.
- Kuzu, M., Kantarcioglu, M., Durham, E. & Malin, B. (2011), A constraint satisfaction cryptanalysis of Bloom filters in private record linkage, in 'Privacy Enhancing Technologies', Springer, pp. 226-245.
- Lai, P., Yiu, S., Chow, K., Chong, C. & Hui, L. (2006), An Efficient Bloom filter based Solution for Multiparty Private Matching, in 'International Conference on Security and Management'.
- Lindell, Y. & Pinkas, B. (2009), 'Secure multiparty computation for privacy-preserving data mining', *Journal of Privacy and Confidentiality* 1(1), 5.
- Mohammed, N., Fung, B. & Debbabi, M. (2011), 'Anonymity meets game theory: secure data integration with malicious participants', *VLDB* 20(4), 567-588.
- O'Keefe, C., Yung, M., Gu, L. & Baxter, R. (2004), Privacy-preserving data linkage protocols, in 'ACM Workshop on Privacy in the Electronic Society'.
- Pang, C., Gu, L., Hansen, D. & Maeder, A. (2009), 'Privacy-preserving fuzzy matching using a public reference table', *Intelligent Patient Management* pp. 71-89.
- 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.
- Schnell, R., Bachteler, T. & Reiher, J. (2009), 'Privacy-preserving record linkage using Bloom filters', *BMC Medical Informatics and Decision Making* 9(1).
- Song, D., Wagner, D. & Perrig, A. (2000), 'Practical techniques for searches on encrypted data', *sp*.
- Trepetin, S. (2008), 'Privacy-preserving string comparisons in record linkage systems: a review', *Information Security Journal: A Global Perspective* 17(5), 253-266.
- Van Eycken, E., Haustermans, K., Buntinx, F., Ceuppens, A., Weyler, J., Wauters, E., VAN, O. et al. (2000), 'Evaluation of the encryption procedure and record linkage in the Belgian National Cancer Registry', *Archives of public health* 58(6), 281-294.
- Vatsalan, D., Christen, P. & Verykios, V. (2011), An efficient two-party protocol for approximate matching in private record linkage, in 'AusDM, CRPIT 121'.
- Vatsalan, D., Christen, P. & Verykios, V. (2013), 'A taxonomy of privacy-preserving record linkage techniques', to appear in *Journal of Information Systems*.
- Verykios, V., Karakasidis, A. & Mitrogiannis, V. (2009), 'Privacy preserving record linkage approaches', *Int. J. of Data Mining, Modelling and Management* 1(2), 206-221.
- Weber, S., Lowe, H., Das, A. & Ferris, T. (2012), 'A simple heuristic for blindfolded record linkage', *Journal of the American Medical Informatics Association*.
- Yakout, M., Atallah, M. & Elmagarmid, A. (2012), 'Efficient and practical approach for private record linkage', *Journal of Data and Information Quality* 3(3), 5.

VICUS - A Noise Addition Technique for Categorical Data

Helen Giggins¹

Ljiljana Brankovic²

¹ School of Architecture and Built Environment
The University of Newcastle,
University Drive, Callaghan, NSW 2308, Australia
Email: Helen.Giggins@newcastle.edu.au

² School of Electrical Engineering and Computer Science
The University of Newcastle,
University Drive, Callaghan, NSW 2308, Australia
Email: Ljiljana.Brankovic@newcastle.edu.au

Abstract

Privacy preserving data mining and statistical disclosure control have received a great deal of attention during the last few decades. Existing techniques are generally classified as restriction and data modification. Within data modification techniques noise addition has been one of the most widely studied but has traditionally been applied to numerical values, where the measure of similarity is straightforward. In this paper we introduce *VICUS*, a novel privacy preserving technique that adds noise to categorical data. Experimental evaluation indicates that *VICUS* performs better than random noise addition both in terms of security and data quality.

1 Introduction

Potential breaches of privacy during statistical analysis or data mining have implications for many facets of modern society (Brankovic & Estivill-Castro 1999, Giggins & Brankovic 2002, 2003). Privacy preserving data mining and statistical disclosure control focus on finding a balance between the conflicting goals of privacy preservation and data utility (Brankovic & Giggins 2007, Brankovic et al. 2007). Existing techniques are generally classified as *restriction* and *data modification* techniques (Brankovic & Giggins 2007). When restriction is applied, a user does not have access to microdata itself, but rather to a restricted collection of statistics (queries). In this context data utility is often referred to as *usability*, or the percentage of queries that can be answered without disclosure of any sensitive individual value (Brankovic et al. 1996a,b). Unfortunately, for general queries the usability tends to be very low (Brankovic & Miller 1995, Griggs 1997), especially when higher levels of privacy are required (Griggs 1999). However, if only range queries are of interest, which is the case in OLAP, the usability can be very high, providing that all cells of OLAP cubes contain positive counts (Brankovic

et al. 2000, 2002, Brankovic & Širán 2002, Horak et al. 1999, Brankovic et al. 1997).

Unlike restriction, data modification techniques allow for the microdata to be made available to users or, alternatively, can provide answers to any query, although the answers are not necessarily exact. Therefore, in this context, utility is often equated to data quality (Islam & Brankovic 2005). In principle, data modification techniques are applicable to both numerical and categorical attributes (Estivill-Castro & Brankovic 1999, Islam & Brankovic 2011); however, techniques such as noise addition are mostly applied to numerical attributes (Willenborg & de Waal 2001, Muralidhar & Sarathy 2003).

In the context of privacy, there have been two different focal points that attracted the most attention, prevention of membership disclosure (Sweeney 2002, Dwork 2006) and prevention of sensitive attribute disclosure (Machanavajjhala et al. 2007, Li et al. 2007, Brickell & Shmatikov 2008). Membership disclosure refers to revealing the existence of a particular individual in the database, while sensitive attribute disclosure occurs when an intruder is able to learn something about a particular individual's sensitive information (Brickell & Shmatikov 2008). The *k*-anonymity privacy requirement introduced by Samarati and Sweeney (Samarati & Sweeney 1998, Sweeney 2002) incorporates generalization to achieve its goal of ensuring that at least *k* records in the microdata file share values on the set of key attributes (quasi-identifiers). While this approach is successful in preventing membership disclosure, it does not prevent sensitive attribute disclosure if (1) there is not enough diversity in the sensitive attribute, or (2) the malicious user has significant background knowledge (Machanavajjhala et al. 2007). The *l*-diversity privacy requirement seeks to achieve sensitive attribute privacy by applying an additional requirement that there must exist at least *l* "well-represented" values of the sensitive attribute in each group of records sharing quasi-identifier values (Machanavajjhala et al. 2007). In the case of very strong background knowledge of the intruder, *l*-diversity may not be sufficient to prevent sensitive attribute disclosure (Li et al. 2007). A stronger requirements has been proposed, namely *t*-closeness, which compares the distances between the distributions of sensitive attribute over the whole microdata file to those for each grouping of records based on the quasi-identifiers (Li et al. 2007).

Differential privacy (Dwork 2006) attempts to capture the notion that one's privacy should not be at any greater risk of being violated by having one's information placed in the microdata file. This principle is applied to answering queries via an output pertur-

This work was supported by ARC Discovery Project DP0452182.

Copyright ©2012, Australian Computer Society, Inc. This paper appeared at the 10th Australasian Data Mining Conference (AusDM 2012), Sydney, Australia, December 2012. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 134, Yanchang Zhao, Jiuyong Li, Paul Kennedy, and Peter Christen, Ed. Reproduction for academic, not-for-profit purposes permitted provided this text is included.

Client	Branch	Financial Product	Advisor
Dr T. Green (1)	Hong Kong (7)	Income Protection Insurance (10)	Mr D. Smith (14)
Mr D. Blue (2)	Hong Kong (7)	Home Mortgage (11)	Mr D. Smith (14)
Mr M. Brown (3)	Newcastle (8)	Managed Investment (12)	Mr R. Jones (15)
Mrs H. Pink (4)	Newcastle (8)	Share Portfolio (13)	Ms W. Wong (16)
Mr K. White (5)	Sydney (9)	Share Portfolio (13)	Ms W. Wong (16)
Mr J. Black (6)	Sydney (9)	Managed Investment (12)	Ms W. Wong (16)
Mr J. Black (6)	Sydney (9)	Home Mortgage (11)	Mr M. James (17)

Table 1: Bank Client Microdata File - Sample

bation technique in (Dwork et al. 2006).

In this paper we focus on sensitive attribute disclosure, namely we aim to minimise the information an intruder is able to reveal about a sensitive attribute value belonging to an individual in the microdata file. By focusing on microdata files containing categorical data we are also limited in the way in which we can apply existing privacy requirements and SCD techniques. For instance, having no natural ordering of categories in an attribute makes the application of generalization techniques difficult when there is no obvious hierarchy to the values. Within data modification techniques noise addition had been one of the most widely studied, but have traditionally been applied to numerical values (see for example (Muralidhar & Sarathy 2003)), and when the data set contains categorical values the application of these techniques tends to be much less straightforward (Willenborg & de Waal 2001). In this paper we introduce “VICUS”, a novel noise addition technique for categorical attributes. An important step in VICUS is the clustering of categorical values while, in turn, an important component of any clustering technique is the notion of similarity between the attribute values. VICUS seeks to maximise the similarity between values in the same cluster, while minimising the similarity between values from different clusters.

In the next section we outline the similarity measure that will be employed in VICUS in Section 3. We first outline the motivation for the similarity measure, before formally defining it. We then present experimental results on several different data sets, which highlight the effectiveness of our measure. In Section 3 we propose VICUS, a noise addition technique for categorical values, which incorporates our similarity measure and assigns transition probabilities based on the discovered clusters of attribute values. We also provide an analysis of experimental results to see how well VICUS performs in the conflicting areas of security and data quality. We provide some concluding remarks in Section 4.

2 Similarity Measure

2.1 Motivating Example

The following example is designed to illustrate the relationships that exist in the microdata file, and how VICUS attempts to capture these relationships. Table 1 shows a sample Bank Client microdata file for customers buying financial products from a fictional bank and similar examples can be constructed from medical, marketing or criminal research area.

On examining Table 1 we can clearly see a connection between Dr Green and Mr Blue, as they are both customers of the Hong Kong branch and both see the same financial advisor. However, it may not be so obvious that there is a connection between Mr Brown and Mr White as they have no attribute values in common, have purchased different financial products

and are seen by different financial advisors at different branches. Nevertheless, these two clients have both purchased financial products that require the purchase of shares, so there should be some notion of similarity between them.

To better understand these connections between the customers we can represent the microdata shown in Table 1 as a graph (see Figure 1). This is done by assigning values that appear in the table to vertices. An edge appears between two vertices when the corresponding two values appear together in a record. Note that each record forms a clique in the graph. The red circled subgraph in Figure 1 represents record 7, that is, Mr Black who has a mortgage and is advised by Mr James at the Sydney branch.

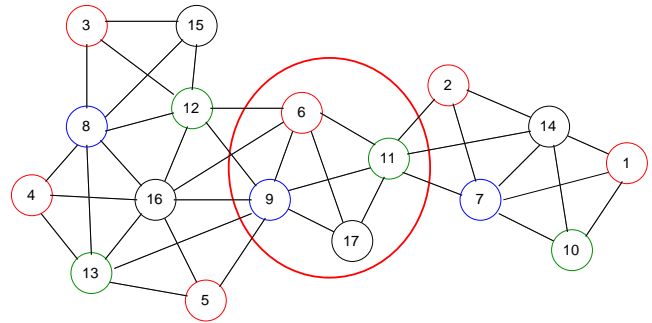


Figure 1: Motivating example microdata represented as a graph.

Note that we will be evaluating similarity only between vertices corresponding to the values of the same attribute in the data set, that is, vertices 1-6, 7-9, 10-13 and 14-17. In the sample database Mr Black (vertex 6) has direct similarity with every other client except for Dr Green. This direct similarity is indicated by one or more common neighbours of the corresponding vertices (or, equivalently, by a path of length two between the vertices).

Figure 1 shows that there are no common neighbours of vertex 3 (Mr Brown) and vertex 5 (Mr White). This effectively means that the records pertaining to Mr White and Mr Brown will have no values in common. So any method only looking at common values (neighbours) would not find these two values at all similar. However, looking at the data set it is clear that there is some transitive similarity between Mr Brown and Mr White, as they both purchased products which would typically be considered similar in the financial context. Although the products purchased by Mr White and Mr Brown were provided by different financial advisors, Mr R. Jones and Ms W. Wong, these two staff are considered similar because they both sell managed investment funds. Thus, Mr White and Mr Brown do indeed have similar products and are serviced by advisors of similar expertise. Consequently, we may still wish to consider Mr White and Mr Brown as similar. Our method

captures this kind of similarity by looking not just at common neighbours of two vertices, but also at common neighbours of their neighbours. We now outline how this type of similarity can be measured.

2.2 Evaluating Similarity

The first step in calculating similarity between attribute values is to create a corresponding graph, where there is an edge between vertices when two corresponding values appear together in a record. Note that we have considered both the simple graph and multigraph created from the data set. In the simple graph we create an edge between two attribute values if they co-occur in any record. In the multigraph form we count the number of co-occurrences of the two values and consider this as the number of edges between the two corresponding vertices.

The first type of similarity we consider is based on the values co-occurring in records. For example, in the Bank Client graph (Figure 1), we would consider that Dr Green and Mr Blue are similar since they see the same financial advisor at the same branch. This type of similarity, which we term S' similarity, is measured by the number of common neighbours of these two vertices in the graph. Looking at Figure 1 we see that vertex 1 (*Dr Green*) and vertex 2 (*Mr Blue*) are both adjacent to vertex 7 (*Hong Kong*) and vertex 14 (*Mr D. Smith*). We consider this as a high similarity since these two values share a majority of neighbours in the graph.

The second type of similarity examines ‘neighbours of neighbours’ and we denote it as S'' similarity, and measure it by first considering the S' similarity of ‘neighbours’. An example of this type of similarity as discussed in Section 2.1 is between Mr Brown and Mr White, who although do not share any attribute values, do have ‘similar’ values. For instance, the *Newcastle* and *Sydney* branches would be considered similar via an S' calculation. Similarly, the *Managed Investment* is similar to *Share Portfolio*, and *Mr R. Jones* is similar to *Ms W. Wong*. This means that all of the values that Mr Brown and Mr White appear with in the data set are considered similar via S' similarity, and hence these two values would have a high S'' similarity.

The Total Similarity S for two attribute values is taken to be composed of both the S' and S'' similarity for the values. We now provide a formal definition of our similarity measure S .

We calculate the total similarity S_{ij} as a weighted sum of the S'_{ij} and S''_{ij} :

$$S_{ij} = c_1 \times S'_{ij} + c_2 \times S''_{ij} \quad (1)$$

where $c_1 + c_2 = 1$. Typical values might be $c_1 = 0.65$ and $c_2 = 0.35$. In the next section we experiment with different values for c_1 and c_2 .

2.2.1 Similarity Measure - S_{ij}

We define a simple graph $G = (V, E)$ on n vertices and m edges, where $v \in V$ represents an attribute value in the data set. An edge $\{i, j\} \in E$ exists between two vertices $i, j \in V$ when the values i and j both appear together in one or more records in the data set. The adjacency matrix, A , for graph G will contain a 1 in position a_{ij} if an edge $\{ij\}$ appears between the vertices i and j , and 0 otherwise.

Input: Graph G , Threshold T

Output: S'' values for G

```

initialise  $S''$  matrix to 0;
for each attribute  $x \in G$  do
    get the list of attribute values  $val_x$ ;
    /* Loop over all pairs of values for the attribute  $x$  */
    for each value  $i \in val_x$  do
        for each value  $j \in val_x$  do
            initialise  $mergedGraph$  to  $G$ ;
            /* Loop over all attributes in  $G$ , excluding  $x$  */
            for each attribute  $y \in G \setminus x$  do
                get the list of attribute values  $val_y$ ;
                /* Loop over all pairs of values in  $y$  */
                for each value  $c \in val_y$  do
                    for each value  $d \in val_y$  do
                        if (there are edges  $\{c, i\}$  and  $\{d, j\} \cup \{c, j\}$  and  $\{d, i\}$  in  $G$ )  $\wedge$  ( $c$  and  $d$  not already in the same vertex in  $mergedGraph$ ) then
                            if  $S'_{cd} > \text{Threshold } T$  then
                                merge vertex  $c$  and  $d$  in  $mergedGraph$ ;
                                /* Note: if one vertex has already been merged with another, merge all together */
                            end
                        end
                    end
                end
            end
             $S''_{ij} = S'_{ij}$  calculated on  $mergedGraph$ 
        end
    end
end
return  $S''$  matrix;
    
```

Algorithm 1: Calculating S'' values for graph G

We define a multigraph $H = (V, E)$ on n vertices and m edges, where $v \in V$ represents an attribute value in the data set. An edge $\{i, j\} \in E$ exists between two vertices $i, j \in V$ for each record that contains both values i and j . We do not allow self-loops in this graph. In the adjacency matrix A for multigraph H , a_{ij} is the number of edges appearing between the vertices i and j in H .

The S'_{ij} similarity between two attribute values is given by

$$S'_{ij} = \frac{\sum_{k=1}^n \sqrt{a_{ik} \times a_{kj}}}{\sqrt{d(i) \times d(j)}} \quad (2)$$

where the sum is over all vertices in the graph G (or H), a_{lm} is the adjacency matrix entry for vertices l and m ($1 \leq l, m \leq n$) and $d(l)$ is the degree of vertex l . Note that S'_{ij} has a maximum value of 1 when the

two vertices have all their ($d(i) = d(j)$) neighbours in common, and a minimum value when two vertices have no neighbours in common ($S'_{ij} = 0$). S' values are only calculated within an attribute and not across attributes.

The S'_{ij} similarity captures a notion of transitive similarity for attribute values that are not necessarily directly connected to a common neighbour but are connected to similar values, that is, values which have a S'_{ij} value greater than the user defined threshold T . A basic version of an algorithm for calculating S''_{ij} is shown in Algorithm 1. Note that the actual algorithm used in experimental analysis is significantly more efficient than Algorithm 1.

2.3 Experiments - Similarity

In this section we present the results of experiments conducted on several data sets to observe the effectiveness of our similarity measure. Note that only a small subset of the full experimental analysis conducted is presented in this paper due to space restrictions.

2.3.1 Data Sets

Several data sets have been selected to best demonstrate various qualities and characteristics of our similarity measure S_{ij} .

Motivating Example. This is the same data set presented in Section 2.1, and is used to illustrate advantages of our technique over other similarity measures.

Mushroom. This data set was selected as it contains only categorical values, and although it is a classification data set, it has also been studied in the context of clustering (Guha et al. 2000). It is obtained from the UCI Machine Learning Repository (Asuncion & Newman 2007). The original data set contains 8124 instances on 23 attributes (including the class attribute), where we removed any records with missing values.

ACS PUMS. The American Community Survey (ACS) is conducted annually by the United States Census Bureau and was designed to provide a snapshot of the community. We took a random sample of 20,000 records from the 2006 Housing Records Public Use Microdata Sample (PUMS)¹ for the whole of the US. The sub-sample was chosen on only 14 attributes of the available 239, and any records with missing values on these attributes was not considered.

2.3.2 Parameter Selection

There is a certain amount of flexibility in the calculation of the similarity measure S . First, there is a choice for the value of the S'_{ij} threshold T , which is in the range $[0,1]$. One observation on the selection of this threshold is that for smaller/sparser graphs the threshold generally needs to be set at a lower value than it does for larger/denser graphs. The second parameter that needs to be selected is the weighting values c_1 and c_2 in Equation 1 where $c_1 + c_2 = 1$, and a typical value choice for these parameters would be $c_1 = 0.6$ and $c_2 = 0.4$. This gives a slightly higher weighting to S'_{ij} than to S''_{ij} . Finally, we have the choice of making this graph generated from the data

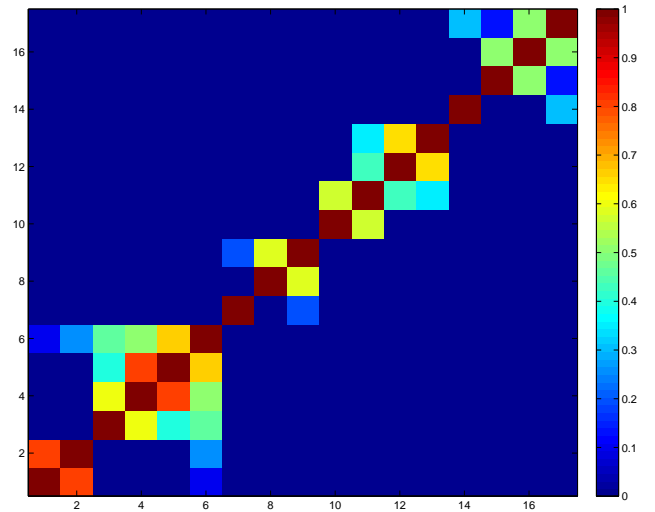


Figure 2: Similarity color map for Motivating Example.

set a simple graph or a multigraph, that is, a graph with multiple edges. When showing results for the similarity measure we generally present a range of parameters for comparison.

2.3.3 Results

We present a range of results to illustrate the effectiveness of our measure. One way in which we present the similarity values is via a colour map, as shown in Figure 2. The colour map assigns different colours to different values as per the colour bar on the right hand side of the diagram. Dark red maps to 1.0 and dark blue to 0.0. This figure shows S_{ij} values for the Motivating Example graph on all 17 values over the 4 attributes. The parameters are as follows; $T = 0.4$, $c_1 = 0.6$ and $c_2 = 0.4$. Value $S_{1,1}$ is in the bottom left hand corner of Figure 2, and value $S_{17,17}$ is in the right hand top corner. If you look at the diagonal between these two values, you will see that all values along the diagonal are 1.0, since each value has maximum similarity with itself. Areas outside of an attribute are dark blue since we do not consider the similarity between values from different attributes.

Motivating Example. Examining the similarity values we can compare the S'_{ij} and S''_{ij} values to the scenarios discussed in Section 2.1. Table 2 gives the S'_{ij} , S''_{ij} and S_{ij} similarity values for the first attribute in our motivating example, that is, *Client* and shows that Vertex 5 (Mr White) and Vertex 3 (Mr Brown) have no S'_{ij} similarity since they have no values in common in the data set. However, when the S''_{ij} threshold T is equal to 0.4, these two values have a S''_{ij} similarity of 1.0. This supports the notion that although these two clients do not have any direct similarity in the data set, they do have a transitive similarity which should be considered in any subsequent clustering of these values. By the appropriate assignment of values to c_1 and c_2 we can give the desired weight to this indirect similarity represented by S'_{ij} . In Table 2 we can see the situation for $c_1 = 0.6$ and $c_2 = 0.4$.

¹http://factfinder.census.gov/home/en/acs_pums.2006.html

S'_{ij}	1	2	3	4	5	6
1	1.000	0.667	0.000	0.000	0.000	0.000
2	0.667	1.000	0.000	0.000	0.000	0.258
3	0.000	0.000	1.000	0.333	0.000	0.258
4	0.000	0.000	0.333	1.000	0.667	0.258
5	0.000	0.000	0.000	0.667	1.000	0.516
6	0.000	0.258	0.258	0.258	0.515	1.000
S''_{ij}	1	2	3	4	5	6
1	1.000	1.000	0.000	0.000	0.000	0.258
2	1.000	1.000	0.000	0.000	0.000	0.258
3	0.000	0.000	1.000	1.000	1.000	0.775
4	0.000	0.000	1.000	1.000	1.000	0.882
5	0.000	0.000	1.000	1.000	1.000	0.882
6	0.258	0.258	0.775	0.882	0.882	1.000
S_{ij}	1	2	3	4	5	6
1	1.000	0.800	0.000	0.000	0.000	0.103
2	0.800	1.000	0.000	0.000	0.000	0.258
3	0.000	0.000	1.000	0.600	0.400	0.465
4	0.000	0.000	0.600	1.000	0.800	0.508
5	0.000	0.000	0.400	0.800	1.000	0.663
6	0.103	0.258	0.465	0.508	0.663	1.000

Table 2: S'_{ij} , S''_{ij} and S_{ij} values for *Client* attribute in Motivating Example.

Mushroom. This data set has only categorical attributes. The results for selected attributes are presented in Figure 3 for parameters $T = 0.6, c_1 = 0.6$ and $c_2 = 0.4$. It can be seen from the figure that for some attributes such as Cup Shape, Cup Colour, Stalk Root and Habitat, all pairs of values exhibit similarity, while for attributes such as Stalk Colour Below Ring and Ring Type there are values which are not very similar to any other value.

ACS PUMS. This data set is good mixture of categorical and numerical attributes of varying sizes. A sample of total similarity results for parameters $T = 0.75, c_1 = 0.6$ and $c_2 = 0.4$ are shown in Figure 4.

The total similarity across all attributes is shown in the image in the top left hand corner of Figure 4, while the S'_{ij} and S''_{ij} values are shown in the bottom right hand corner. There are several numerical values worth mentioning here, including the two attributes related to income, WAGP (*Wages or income in the past 12 months*) and PINCP (*Person's total income*). Both of these attributes exhibit very numerical tendencies in that values that are close together numerically tend to be more similar than those that are further apart numerically. However, there is a noted exception to this rule for the attribute PINCP, since when the income is below zero these values have very low similarity to values just above zero, yet appear more similar to high incomes.

Another attribute worth noting is *Educational attainment* (SCHL), in the bottom row of Figure 4. The values in this attribute appear to be partitioned into two distinct groups that have a high level of similarity within a partition, and lower similarity outside of it. The two values at the boundary of these two groups are values 8 and 9, which correspond to 'Grade 12 no diploma' and 'High school graduate' respectively. This result indicates that based on the subset of attributes in the data set, there is a strong relationship between levels of education above that of high school graduate, and also between the levels of education that fall below this benchmark.

An example of a numerical attribute from the ACS PUMS data set which does not exhibit a numerical ordering is that of 'Usual hours worked per week last 12 months' (WKHP), shown in the top right hand

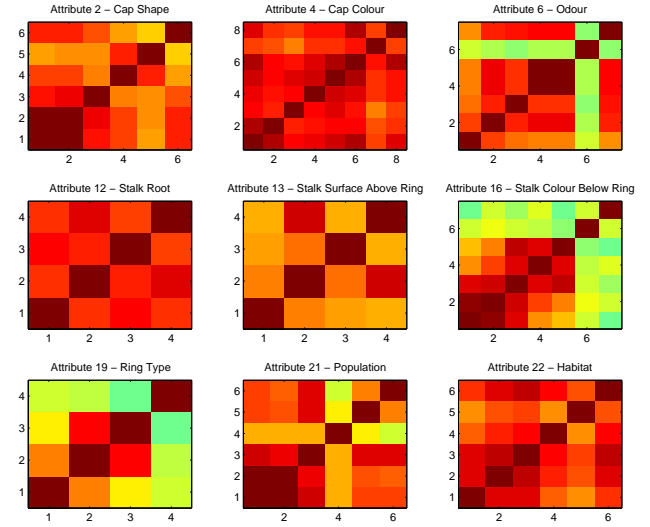


Figure 3: A close look at S_{ij} values for selected attributes in Mushroom. ($T = 0.6, c_1 = 0.6, c_2 = 0.4$).

corner of Figure 4. Although there are quite a few of the values which are numerically close that also have a high level of similarity, there are also many values which do not follow this convention.

In the next section we will incorporate our similarity measure into a noise addition technique for categorical values.

3 Noise Addition

In this section we propose a noise addition technique for categorical values which incorporates our similarity measure from Section 2 and uses it to cluster these values. It then assigns transition probabilities based on the discovered clusters. We also provide an analysis of experimental results to see how well our technique performs in the conflicting areas of security and data quality.

Recall our Motivating Example from Section 2.1. Having evaluated the similarity values for the attributes in this data set, we are now faced with the problem of how best to partition the values so as to maximise similarity within a partition, and minimise similarity across partitions. Although it is not difficult to define a maximisation function that will indicate the quality of a selected partitioning of the graph, it is more challenging to decide how best to arrive at an optimal solution.

3.1 VICUS - Noise Addition Technique

Noise is added to a data set by applying the following three steps.

Step 1: We partition the graph using the similarity measure for values within an attribute. We use a genetic algorithm to explore the solution space and arrive at a close to optimal partitioning of the graph. **Step 2:** Using the partitioning of the graph obtained from Step 1, we generate a transition probability matrix for all attribute values. The transition matrix gives the probabilities of each attribute value changing to every other value within the attribute.

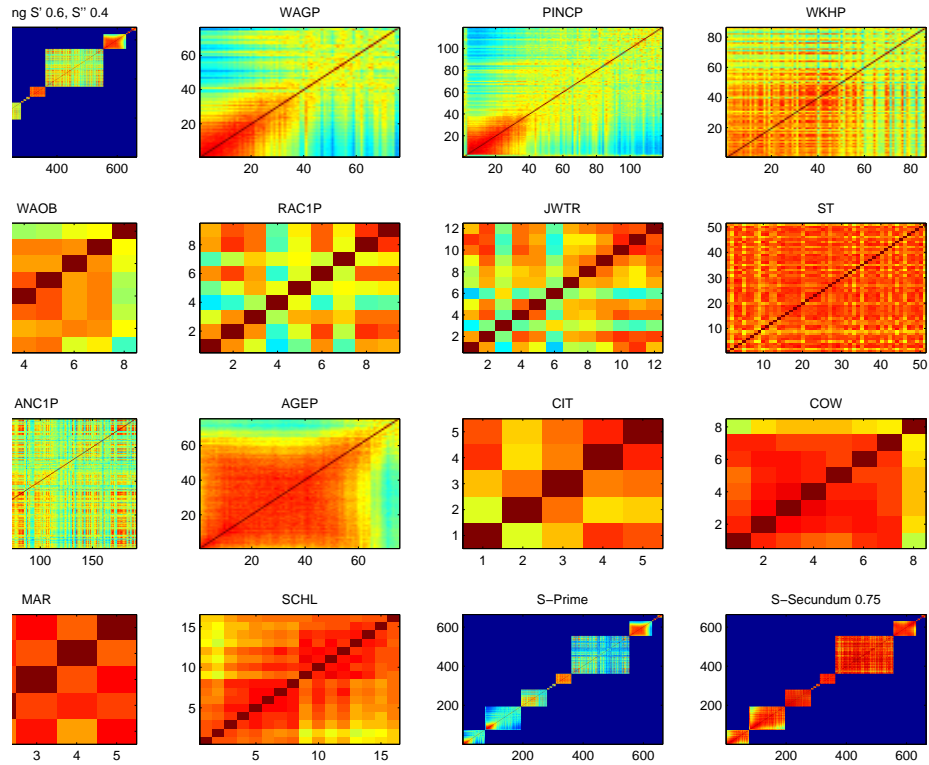


Figure 4: Sample results for Census PUMS data set ($T = 0.75, c_1 = 0.6, c_2 = 0.4$).

Step 3: We perturb each individual value in the original data file by applying the transition probabilities. Note that the value will generally have a relatively high probability of remaining the same in the perturbed file.

We next describe each of the steps in more detail.

Graph Partitioning. We now define the graph partitioning problem as presented in Bui and Moon (Bui & Moon 1996). Given a graph $G = (V, E)$ on n vertices and m edges, we define a partition \mathcal{P} to consist of disjoint subsets of vertices of G . The *cut-size* of a partition is defined to be the number of edges whose end-points are in different subsets of the partition. A balanced k -way partition is the partitioning of the vertex set V into k disjoint subsets where the difference of cardinalities between the largest subset and the smallest one is at most one. The k -way partitioning problem is the problem of finding a k -way partitioning with the minimum cut-size (Bui & Moon 1996). We relax the condition of difference of partition sizes being at most 1, and we impose a lower bound on the minimum size of the partition $minS$. The k -way partitioning problem has been well studied and has been shown to be NP -complete in both the balanced and unbalanced form (Garey & Johnson 1979, Bui & Moon 1996). Hence, we will apply a heuristic, namely a genetic algorithm, to solve the problem of moving from one solution to the next.

Transition Probability Matrix. When deciding how much noise to add when we perturb a data set, we must decide on how best to distribute the transition probabilities amongst the possible choices. Note that we explore two separate methods for defining

the transition probabilities, the first being our *VICUS* method, and the second being a method we term *Random*, which is used to evaluate the effectiveness of the *VICUS* method. We next describe both of them in more detail.

VICUS Method. Given a partition \mathcal{P} of the original data set which divides all possible values into k disjoint sets, we calculate the transition probabilities for each attribute individually. We use the following notation.

B - the set of all values in the microdata file.

$\mathcal{P} = \{B_1, B_2, \dots, B_k\}$ - a partition - a collection of disjoint subsets (some of which may be empty) of B such that $\bigcup_{i=1}^k B_i = B$.

A - the set of values of attribute A .

$\mathcal{P}_A = \{A_1, A_2, \dots, A_k\}$ - a partition - a collection of disjoint subsets (some of which may be empty) of A such that $\bigcup_{i=1}^k A_i = A$ and $A_i \subset B_i, 1 \leq i \leq k$.

\mathcal{S} - the subset from \mathcal{P}_A containing an attribute value, $a \in A$.

$|\mathcal{S}|$ is the number of attribute values in the subset containing the value a .

$\mathcal{S}' = \mathcal{P}_A \setminus \mathcal{S}$ - the relative complement set containing all other subsets.

$|\mathcal{S}'|$ is the number of attribute values that are in a different subset to a .

P_s - the probability of an attribute value remaining unchanged.

p_{sp} - the probability that an attribute value is changed into a different attribute value from the same subset.

$P_{sp} = (|\mathcal{S}| - 1) \times p_{sp}$ - the probability that the attribute value remains in the same subset, but not unchanged.

p_{dp} - the probability that the value a changes to a value from a different subset.

$P_{dp} = |\mathcal{S}'| \times p_{dp}$ - the probability that the attribute value changes the subset.

The transition probabilities for an attribute value a satisfy the following;

$$P_s + P_{sp} + P_{dp} = 1 \quad (5)$$

We now introduce two parameters that allow the data manager to adjust the amount of noise to be added to the microdata file. The first parameter, k_1 , is defined such that an attribute value a is k_1 times more likely to stay the same than to change to another value in the same subset. The second parameter, k_2 , tells us how many times more likely a value a is to change to another in the same subset than one in a different subset. Hence, we can reformulate our probabilities as

$$P_s = k_1 \times p_{sp} = k_1 \times k_2 \times p_{dp} \quad (8)$$

and Equation 5 becomes

$$P_s + (|\mathcal{S}| - 1) \times \frac{P_s}{k_1} + |\mathcal{S}'| \times \frac{P_s}{k_1 \times k_2} = 1 \quad (9)$$

From the above, the probability that a value remains the same becomes

$$P_s = \frac{k_1 \times k_2}{k_1 \times k_2 + k_2 \times (|\mathcal{S}| - 1) + |\mathcal{S}'|} \quad (9)$$

$$p_{dp} = \frac{1}{k_1 \times k_2 + k_2 \times (|\mathcal{S}| - 1) + |\mathcal{S}'|} \quad (10)$$

$$p_{sp} = \frac{k_2}{k_1 \times k_2 + k_2 \times (|\mathcal{S}| - 1) + |\mathcal{S}'|} \quad (11)$$

3.1.1 Random Method

We also define a set of transition probabilities for the method we term *Random*. This method does not assign probabilities for P_{sp} and P_{dp} , but rather introduces the probability of a value changing to any other value in the attribute, which is denoted P_c . However, it still uses Equation 9 to calculate the probability of a value remaining unchanged in the perturbed data set. We define the probability of a value changing to any other value in the attribute as follows

$$P_c = \frac{1 - P_s}{|\mathcal{S}| + |\mathcal{S}'| - 1} \quad (12)$$

The resulting method will perform better than a truly random method, as it is imparting some of the information from our partitioning of the values when calculating the value for P_s . However, to evaluate the quality of our method we need to perturb the 'random' in such a way as to be able to compare the results of our security measure and data quality tests.

Perturbing Microdata File. Once the transition probability matrix has been generated for each attribute, the next step is to simply perturb the original microdata file according to the transition probabilities assuming that a random value is drawn to decide if the value changes to another value in the same partition, one from a different partition, or remains unchanged.

3.2 Evaluation Methods

We now evaluate *VICUS* both in terms of security and data quality. In evaluating the security of a perturbed data set we assume that the intruder is aware of the exact perturbation technique. We apply an information theoretic entropy (Shannon 1948) measure to estimate the amount of uncertainty the intruder has about the identity of a record as well as the value of a confidential attribute. In order to gauge how well our noise addition technique preserves the underlying data quality, we apply the chi-square statistic test.

```

Input: Transition Probability matrix  $M$ ,
          Perturbed microdata file  $P$ ,
          Probabilities  $p_x$  for all records
Output:  $H(D)$  entropy
for each value  $c_i \in C$  do
    | initialise probability  $D_i$  to 0;
end
for each record  $x$  in  $P$  with  $C_x$  for  $C$  do
    | for each confidential value in  $c_i \in C$  do
    |   /* Sum the probability that  $C_x$  in
    |   /*  $P$  originated from  $c_i$  */
    |   /* in  $O$  and multiply by the
    |   /* probability that
    |   /* record  $x$  is the record the
    |   /* intruder 'knows'
    |    $D_i + = p(c_i = O(C_x)) \times p_x$ ;
    | end
    end
    /* Now calculate the entropy of the
    /* confidential value  $V_c$  */
     $H(D) = \sum_1^{|\mathcal{C}|} D_i \log_2 \frac{1}{D_i}$ ;
return  $H(D)$ ;
    
```

Algorithm 2: Calculating entropy of confidential attribute in perturbed microdata file.

Security Measure. One way in which we can measure the security of a released microdata file is by estimating how certain an intruder is that they have identified a record, and more importantly the correct confidential value for that record (Öganian & Domingo-Ferrer 2003). To gauge the amount of uncertainty an intruder has about having identified a particular record in the perturbed microdata file, we calculate the entropy for this record. Similarly, by calculating the entropy of a confidential value we can estimate the amount of uncertainty the intruder has about this value. We assume that there is only one confidential or sensitive attribute in the microdata file; it is straightforward to generalise to a case where there is more than one confidential attribute. We assume that an intruder (1) knows how noise has been added to the microdata file (2) knows one or more

attribute values about a particular record for which they wish to learn the confidential value (3) only has access to the perturbed and not the original micro-data file (4) is trying to compromise one particular record in the database and that they know the original values of some or all non-confidential attributes for that record. The algorithm used to calculate the entropy of a confidential attribute is given in Algorithm 2.

Data Quality. Information loss is an important consideration when evaluating the quality of a perturbation technique (Trotini 2003). The goal of the data manager is to minimise the reduction in data quality while at the same time maximise the security of released data. In order to evaluate how *VICUS* performs in terms of information loss apply a chi-square statistical test to both the original and perturbed data sets to ascertain how successfully *VICUS* preserves the underlying statistics from the original data set.

The *chi-square test* is a commonly applied statistical measure for determining the statistical significance of an association between two categorical attributes (Utts & Heckard 2004). We follow the five step approach to determining statistical significance as outlined by Utts and Heckard (Utts & Heckard 2004, p.184).

Note that our aim here is not to determine if there is any statistical significance of the attribute associations studied, rather we aim to determine that any such significance is undisturbed by our perturbation method.

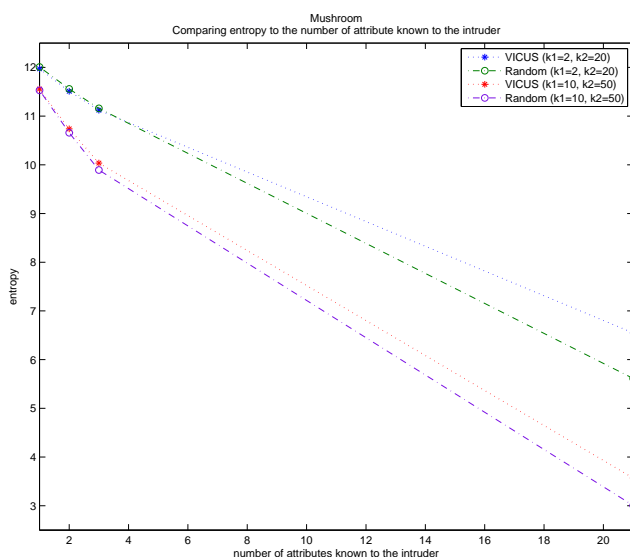


Figure 5: Record entropy vs number of known attributes, Mushroom data set.

We analyse our data set in the form of two-way contingency tables, which count the co-occurrence of a categorical value from one attribute with a value in another attribute, for all combinations of values.

Since we are dealing with categorical data specifically, and all of our experimental data sets have been perturbed under this assumption, the chi-square statistic is a natural choice. The chi-square statistic, χ^2 , measures the difference between the observed counts in the contingency table and the so-called expected counts, which are those that would occur if the there was no relationship between the two categori-

cal variables (Utts & Heckard 2004). An alternative statistic is the Likelihood Chi-Square χ^2_{LR} was also used in our analysis. For large data sets the values of χ^2 and χ^2_{LR} should be comparable. The same method is used to compare these statistics to the p -value to evaluate the correctness of the null hypothesis.

3.3 Experiments - Noise Addition

From the data sets examined in Section 2.3, where we looked at experimental results for our similarity measure, we present only the results for the Mushroom Data Set in this paper. In preparing our experiments we generated a multigraph from the original data set, and we ran a genetic algorithm 30 times and selected the partition with the largest fitness function. We next select five different combinations of parameters for the transition probability generation, and for each selection we perturbed 30 files according to the generated transition probabilities.

Security. We calculated the entropies for the situation when the user knows one, two, three and all of the attributes excluding the confidential one, to give a comparison ‘worst case scenario’ result. For each of the 30 perturbed files, we averaged the entropies over all records and all files for when the intruder knows one attribute value, and present the results in Table 3. The average entropies do not differ significantly between *VICUS* and *Random* method and the range between 11 and 12 bits for record entropy and 1.6 and 2.6 bits for confidential attribute entropy.

In Figure 5 we show how the entropy drops when the user learns more attribute values for a particular record. We are also interested to know if certain attributes are more or less revealing than others, that is, if they yield a lower or higher entropy than average. Figure 6 provides a close up view of the entropies for each individual attribute.

Data Quality. We used the SPSS Statistical Software package to analyse the Chi-square statistics of the mushroom data set. For each attribute pair we calculated the Pearson’s Chi-Square statistic and Likelihood Ratio Chi-Square statistic on the original data set, 30 files perturbed using *VICUS* method and 30 file perturbed via the *Random* method. We compared both the Chi-square statistic value and associated p -value for each. We first want to see *VICUS* performed in terms of how far the χ^2 values were from those on the original file and the Randomly perturbed files. We next wanted to verify if there was a change in the outcome of the null hypothesis for the files perturbed with the *VICUS* method.

We chose to look at Attribute 4 (*Cap Colour*) against the other attributes, since this attribute showed to be the most sensitive in terms of security when we calculated the entropy for the user knowing one attribute value (Figure 6).

We also selected only a single combination of values for the parameters, namely $k_1 = 5$ and $k_2 = 20$, as this combination gave middle of the range results on entropy.

Of the 21 attribute combinations, there were 7 attribute pairs that satisfied the large sample requirement on the original data set. That is, these attributes had over 80% of cells in the contingency table with expected counts larger than 5, and all cells had an expected count larger than 1.

Figure 7 compares the distributions of the χ^2 statistic values for the 30 files perturbed via the *VICUS* and *Random* methods, and shows how far away they are from the χ^2 statistic for the original file,

Perturbation	k_1	k_2	VICUS	Random	VICUS	Random
			Record	Entropy	Confidential	Entropy
Mush1	2	20	11.9720	12.0074	2.0743	2.4496
Mush2	5	20	11.7502	11.7316	1.8778	2.1962
Mush3	10	10	11.6141	11.5772	1.8280	2.0036
Mush4	10	20	11.5826	11.5487	1.7220	1.9450
Mush5	10	50	11.5567	11.5284	1.6409	1.9045

Table 3: Average record and confidential attribute entropy.

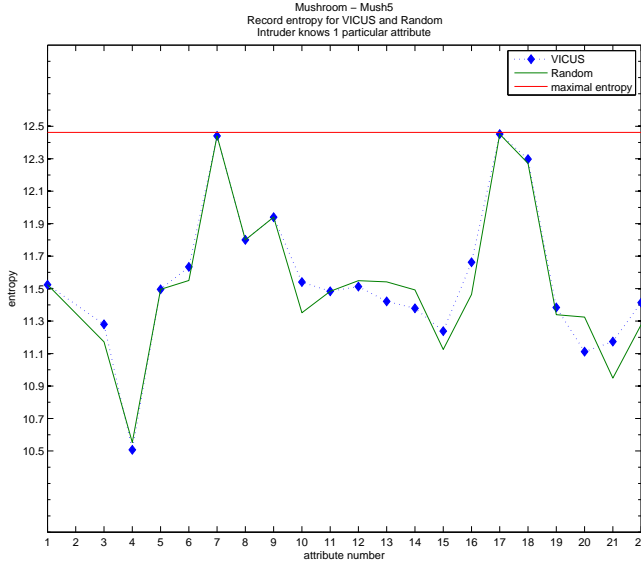


Figure 6: Record entropy sensitivity.

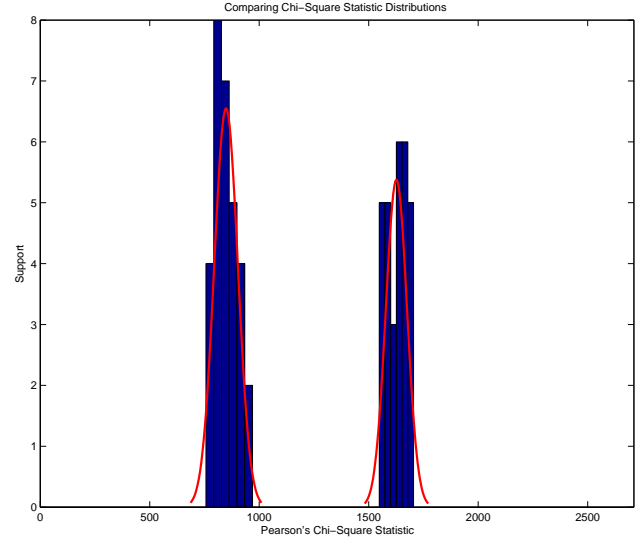


Figure 7: Chi-square statistic

which had the value of 2711.8. For all 30 files *VICUS* had the χ^2 greater than 1500, while *Random* had the χ^2 value less than 1000. On all attribute pairs examined, our *VICUS* method produced χ^2 values closer to the original than the those of the random method.

4 Conclusion

In this paper we have presented a new noise addition technique, *VICUS*, for application on categorical data. The first step of *VICUS* is to calculate the similarity between the categorical attribute values. To this end we have designed a new similarity measure specifically for use on categorical attribute values. The similarity measure aims to capture the notion of transitive similarity between values of an attribute, the so called S'' similarity. As the results of experimental analysis show, our similarity measure is effective in capturing the similarities that occur in the microdata when values no neighbours in common. Although *VICUS* is designed for application on categorical values, it can also be applied to numerical attributes by treating the discrete values as categories. The experiments showed that not all numerical attributes exhibit a numerical ordering according to our similarity measure, that is, values that are numerically close together do not necessarily have a high similarity. This would seem to indicate that the application of traditional numerical noise addition techniques on such attributes could result in reduced quality of the perturbed data set. Experimental results indicate that *VICUS* performs well in both the areas

of security and data quality. We observed that a low value for k_1 and high value for k_2 transition probability parameters lead to improved performance in terms of both data quality and security of *VICUS* over the *Random* method. Setting the product ($k_1 \times k_2$) of these parameters to a value of 100 or higher, while also ensuring that k_1 is low and k_2 is high appears to give the best balance between the conflicting goals of security and data quality.

References

- Asuncion, A. & Newman, D. (2007), 'UCI machine learning repository'.
URL: <http://www.ics.uci.edu/~mllearn/MLRepository.html>
- Brankovic, L. & Estivill-Castro, V. (1999), Privacy issues in knowledge discovery and data mining, in 'Proceeding of the Australian Institute of Computer Ethics, AICE99', Lilydale, Melbourne, Australia.
- Brankovic, L. & Giggins, H. (2007), *Security, Privacy and Trust in Modern Data Management*, Springer, New York, chapter Statistical Database Security, pp. 167–182.
- Brankovic, L., Horák, P. & Miller, M. (2000), 'An optimization problem in statistical databases.', *SIAM J. Discrete Math.* **13**(3), 346–353.
- Brankovic, L., Horak, P., Miller, M. & Wrightson, G. (1997), Usability of compromise-free statistical databases for range sum queries, in 'Proceed-

- ing of Ninth International Conference on Scientific and Statistical Database Management, IEEE Computer Society', August 11-13, Olympia, Washington, pp. 144-154.
- Brankovic, L., Islam, M. Z. & Giggins, H. (2007), *Security, Privacy and Trust in Modern Data Management*, Springer, New York, chapter Privacy Preserving Data Mining, pp. 151-165.
- Brankovic, L. & Miller, M. (1995), 'An application of combinatorics to the security of statistical databases', *Australian Mathematical Society Gazette* **22**(4), 173-177.
- Brankovic, L., Miller, M. & Širáň, J. (1996b), 'Graphs, 0-1 matrices, and usability of statistical databases', *Congressus Numerantium* **12**, 196-182.
- Brankovic, L., Miller, M. & Širáň, J. (2002), 'Range query usability of statistical databases', *Int. J. Comp. Math.* **79**(12), 1265-1271.
- Brankovic, L., Miller, M. & Širáň, J. (1996a), Towards a practical auditing method for the prevention of statistical database compromise, in 'Proceeding of Australasian Database Conference'.
- Brankovic, L. & Širáň, J. (2002), 2-compromise usability in 1-dimensional statistical databases, in 'Proc. 8th Int. Computing and Combinatorics Conference, COCOON2002'.
- Brickell, J. & Shmatikov, V. (2008), The cost of privacy: destruction of data-mining utility in anonymized data publishing, in 'KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, New York, NY, USA, pp. 70-78.
- Bui, T. N. & Moon, B. R. (1996), 'Genetic algorithm and graph partitioning', *IEEE Transactions on Computers* **45**(7), 841-855.
- Dwork, C. (2006), 'Differential privacy', *Lecture Notes in Computer Science* **4052**, 1-12.
- Dwork, C., McSherry, F., Nissim, K. & Smith, A. (2006), Calibrating noise to sensitivity in private data analysis., in 'Proceedings of Third Theory of Cryptography Conference, TCC 2006', New York, NY, USA, pp. 265-284.
- Estivill-Castro, V. & Brankovic, L. (1999), Data swapping: Balancing privacy against precision in mining for logic rules, in 'Proceedings of Data Warehousing and Knowledge Discovery, DaWaK99', Florence, Italy, pp. 389-398.
- Garey, M. R. & Johnson, D. S. (1979), *Computers and intractability: A guide to the theory of NP-completeness*, W. H. Freeman and Company, San Francisco.
- Giggins, H. & Brankovic, L. (2002), Ethical and privacy issues in genetic databases, in 'Proceedings of the Third Australian Institute of Computer Ethics Conference', Sydney, Australia.
- Giggins, H. & Brankovic, L. (2003), Protecting privacy in genetic databases, in R. L. May & W. F. Blyth, eds, 'Proceedings of the Sixth Engineering Mathematics and Applications Conference', Sydney, Australia, pp. 73-78.
- Griggs, J. R. (1997), 'Concentrating subset sums at k points', *Bulletin Institute Combinatorics and Applications* **20**, 65-74.
- Griggs, J. R. (1999), 'Database security and the distribution of subset sums in \mathbf{R}^m ', *János Bolyai Math. Soc. 7, Graph Theory and Combinatorial Biology* pp. 223-252.
- Guha, S., Rastogi, R. & Shim, K. (2000), 'Rock: A robust clustering algorithm for categorical attributes', *Information Systems* **25**(5), 345-366.
URL: citeseer.ist.psu.edu/guha00rock.html
- Horak, P., Brankovic, L. & Miller, M. (1999), 'A combinatorial problem in database security', *Discrete Applied Mathematics* **91**(1-3), 119-126.
- Islam, M. Z. & Brankovic, L. (2005), Detective: A decision tree based categorical value clustering and perturbation technique in privacy preserving data mining, in 'Proceedings of the 3rd International IEEE Conference on Industrial Informatics (INDIN 2005)', Perth, Australia.
- Islam, M. Z. & Brankovic, L. (2011), 'Privacy preserving data mining: A noise addition framework using a novel clustering technique', *Knowledge-Based Systems* **24**, 1214-1223.
- Li, N., Li, T. & Venkatasubramanian, S. (2007), t -closeness: Privacy beyond k -anonymity and l -diversity, in 'Proceedings of IEEE 23rd International Conference on Data Engineering, 2007. ICDE 2007.', pp. 106-115.
- Machanavajjhala, A., Kifer, D., Gehrke, J. & Venkatasubramanian, M. (2007), 'L-diversity: Privacy beyond k -anonymity', *ACM Trans. Knowl. Discov. Data* **1**(1), 3.
- Muralidhar, K. & Sarathy, R. (2003), 'A theoretical basis for perturbation methods', *Statistics and Computing* **13**, 329-335.
- Oganian, A. & Domingo-Ferrer, J. (2003), 'A posteriori disclosure risk measure for tabular data based on conditional entropy', *SORT - Statistics and Operations Research Transactions* **27**(2), 175-190.
- Samarati, P. & Sweeney, L. (1998), Protecting privacy when disclosing information: k -anonymity and its enforcement through generalization and suppression, in 'Proceedings of the IEEE Symposium on Research in Security and Privacy', Oakland, California, USA.
- Shannon, C. E. (1948), 'A mathematical theory of communication', *Bell Syst. Tech. J.* **27**, 379-423.
- Sweeney, L. (2002), ' k -anonymity: a model for protecting privacy. international journal on uncertainty', *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems* **10**(5), 557-570.
- Trottini, M. (2003), Assessing disclosure risk and data utility: A multiple objectives decision problem, in 'Joint ECE/Eurostat Work Session on Statistical Confidentiality', Luxembourg.
- Utts, J. M. & Heckard, R. F. (2004), *Mind on statistics*, 2nd edn, Thomson-Brooks/Cole, Belmont, Calif.
- Willenborg, L. & de Waal, T. (2001), *Elements of Statistical Disclosure Control*, Lecture Notes in Statistics, Springer, New York, USA.

Cartesian Genetic Programming for Trading: A Preliminary Investigation

Michael Mayo

School of Computing and Mathematical Sciences
University of Waikato, Hamilton, New Zealand
Email: mmayo@waikato.ac.nz

Abstract

In this paper, a preliminary investigation of Cartesian Genetic Programming (CGP) for algorithmic intraday trading is conducted. CGP is a recent new variant of genetic programming that differs from traditional approaches in a number of ways, including being able to evolve programs with limited size and with multiple outputs. CGP is used to evolve a predictor for intraday price movements, and trading strategies using the evolved predictors are evaluated along three dimensions (return, maximum drawdown and recovery factor) and against four different financial datasets (the Euro/US dollar exchange rate and the Dow Jones Industrial Average during periods from 2006 and 2010). We show that CGP is capable in many instances of evolving programs that, when used as trading strategies, lead to modest positive returns.

Keywords: Cartesian Genetic Programming, Algorithmic Trading, Rule Learning

1 Introduction

Algorithmic trading is the problem of automating decisions to buy and sell financial assets such that, even after trading costs and losses are taken into account, the cumulative net return from the decision series is positive. The main tasks of these decision strategies are (i) market direction prediction and (ii) position sizing, risk management, and entry/exit management. The main problem with task (i), of course, is that markets are notoriously difficult to predict. In fact, there is a long history of debate about the efficient market hypothesis (Fama, 1970) and the issue of whether or not market price movements are essentially random walks (see, for example, Beechey et al. (2000) for a recent counter-analysis). In spite of this, past research efforts from computer scientists appear to show that pattern recognition techniques such as machine learning can make profits in the markets. Recent examples include the works of Contreras et al. (2012), Lean and Lai (2007), Liu and Xiu (2009), Ni and Yin (2009), Barbosa and Belo (2008), Hirabayashi et al. (2009), and Larkin and Ryan (2010).

Putting aside the debate for a moment, task (ii) mentioned above (which is concerned with the details about whether to act on a prediction and if so, how to

act) is also not without its difficulties. For example, a market may be quiet one day and volatile the next. Therefore a strategy that that assigns a large position size to a trade on the quiet day (where the risk is low) may be in violation of its own risk management rules if it assigns the same position size the following day (where the risk is higher due to an increased likelihood of sharp price movements). Markets behaviours are well known to be non-stationary series (Sewell, 2011) and therefore methods and strategies that worked in the past cannot be expected to continue working. Furthermore, non-stationarity applies not just to prices but also to other important factors such as volatility and seasonal aspects (where seasonality includes not only properties that change with an annual cycle but also those that follow intraday, time-of-day-based cycles and weekly cycles). Non-stationarity probably explains why some technical strategies that traditionally worked in the past now may no-longer yield profits.

This paper takes the view that intraday market prices may be predictable to a small degree, although that “edge” may be very slim indeed. Financial engineering may be required to actually make such predictions profitable. We also take the stance that due to non-stationarity, a large amount of past training data is *not* required for learning trading strategies – in fact, too much data may lead to problems such as the learning of patterns that are now defunct. We therefore, in our experiments, use two months of intraday data to learn a trading strategy, and test it on the following month of intraday data.

The machine learning method of choice in this paper is Cartesian Genetic Programming (CGP) (Miller, 2011). We chose this method for a number of reasons. Firstly, CGP evolves programs that can have a fixed upper limit on size because they are represented as a fixed-size array. In contrast, traditional tree-based Genetic Programming methods have no limit on size and the problems of bloat are well known. Secondly, CGP can evolve programs with multiple outputs as well as multiple inputs. Although we do not use more than one output in the experiments presented here, in the future this would be advantageous for learning trading strategies because the multiple outputs can be used to emit different aspects of the strategy. For example one output may be a prediction of direction, and the second output may be a position size indicator (with a zero indicating “no trade”). A third output could possibly be a distance to a stop loss price.

The third and final reason for CGP being interesting from a trading perspective is that as learning proceeds over time (in generations), programs tend to reduce in complexity whenever fitness hits a plateau (Miller, 2011). That is, in the absence of further im-

Copyright ©2012, Australian Computer Society, Inc. This paper appeared at the 10th Australasian Data Mining Conference (AusDM 2012), Sydney, Australia, December 2012. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 134, Yanchang Zhao, Jiuyong Li, Paul Kennedy, and Peter Christen, Ed. Reproduction for academic, not-for-profit purposes permitted provided this text is included.

provements, CGP programs tend to have less active nodes due to the genetic drift feature of CGP. For a trading strategy, this is a very desirable property because smaller programs are easier for humans to understand, making them more like “traditional” indicators. Furthermore, smaller programs are less prone to overfitting.

2 Background

In this section, a brief review is given of the important concepts used in this paper. In particular, we describe the CGP approach used, and overview the important financial ideas.

2.1 Cartesian Genetic Programming

CGP is a relatively new field of genetic programming. It has found application in areas either where there is a significant amount of low-level data to be processed (e.g. in the evolution of image processing filters (Sekanina et al., 2011a)) or where the programs must adhere to significant physical constraints (e.g. the layout of circuits on a board (Sekanina et al., 2011b)). Financial applications are more related to the image processing scenario, because a trading strategy can be thought of as a “filter” on data that produces an output (that being signal to trade or not to trade), where the data is not 2D image data but is instead a stream of historical 1D price series data.

The canonical CGP algorithm (described more fully in (Miller, 2011)) is defined as follows. Firstly, a small number of fixed parameters must be specified. The first is the population size *popsiz*, which in canonical CGP is set to 5. This very small population size is offset by the fact that it is customary for CGP to run for a very large number of generations, *maxgens*, which may have a value in the millions.

Further parameters describe the fixed features of each program, such as the number of inputs, n_{in} ; the number of outputs, n_{out} ; and the maximum number of function call instructions (or *nodes*) in a program, n_l . Typically there also needs to be a fixed *arity* parameter that specifies the number of inputs each function/node takes. We also in this research fix the number of point mutations per offspring to n_m , although in general this parameter need not be fixed.

Next, a table *Functions* must be defined. A program in CGP is defined as a linear array of function calls of length n_l . Traditionally, basic numerical functions such as addition, subtraction, sin, cosine, square root, etc are used; alternatively, if the domain is logic circuits, low level AND and OR gates are sensible choices for functions. In our domain, we are interested in learning programs that resemble financial indicators, so the functions chosen are similar to the basic components used in those traditional indicators, such as comparison (greater than, less than, min, max), basic arithmetic operators, and the mean function. We also include functions that use none of their inputs at all, but instead have a fixed constant output such as 1 or -1. These resemble “bias” nodes from neural networks and often have an impact on the performance of CGP. The complete list of functions used in this paper are given in Table 1.

Once the parameters and *Functions* table have been specified, the next step is to give the algorithm a value function *Fitness()* with which to evaluate each individual program. The basic CGP algorithm can then proceed, and it does so as a simple $1 + \lambda$ evolutionary strategy (Miller, 2011) where $\lambda = 4$. In

Table 1: Functions used to construct individual programs. Functions either take two inputs x and y , or they ignore the inputs and produce a constant value.

Function	Description
+	Returns $x + y$
-	Returns $x - y$
\times	Returns $x * y$
/	Returns x/y , or 1.0 on divide-by-zero error
>	Returns 1 if $x > y$, -1 otherwise
<	Returns 1 if $x < y$, -1 otherwise
MAX	Returns $\max(x, y)$
MIN	Returns $\min(x, y)$
MEAN	Returns $(x + y)/2$
C_1	Returns 1
C_{-1}	Returns -1
C_0	Returns 0

other words, the search starts from randomly generated programs, and proceeds generationally. In each generation, only the best program is retained and the others in the population are replaced by mutated offspring of the best program. There is no crossover in canonical CGP.

One interesting facet of the algorithm that differentiates CGP from other evolutionary algorithms is its method of selecting the parent for the next generation. In CGP, an offspring replaces its parent if its fitness is greater than *or equal to* its parent’s fitness. That is, even if there is no improvement in fitness, the search algorithm can adopt a new “best program” and parent for the next generation as long as the offspring’s fitness is at least equal to its parents. This mechanism permits neutral mutations that allow for genetic drift, a feature that adds random diversity into the population without any cost. It has been shown previously such diversity significantly improves the search performance of CGP.

An example of a program evolved using CGP in the experiments reported here is shown in Figure 1.

This example illustrates the phenomenon of non-coding regions in CGP quite well. Each node/function call in a program may be coding or non-coding. If a node is defined as coding, this means that it is connected to the inputs either directly or indirectly. In the case of indirect connection, the connection is via the outputs of another function. Furthermore, in order to be a coding node, the node’s own output must *also* be used to compute the final outputs of the program. Any other function nodes are essentially useless and constitute “junk” regions of the genotype. In Figure 1, the example program has 7 coding and 23 non-coding function nodes. Note that CGP programs are directed acyclic graphs as opposed to trees or linear sequences of instructions.

2.2 Financial Concepts

The main financial trading concepts will be explained briefly in this subsection.

The first important concept to understand is the notion that assets can be *bought* and *sold* as well as *sold short*. Short selling is different from selling an asset that you already own, because rather than reducing your (positive) quantity of the asset by selling it, you actually sell your asset first (i.e. acquire a negative quantity of the asset) and gamble that the price will go down so that you can buy it back later

at a lower price, thus making a profit. Short selling is therefore the opposite of normal “long” buying and selling. In all of our experiments we assume that a trading strategy can both buy long and sell short, and that a trade (buying or selling) is closed with the opposite action.

Another important concept to understand is the way that trading strategies are evaluated. Whereas normal machine learning classifiers are evaluated via standard measures such as accuracy or ROC, in finance these concepts have very little relevance if the strategy’s financial performance is also not considered. For example, a strategy with a 60% accuracy rate in picking direction will consistently lose money if its average loss per trade in dollar terms is twice its average win, even though the accuracy is greater than random.

We therefore utilise in this research the following three measures of a trading strategy’s performance: *cumulative return*, i.e. the sum of the consecutive small wins and losses that a strategy makes over its testing period; *maximum drawdown*, which is defined as the maximum drop in cumulative return over the same period; and *recovery factor*, which is defined as the ratio of the first of these quantities to the second.

To illustrate, suppose that a strategy yields a profit of \$50 in the first week, but loses it all plus a further \$25 in the second week (yielding a balance of \$-25). In the third week, the strategy earns \$35 profit, thus ending the three weeks with a \$10 profit. The cumulative return in this case is \$10; the maximum drawdown is \$75; and the recovery factor is $\frac{\$10}{\$75}$ or 0.133.

Note that the recovery factor essentially normalises the return against maximum drawdown; strategies with both high returns and drawdowns should yield the same recovery factor as those with low returns but correspondingly low maximum drawdowns. A negative recovery factor indicates that the strategy made a loss, while a recovery factor of less than 1 indicates that the strategy’s drawdown was greater than its eventual profit. Strategies with a recovery factors of 1 or more are therefore desirable.

3 Experimental Setup

In this section, the datasets used in the experiments are described. We then move on to outlining the way in which CGP programs were evaluated for fitness estimation purposes.

3.1 Datasets

Four datasets from two different major markets were utilised in our evaluation of CGP for trading strategy learning. The two markets selected were deliberately chosen because they are highly liquid, meaning that there is simply a larger number of traders. The “herding behaviour” of the crowd may therefore more easily become apparent in these markets. Smaller markets, on the other hand, are less liquid and therefore more prone to sudden large price movements arising from single trades and other such noise. The two markets that we chose are quite disparate in order to ensure that our approach was tested rigorously.

The chosen markets were (i) the market for US currency, as determined by the Euro/US dollar exchange rate, and (ii) the US share market, as measured by the Dow Jones Industrial Average. Both markets have quite different characteristics. We also chose two quite distinct time-periods from their market price series, namely pre-recession 2006 and post-

recession 2010. The two time periods combined with the two markets yielded four datasets.

Each dataset consisted of three month’s worth of data, of which the first two months were used for training and the last month for out-of-sample testing. The exact dates and details of the datasets are given in Table 2.

The data we used is available from a financial data firm, Pi Trading¹ and comes in the form of an EST time-stamped series of open, high, low and close prices for every minute that a market is open. There are no records for minute bars where there are no transactions (i.e. where the open, low, high and close values are identical), so the actual number of records in the dataset is less than the number of minutes that the markets were open for. For the exchange rate data, this amounts to about 80,000 minute records in both the 2006 and 2010 periods, and for the Dow Jones data (which is open during US business hours only) this comprises approximately 25,000 records.

3.2 Trading Simulation using CGP Programs

In order to evaluate a trading strategy with historical data, it must be simulated. However, a simulation of a trading strategy can only ever be a rough approximation, simply because real trading has many other factors that are beyond the scope of a simulation. For example, brokers usually charge transaction costs on trades, but the charging scheme may vary from broker to broker and across time. Likewise, live data may contain errors that are subsequently cleaned in historical datasets. Historical data also does not contain information about slippage and other order filling problems. In the simulations described here, we assume no transaction costs and that there are no complications with order filling such as slippage or incorrect prices.

Given the assumptions, each CGP program was evaluated in the following way. The data (either the in-sample split during learning or the out-of-sample split during testing) was divided into days. It was assumed that each strategy would make one trade per day, at the start of the day, and that the trade would remain open until the last minute of same day. At that point it would be closed and the cumulative return or loss of the strategy updated. We do not simulate position sizes in these experiments – instead, the cumulative return is measured in points, which are a standard unit for measuring market prices. In the Euro/US dollar market, the standard point size is 0.0001, whereas for the Dow it is 0.01. This method of recording performance is ideal because it is independent of the size of the trades, which depends on many other factors (such as whether the amount invested is fixed or compounding, etc).

How does the CGP program decide which action (buy or sell) to take? Refer again to Figure 1. Each program has a single output node for each program, which if positive indicates a buy or long position for the following day, and if negative, indicates a sell or short position. There are seven inputs for each program corresponding to the closing prices of minute bars during the day prior to the trade. The exact minute bars are -1 (i.e. the closing price of the immediately previous day), -60 (the price 60 minute bars ago), -120, -180, -240, -300, and -360. Note that we skip minutes bars for which there is no trading activity or price change. These closing prices are mapped onto the input variables for the program, namely i_1 , i_2 , etc, which Figure 1 depicts as an example. The

¹<http://pitradings.com/>, data obtained 2011

Table 2: Datasets used in the experiments (EURUSD=Euro/US dollar exchange rate; INDU=Dow Jones Industrial Average).

Dataset	In-Sample Period	Out-of-Sample Period	Out-of-Sample Size
EURUSD ₁	1/5/2006 - 31/6/2006	2/7/2006 - 31/7/2006	26 days
EURUSD ₂	3/1/2010 - 28/2/2010	1/3/2010 - 30/3/2010	27 days
INDU ₁	1/5/2006 - 31/6/2006	2/7/2006 - 31/7/2006	20 days
INDU ₂	3/1/2010 - 28/2/2010	1/3/2010 - 30/3/2010	23 days

Table 3: Parameters used by the canonical CGP algorithm.

Parameter	Value
n_{in}	7
n_{out}	1
n_l	30
n_m	6
$popsiz$	5
$maxgens$	100,000

inputs are thus a sample of the prices that occurred during the day leading up to the trade.

Besides the number of inputs and outputs, CGP also has a number of other parameters that must be specified. During initial experiments, we discovered that setting *maxgens* to a very high value such as 10,000,000 (as suggested in some references) resulted in programs that grossly overfitted the training data and therefore performed poorly on out-of-sample data. We therefore reduced the number of generations to 100,000 and obtained far better results.

We also found that a relatively high mutation was effective. In our setup, the total number of alleles is 91 (that being 30 function nodes plus 2×30 inputs per node plus 1 output node specification). We set the mutation rate $n_m=6$, which corresponds to approximately 6.5% of the alleles. Although this is higher than the recommended mutation rate (Miller, 2011), it resulted in better performance than a lower mutation rate. A summary table of the CGP parameter settings used in our experiments are shown in Table 3.

Finally, because CGP is a randomised algorithm, it is insufficient to run CGP only once per training/testing dataset and expect the results to be significant statistically. Instead, we repeated each experiment 100 times (i.e. we perform 100 independent trials per train/test split) and calculated the average and standard error of each of the three performance measures. We then used these values to calculate the 99% upper and lower confidence limits for each measure.

4 Results

In this section, we report on the results of our experiments and examine the types of program that CGP evolves for trading.

4.1 CGP Trading Strategy Performance

Before considering how strategies learned using CGP performed on the out-of-sample data, it is prudent to firstly consider how simplistic strategies perform. The most commonly used baseline method in trading strategies research is the buy and hold strategy; the

Table 4: Out-of-Sample Returns for Simple Positive and Negative Strategies, expressed in market points (0.0001 for EURUSD and 0.01 for INDU).

Dataset	Rtn(Pos)	Rtn(Neg)
EURUSD ₁	-0.0077	0.0077
EURUSD ₂	-0.0192	0.0192
INDU ₁	-58	58
INDU ₂	280	-280

equivalent of this in our context is a strategy that buys every day, which we refer to as a positive simple strategy. The opposite strategy to this is the sell everyday strategy, or negative simple strategy. We simulated these simple strategies and calculated the returns (in cumulative points) over the test period, which are given in Table 4.

Because these simplistic strategies are essentially opposites, one simplistic strategy is likely to make a profit and the other will make an equivalent loss, as the table demonstrates. The main problem in actually applying these simplistic strategies is deciding which one to take. As the table shows, if a unilateral decision were taken to follow the positive simple strategy (i.e. just buy every day), then a loss would have been incurred in three out of the four out-of-sample market periods.

Having covered the simple baselines, we now turn to the performance of CGP for trading strategy learning.

We assessed three different value/fitness measures. In each case, the objective of evolution was to find an individual that maximized the measure. The measures were: total cumulative return (i.e. net profit); negative maximum drawdown (negating drawdown makes small drawdowns more desirable); and the recovery factor.

CGP was run 100 times on each of the four in-sample datasets using one of each of the three different fitness measures just described. This yielded a total of $100 \times 4 \times 3 = 100 \times 12$ individual CGP runs. The in-sample best-of-run individual program was then tested on the corresponding out-of-sample data, and the average and standard error of the performance over 100 runs per combination of dataset/measure was calculated. We also calculated the 99% upper and lower confidence bounds for the average (which by definition is 2.58 standard errors above and below the sample mean). The results are given in Tables 5-8.

Examining these tables, we can make a number of observations.

Firstly, consider the recovery factor. Recovery factor is a ratio and therefore comparable across all markets despite their different units and different characteristics such as volatility. In every case, the average recovery factor is positive. Furthermore, in terms of statistical significance, the lower 99% confidence

Table 5: Out-of-Sample results for EURUSD₁ using three different in-sample optimization methods, 100 independent runs per method.

	Return Opt.			Negative	MaxDD Opt.		Recovery Opt.		
	-MaxDD	Return	Rec.		-MaxDD	Return	-MaxDD	Return	Rec
Avg	0.0216	0.0173	1.10	0.0263	0.0099	0.83	0.0226	0.0180	1.19
StdErr	0.0007	0.0016	0.13	0.0010	0.0023	0.17	0.0009	0.0018	0.17
Upper	0.0234	0.0215	1.43	0.0288	0.0158	1.26	0.0248	0.0227	1.64
Lower	0.0199	0.0130	0.78	0.0238	0.0039	0.41	0.0203	0.0133	0.75

 Table 6: Out-of-Sample results for EURUSD₂ using three different in-sample optimization methods, 100 independent runs per method.

	Return Opt.			Negative	MaxDD Opt.		Recovery Opt.		
	-MaxDD	Return	Rec.		-MaxDD	Return	-MaxDD	Return	Rec
Avg	0.0374	0.0006	0.36	0.0344	0.0048	0.61	0.0263	0.0099	0.83
StdErr	0.0015	0.0028	0.10	0.0015	0.0031	0.13	0.0010	0.0023	0.17
Upper	0.0413	0.0078	0.63	0.0383	0.0128	0.96	0.0288	0.0158	1.26
Lower	0.0335	-0.0066	0.10	0.0304	-0.0032	0.27	0.0238	0.0039	0.41

bound on recovery factor, in all but one case, is also positive. This is a strong indication that the CGP method is effective.

However, the mean recovery factor is not always more than 1.0, which is desirable. For the 2006 EURUSD dataset, the average recovery factor is around 1.0, but it is much lower in the 2010 EURUSD dataset and the 2006 Dow Jones dataset. Surprisingly, the recovery factor is on average greater than 1.0 for the 2010 Dow Jones dataset.

The second result we will consider is the average return. Again, examining the tables, we see that while the returns are positive, they are often quite modest. For example, in the EURUSD 2006 dataset result shown in Table 5 the best return is 0.0031 or 31 points, which would only be significantly profitable if a significant investment was made (for a standard lot size of \$100,000, this would amount to about \$31 profit.) However, the simple negative strategy results shown in Table 4 are also quite modest at only 77 points, indicating that the market did not move far during the testing period.

Where the market did move a significant amount (for example, the Dow Jones 2010 dataset where the simple positive strategy records a \$280 profit), the CGP strategies capture a significant chunk of that movement – a little over half of it with a net return of \$169.35 on average.

Which of the three optimization measure is optimal in the experiments? An examination of the results shows that for the Euro/US dollar datasets, it is optimization of recovery factor that leads to the best on-average cumulative returns (those being 0.0180 and 0.099 for the 2006 and 2010 datasets respectively). For the Dow Jones datasets, optimizing negative maximum drawdown leads to the best cumulative returns.

Interestingly, in none of the four experimental datasets does direct optimization for in-sample return lead to the best out-of-sample return. Additionally, optimization for return is the only strategy that leads to a negative out-of-sample return, that being \$-31.03 for the 2006 Dow Jones dataset. The lesson to be learned here seems to be that it is better to optimize for minimal drawdowns than it is to optimize directly for maximum return.

4.2 Analysis of Programs

In addition to the performance of CGP-based programs as trading strategies, we were also interested in the composition of the programs that were evolved. Figure 1 gives one specific example of a program that was evolved. To perform a more general analysis, we examined, for each of four datasets, the 100 best-of-run programs that were tested out-of-sample. Our analysis primarily concerned the frequency with which individual functions appeared in these programs. These frequencies are given in Table 9.

An examination of this table shows that by far the most frequently selected operator is the subtraction – operator, followed closely the comparison < and > operators, and then the *MEAN* and *MIN* operators. These are indeed the type of operators that one would expect to see if designing an indicator-based trading system. Interestingly, functions representing constant outputs (C_1 , C_{-1} , and C_0) are used very infrequently.

We also computed the average size of each best-of-run program for all four of the datasets. Those averages, in terms of the number of active nodes, are 6.16 and 6.21 for the EURUSD datasets and 4.58 and 5.34 for the INDU datasets. This shows that whereas programs of length up to 30 could have evolved, that many functions were not required and that the resulting programs were actually reasonably simple.

5 Conclusion

To conclude, an investigation of Cartesian Genetic Programming (CGP) with different objective functions for the purpose of learning trading strategies has been undertaken. CGP has been shown to be effective at learning strategies that often make a modest but significant net positive returns on data from two different markets and two different time periods. Furthermore, the method produces rules that are relatively simple, containing on average 5-6 functions per rule.

References

Barbosa R., Belo O. (2008) Autonomous Forex Trading Agents, in *Proc. 2008 International Conference*

Table 7: Out-of-Sample results for INDU₁ using three different in-sample optimization methods, 100 independent runs per method.

	Return Opt.			Negative MaxDD Opt.			Recovery Opt.		
	-MaxDD	Return	Rec.	-MaxDD	Return	Rec.	-MaxDD	Return	Rec.
Avg	494.32	-31.03	0.29	354.85	232.30	1.01	440.02	56.71	0.35
StdErr	18.30	33.85	0.14	11.58	39.42	0.14	13.46	26.78	0.10
Upper	541.54	56.29	0.65	384.73	334.00	1.39	474.75	125.80	0.61
Lower	447.10	-118.36	-0.07	324.97	130.60	0.64	405.30	-12.39	0.08

Table 8: Out-of-Sample results for INDU₂ using three different in-sample optimization methods, 100 independent runs per method.

	Return Opt.			Negative MaxDD Opt.			Recovery Opt.		
	-MaxDD	Return	Rec.	-MaxDD	Return	Rec.	-MaxDD	Return	Rec.
Avg	148.49	118.63	1.18	142.37	169.35	1.59	146.27	140.30	1.36
StdErr	5.55	13.68	0.14	4.32	11.09	0.21	5.67	14.03	0.16
Upper	162.80	153.92	1.55	153.52	197.95	2.14	160.89	176.49	1.76
Lower	134.18	83.34	0.81	131.22	140.74	1.04	131.66	104.11	0.96

on Data Mining, ICDM 2008, P. Perner Ed., LNAI 5077, pp. 389-403.

Beechey M., Gruen D., Vickrey J. (2000). *The Efficient Markets Hypothesis: A Survey*. Reserve Bank of Australia.

Contreras I., Hidalgo J., Nunez-Letamendia L. (2012), A GA combining technical and fundamental analysis for trading the stock marking. In *EvoApplications 2012*, Springer, pp.. 174-183.

Fama, E. (1970), Efficient Capital Markets: A Review of Theory and Empirical Work. *Journal of Finance* 25 (2): 383-417. doi:10.2307/2325486. JSTOR 2325486.

Hirabayashi A., Aranha C., Iba H. (2009), Optimization of the Trading Rule in Foreign Exchange using Genetic Algorithm, in *Proc. GECCO'09*, pp. 1529-1536.

Larkin F. and Ryan C. (2010), Modesty is the Best Policy: Automatic Discovery of Viable Forecasting Goals in Financial Data. In *Proc. EvoApplications 2010*, Part II, pp. 202-211.

Lean Y., Lai K. (2007), *Foreign Exchange Rate Forecasting with Artificial Neural Networks*. Springer-Verlag.

Liu Z., Xiu D. (2009), An automated trading system with multi-indicator fusion based on D-S evidence theory in forex market, in *Proc. Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, IEEE, pp. 239-243.

Miller J., ed. (2011), *Cartesian Genetic Programming*. Springer.

Ni H., Yin H. (2009), Exchange rate prediction using hybrid neural networks and trading indicators, *Neurocomputing* 72:2815-2832.

Sekanina L., Harding S., Banzhaf W., Kowaliw T. (2011), Image Processing and CGP. In Miller (2011), pp. 181-216.

Sekanina L., Walker J., Kaufmann P., Platzner M. (2011), Evolution of Electronic Circuits. In Miller (2011), pp. 125-180.

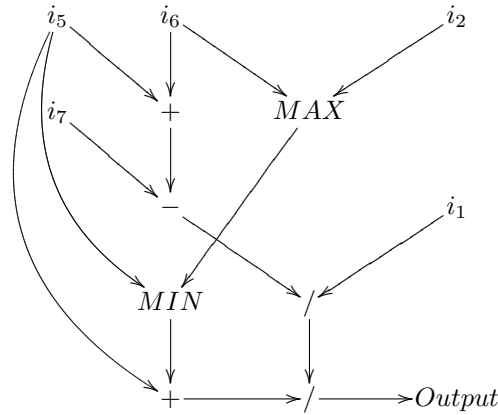
Sewell, M. (2011), *Characterization of Financial Time Series*. Research Note RN/11/01, Dept. of Computer Science UCL.

Table 9: Percentage probability of a function being selected for a node in a best-of-run individual by dataset, over 100 independent runs per dataset.

Function	EURUSD ₁	EURUSD ₂	INDU ₁	INDU ₂
+	6.15%	4.94%	9.29%	6.31%
−	15.05%	10.37%	15.55%	11.87%
×	8.58%	7.97%	6.91%	7.61%
/	11.33%	8.13%	8.86%	11.50%
>	11.97%	16.91%	12.96%	11.32%
<	12.30%	13.24%	12.53%	12.80%
MAX	10.36%	9.89%	5.40%	8.16%
MIN	8.25%	12.92%	8.86%	14.66%
MEAN	12.14%	7.97%	13.61%	10.39%
C ₁	1.46%	1.91%	2.38%	2.97%
C _{−1}	1.29%	2.71%	1.94%	1.30%
C ₀	1.13%	3.03%	1.73%	1.11%

 Figure 1: An example of a program evolved using CGP. Function node 17 is the output node. Each function call has two inputs. Inputs may be input data (denoted by i_0, i_1 , etc) or the output of another function node (denoted by an integer identifying the node). In the array representation in figure (a), active nodes are marked by marked by *. Figure (b) is the corresponding evaluation graph.

Node	F	Inputs	Node	F	Inputs	Node	F	Inputs
0*	+	i_5, i_6	10	−	6, i_1	20	MAX	i_4, i_7
1*	−	$i_7, 0$	11	MIN	2, 6	21	×	3, 5
2*	MAX	i_2, i_6	12*	/	1, i_1	22	MIN	$i_3, 10$
3	+	$i_7, 0$	13	−	10, 10	23	MEAN	4, 3
4	C ₁	i_5, i_4	14	MAX	5, i_6	24	MEAN	3, i_5
5	×	$i_5, 2$	15*	+	$i_5, 9$	25	<	8, 15
6	MAX	0, 4	16	/	$i_1, 2$	26	×	0, 4
7	×	$i_7, 0$	17*	/	15, 12	27	>	$i_1, 24$
8	C ₁	1, 2	18	MEAN	$i_3, 11$	28	+	8, 11
9*	MIN	$i_5, 2$	19	>	8, 3	29	+	4, 2



A Comparative Study of MRI Data using Various Machine Learning and Pattern Recognition Algorithms to Detect Brain Abnormalities

¹Lavneet Singh, ²Girija Chetty

^{1,2}Faculty of Information Sciences and Engineering
University of Canberra, Australia
Lavneet.singh@canberra.edu.au
Girija.chetty@canberra.edu.au

Abstract

In this study, we present the investigations being pursued in our research laboratory on magnetic resonance images (MRI) of various states of brain by extracting the most significant features, and to classify them into normal and abnormal brain images. We propose a novel method based on deep and extreme machine learning on wavelet transform to initially decompose the images, and then use various features selection and search algorithms to extract the most significant features of brain from the MRI images. By using a comparative study with different classifiers to detect the abnormality of brain images from publicly available neuro-imaging dataset, we found that a principled approach involving wavelet based feature extraction, followed by selection of most significant features using PCA technique, and the classification using deep and extreme machine learning based classifiers results in a significant improvement in accuracy and faster training and testing time as compared to previously reported studies.

Keywords: -Deep Machine Learning, Extreme Machine Learning, MRI, PCA

1. INTRODUCTION

Magnetic Resonance Images (MRI) is an advance technique used for medical imaging and clinical medicine and an effective tool to study the various states of human brain. MRI images provide the rich information of various states of brain which can be used to study, diagnose and carry out unparalleled clinical

analysis of brain to find out if the brain is normal or abnormal. However, the data extracted from the images is very large and it is hard to make a conclusive diagnosis based on such raw data. In such cases, we need to use various image analysis tools to analyze the MRI images and to extract conclusive information to classify into normal or abnormalities of brain. The level of detail in MRI images is increasing rapidly with availability of 2-D and 3-D images of various organs inside the body.

Magnetic resonance imaging (MRI) is often the medical imaging method of choice when soft tissue delineation is necessary. This is especially true for any attempt to classify brain tissues (Fletcher-Heath, L. M., Hall, L. O., Goldgof, D. B. and Murtagh, F.R. 2001). The most important advantage of MR imaging is that it is non-invasive technique (Chaplot, S., Patnaik, L.M. and Jagannathan N.R. 2006). The use of computer technology in medical decision support is now widespread and pervasive across a wide range of medical area, such as cancer research, gastroenterology, heart diseases, brain tumors etc. (Gorunescu, F. 2007, Kara, S. and Dirgenali, F. 2007). Fully automatic normal and diseased human brain classification from magnetic resonance images (MRI) is of great importance for research and clinical studies. Recent work (Chaplot, S., Patnaik, L.M. and Jagannathan N.R. 2006, Maitra, M. and Chatterjee A. 2007) has shown that classification of human brain in magnetic resonance (MR) images is possible via machine learning and classification techniques such as artificial neural networks and support vector machine (SVM) (Chaplot, S., Patnaik, L.M. and Jagannathan N.R. 2006, Mishra, Anurag, Singh, Lavneet and Chetty, Girija 2012), and unsupervised techniques such as self-organization maps (SOM) (Chaplot, S., Patnaik, L.M. and Jagannathan N.R. 2006, Singh, Lavneet and Chetty, Girija. 2012) and fuzzy c-means combined with appropriate feature extraction techniques (Maitra, M. and Chatterjee A. 2007). Other supervised classification techniques, such as k-nearest neighbors (k-NN), which group pixels

¹ Copyright (c) 2012, Australian Computer Society, Inc. This paper appeared at the 10th Australasian Data Mining Conference (AusDM 2012), Sydney, Australia, December 2012. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 134, Yanchang Zhao, Jiuyong Li, Paul Kennedy, and Peter Christen, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

based on their similarities in each feature image (Fletcher-Heath, L. M., Hall, L. O., Goldgof, D. B. and Murtagh, F.R. 2001, Abdolmaleki, P., Mihara, F., Masuda, K. and DansoBuadu, Lawrence. (1997), Rosenbaum, T., Engelbrecht, V., Krolls, W. and Lenard, H. 1999, Cocosco, C., Zijdenbos, Alex P. and Evans, Alan C. 2003) can be used to classify the normal/pathological T2-weighted MRI images.

Out of several debilitating ageing related health conditions, white matter lesions (WMLs) are commonly detected in elders and in patients with multiple brain abnormalities like Alzheimer's disease, Huntington's disease and other neurological disorders. According to previous studies, it is believed that total volume of the lesions (lesion load) and their progression relate to the aging process as well as disease process. Therefore, segmentation and quantification of white matter lesions via texture analysis is very important in understanding the impact of aging and diagnosis of various brain abnormalities. Manual segmentation of WM lesions, which is still used in clinical practices, shows the limitation to differentiate brain abnormalities using human visual abilities. Such methods can produce a high risk of misinterpretation and can also contribute to variation in correct classification. Automated texture analysis algorithms have been developed to detect brain abnormalities using image segmentation techniques and machine learning algorithms. The signal of homogeneity and heterogeneity of abnormal areas in Region of Interest (ROI) in white matter lesions of brain in T2-MRI images can be quantified by texture analysis algorithms [reference]. The ability to measure small differences in MRI images is essential and important to reduce the diagnosis errors of brain abnormalities. The supervised feature classification from T2 MRI images, however, suffers from two problems. First, because of the large variability in image appearance between different datasets, the classifiers need to be retrained from each data source to achieve good performances. Second, these types of algorithms rely on manually labeled training datasets to compute the multi-spectral intensity distribution of the white matter lesions making the classification unreliable. Inspired by new segmentation algorithms in computer vision and machine learning, we propose an efficient semi-automatic and deep learning algorithm for white matter (WM) lesion segmentation around ROI based on extreme and deep machine learning. Further, we compare this novel approach with some of the other supervised machine learning techniques reported previously.

Rest of the paper is organized as follows. Next Section gives a brief background of materials and methods used in Section 2. The details of the feature extraction, and feature selection, and other classifiers techniques used is described in same Section 2, 3 and Section 4 presents some of the experimental work carried. The paper concludes with in section 5 with some outcomes of the experimental work using proposed approach, and outlines plans for future work.

2. Materials and Methods

2.1 Datasets

The input dataset consists of axial, T2-weighted, 256 X 256 pixel MR brain images (Fig. 1). These images were downloaded from the (Harvard Medical School website ([http:// med.harvard.edu/AANLIB/](http://med.harvard.edu/AANLIB/), Harvard Medical School 1999). Only those sections of the brain in which lateral ventricles are clearly seen are considered in our study. The number of MR brain images in the input dataset is 60 of which 6 are of normal brain and 54 are of abnormal brain. The abnormal brain image set consists of images of brain affected by Alzheimer's and other diseases. The remarkable feature of a normal human brain is the symmetry that it exhibits in the axial and coronal images. Asymmetry in an axial MR brain image strongly indicates abnormality. Hence symmetry in axial MRI images is an important feature that needs to be considered in deciding whether the MR image at hand is of a normal or an abnormal brain. A normal and an abnormal T2-weighted MRI brain image are shown in Fig. 1(a) and 1(b), respectively. Indeed, for multilayer learning models like deep and extreme machine learning algorithms needed big datasets for training, however due to lack of availability of proper datasets in MRI imaging, we used this dataset for examining the performance of proposed approaches for this paper, but acquiring other suitable datasets for future studies.

2.2 Coarse Image Segmentation

Color image segmentation is useful in many applications. From the segmentation results, it is possible to identify regions of interest and objects in the scene, which is very beneficial to the subsequent image analysis or annotation. However, due to the difficult nature of the problem, there are few automatic algorithms that can work well on a large variety of data. The problem of segmentation is difficult because of image texture. If an image contains only homogeneous color regions, clustering methods in color space are sufficient to handle the problem. In reality, natural scenes are rich in color and texture. It is difficult to identify image regions containing color-texture patterns. The approach taken in this work assumes the following:

- Each region in the image contains a uniformly distributed color-texture pattern.
- The color information in each image region can be represented by a few quantized colors, which is true for most color images of natural scenes.
- The colors between two neighboring regions are distinguishable - a basic assumption of any color image segmentation algorithm.

K-Means clustering based Coarse Image Segmentation

K-Means clustering algorithm is a well-known unsupervised clustering technique to classify any given input dataset. This algorithm classifies a given dataset into discrete k-clusters using which k-centroids are defined, one for each cluster. The next step is to take each

point in the given input data set and associate it to the possible nearest centroid. This process is repeated for all the input data points, based on which next level of clustering and the respective centroids are obtained. This procedure is iterated until it converges. This algorithm minimizes the following objective function.

$$J = \sum_{j=1}^k \sum_{i=1}^k \|x_i^j - c^j\|^2$$

Where $\|x_i^j - c^j\|^2$ is a chosen distance measure between a data point $(x_i)^j$ and the cluster centre, c_j is an indicator of the distance of the k data points from their respective cluster centers. The proposed unsupervised segmentation algorithm uses the principle of K-means clustering.

The proposed technique segments the region of interest (ROI) of an input image (input_img) by an interactive user defined shape of square or rectangle to obtain select_img. Then, the number of bins for coarse data computation (bin size), the size of overlapping kernel to partition (w-size) and the maximum number of clusters for segmentation (max_class) are fed as input data for the computation of coarse data. The coarse data identified by each kernel is aggregated to form the final_coarse_data which is further clustered using the principle of K-means clustering in order to produce the segment_img. The algorithmic description of the proposed technique is given herein under:

Algorithm

1. Read a grayscale image as input_img
- /* Define the area to be segmented as a runtime interactive input. The shape of the selection can either be a square or a rectangle */
2. Let select_img is the selected subimage of input_img
3. Assign:
 - a. binsize=5
 - /* number of bins for coarse data computation */
 - b. wsize= 7
 - /* wsize is the size of overlapping kernel to partition the select_img */
 - c. max_class= 3
 - /* maximum number of clusters for segmentation */
4. Repeat step 5 and 6 in algorithm until the select_img is read
5. Read select_img in the order of (wsize*wsize) as window_img
6. Compute coarse_img for window_img as coarse_win_data
7. Aggregate coarse_win_data for select_img as final_coarse_data
8. Cluster final_coarse_data using K-means clustering technique using max_class in order to obtain segment_img
9. Stop

This algorithm can segment an object either fully or partially based on user's choice. If the image has a background and object(s) then it partitions the object from the background and displays its coarse image. If the image has no background, then the segmented image reveals the inner details of the object.

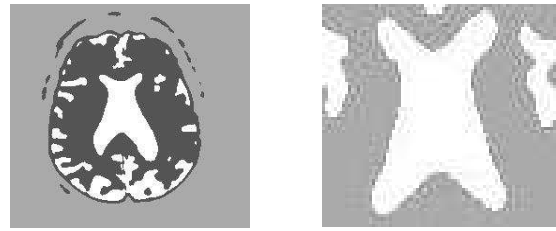


Figure 1. (a) Coarse Segmented MRI Image based on above algorithm (b) ROI segmented image of White Lesions

3. Decomposition of images Using Wavelets

Wavelets are mathematical functions that decompose data into different frequency components and then study each component with a resolution matched to its scale. Wavelets have emerged as powerful new mathematical tools for analysis of complex datasets. The Fourier transform provides representation of an image based only on its frequency content. Hence this representation is not spatially localized while wavelet functions are localized in space.

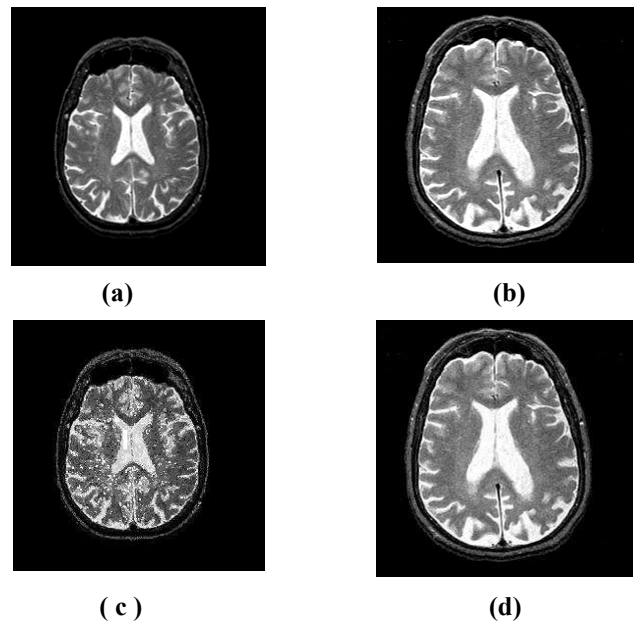


Fig.2. (a) T2, weighted an axial MRI Brain Image; (b) T2, weighted an axial MR brain image as abnormal brain; (c) and (d) T2, weighted an axial MR brain image as normal and abnormal brain after Wavelets Decomposition and denoising

Discrete wavelets transform (DWT)

The DWT is an implementation of the wavelet transform using a discrete set of the wavelet scales and translation obeying some defined rules. For practical computations, it is necessary to discretize the wavelet transform. The scale parameters are discretized on a logarithmic grid. The translation parameter (τ) is then discretized with respect to the scale parameter, i.e. sampling is done on the dyadic (as the base of the

logarithm is usually chosen as two) sampling grid. The discretized scale and translation parameters are given by, $s = 2^{-m}$ and $t = n2^{-m}$, where $m, n \in \mathbb{Z}$, the set of all integers. Thus, the family of wavelet functions is represented in Eq. (1) and (2),

$$\psi_{m,n}(t) = 2^{\frac{m}{2}} \psi(2^m t - n) \quad (1)$$

$$W\psi(a, b) = \int_{-\infty}^{\infty} f(x) * \psi_{a,b}(t) dx \quad (2)$$

In case of images, the DWT is applied to each dimension separately. This result in an image Y is decomposed into a first level approximation component Y_a^1 and detailed components Y_h^1 , Y_v^1 and Y_d^1 corresponding to horizontal, vertical and diagonal details. Fig.1 depicts the process of an image being decomposed into approximate and detailed components.

The approximation component (Y_a) contains low frequency components of the image while the detailed components (Y_h , Y_v and Y_d) contain high frequency components. Thus,

$$Y = Y_a^1 + \{ Y_h^1 + Y_v^1 + Y_d^1 \} \quad (3)$$

At each decomposition level, the length of the decomposed signals is half the length of the signal in the previous stage. Hence the size of the approximation component obtained from the first level decomposition of an $N \times N$ image is $N/2 \times N/2$, second level is $N/4 \times N/4$ and so on. As the level of decomposition is increased, compact but coarser approximation of the image is obtained. Thus, wavelets provide a simple hierarchical framework for interpreting the image information.

4. Deep Belief Nets

DBNs (Hinton, G.E. and Salakhutdinov, R.R. 2006) are multilayer, stochastic generative models that are created by learning a stack of Restricted Boltzmann Machines (RBMs), each of which is trained by using the hidden activities of the previous RBM as its training data. Each time a new RBM is added to the stack, the new DBN has a better variation lower bound on the log probability of the data than the previous DBN, provided the new RBM is learned in the appropriate way (Hinton, G.E. and Osindero, S. 2006).

A Restricted Boltzmann Machine (RBMs) is a complete bipartite undirected probabilistic graphical model. The nodes in the two partitions are referred as hidden and visible units. An RBM is defined as

$$p(v, h) = \frac{e^{-E(v, h)}}{\sum_u \sum_g e^{-E(u, g)}} \quad (4)$$

Where $v \in V$ are the visible nodes and $h \in H$ are the latent random variables. The energy function $E(v, h, W)$ is described as

$$E = - \sum_{i=1}^D \sum_{j=1}^K v_i W_{ij} h_j \quad (5)$$

Where $W \in \mathbb{R}^{D \times K}$ are the weights on the connections, and where we assume that the visible and hidden units both contain a node with value of 1 that acts to introduce bias. The conditional distribution for the binary visible and hidden units are defined as

$$p(v_i = 1/h, W) = \sigma(\sum_{j=1}^K W_{ij} h_j) \quad (6)$$

$$p(h_j = 1/v, W) = \sigma(\sum_{i=1}^D W_{ij} v_i) \quad (7)$$

Where σ is the sigmoid function. Using above equations, it easy to go back and forth between the layers of RBM. While training, it consists of some input to the RBM on the visible layer, and updating the weights and the biases such that $p(v)$ is high. In generalized way, in as set of C training cases $\{v^c \in \{1, \dots, C\}\}$, the objective is to maximize the average log probability defined as

$$\sum_{c=1}^C \log p(v^c) = \sum_{c=1}^C \log \frac{\sum_g e^{-E(v^c, g)}}{\sum_u \sum_g e^{-E(u, g)}} \quad (8)$$

The whole training process involves updating the weights with several numbers of epochs and the data is split in 20 batches which we take it randomly and the weights are update at the end of every batch. We use the binary representation of hidden units' activation pattern for classification and visualization. The activation of each hidden unit is defined as

$$f(x) = g(Wx + b) \quad (9)$$

Where $g(z) = 1/(1 + \exp(-z))$ is the logistic sigmoid function, applied component-wise to the vector z , W is a weight vector between visible nodes and hidden nodes and b is a bias. The autoencoder with N_h hidden nodes is trained and fine-tuned using back-propagation to minimize squared reconstruction error, with a term encouraging low average activation of the units.

5. Extreme Machine Learning

The Extreme Learning Machine (Lin, M.B., Huang, G.B., Saratchandran P. and Sudararajan N. 2005, Huang, G.B., Zhu, Q.Y. and Siew, C.K. 2006, Huang, G.B., Zhu, Q.Y. and Siew, C.K. 2002, Mishra, Anurag, Singh, Lavneet and Chetty, Girija 2012, Singh, Lavneet and Chetty, Girija 2012 is a Single hidden Layer Feed forward Neural Network (SLFN) architecture. Unlike traditional approaches such as Back Propagation (BP) algorithms which may face difficulties in manual tuning control parameters and local minima, the results obtained after ELM computation are extremely fast, have good accuracy and has a solution of a system of linear equations. For a given network architecture, ELM does not have any control parameters like stopping criteria, learning rate, learning epochs etc., and thus, the implementation of this network is very simple. The main concept behind this algorithm is that the input weights (linking the input layer to the hidden layer) and the hidden layer biases are randomly chosen based on

some continuous probability distribution function such as uniform probability distribution in our simulation model and the output weights (linking the hidden layer to the output layer) are then analytically calculated using a simple generalized inverse method known as Moore – Penrose generalized pseudo inverse (Serre, D. 2002).

Given a series of training samples $(x_i, y_i)_{i=1, 2 \dots N}$ and \hat{N} the number of hidden neurons where $x_i = (x_{i1}, \dots, x_{in}) \in \mathbb{R}^n$ and $y_i = (y_{i1}, \dots, y_{im}) \in \mathbb{R}^m$, the actual outputs of the single-hidden-layer feed forward neural network (SLFN) with activation function $g(x)$ for these N training data is mathematically modeled as

$$\sum_{k=1}^{\hat{N}} \beta_k g((w_k, x_i) + b_k) = 0_i, \forall i = 1, \dots, N \quad (10)$$

Where $w_k = (w_{k1}, \dots, w_{kn})$ is a weight vector connecting the k^{th} hidden neuron, $\beta_k = (\beta_{k1}, \dots, \beta_{km})$ is the hidden neuron. The weight vectors w_k are randomly chosen. The term (w_k, x_i) denotes the inner product of the vectors w_k and x_i and g is the activation function. The above N equations can be written as $H\beta = O$ and in practical applications \hat{N} is usually much less than the number N of training samples and $H\beta \neq Y$, where

$$H = \begin{bmatrix} g((w_1, x_1) + b_1) & \cdots & g((w_{\hat{N}}, x_1) + b_{\hat{N}}) \\ \vdots & \ddots & \vdots \\ g((w_1, x_{1N}) + b_1) & \cdots & g((w_{\hat{N}}, x_N) + b_{\hat{N}}) \end{bmatrix}_{N \times \hat{N}}$$

$$\beta = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_{\hat{N}} \end{bmatrix}_{\hat{N} \times m} \quad O = \begin{bmatrix} O_1 \\ \vdots \\ O_N \end{bmatrix}_{N \times m} \quad Y = \begin{bmatrix} Y O_1 \\ \vdots \\ Y O_N \end{bmatrix}_{N \times m} \quad (11)$$

The matrix H is called the hidden layer output matrix. For fixed input weights $w_k = (w_{k1}, \dots, w_{kn})$ and hidden layer biases b_k , we get the least-squares solution $\hat{\beta}$ of the linear system of equation $H\beta = Y$ with minimum norm of output weights β , which gives a good generalization performance. The resulting $\hat{\beta}$ is given by $\hat{\beta} = H^+ Y$ where matrix H^+ is the Moore-Penrose generalized inverse of matrix H (Serre, D. 2002).

6. Trained Classifiers

In this study, apart from deep learning based on Restricted Boltzmann machines and extreme machine learning based on Single hidden Layer Feed forward Neural Network (SLFN) architecture as classifiers, several other classifiers are also examined in terms of accuracy and performance, including K-nearest neighbor (Wang, Jun. and Zucker, Daniel J. 2000), SVM (Vapnik, V. 1995), Naive Bayes George, H. and John, Pat Langley. 1995), MultiboostAB (Webb, Geoffrey. 2000), RotationForest (Rodriguez, Juan J., Kuncheva, Ludmila I. and Alonso, Carlos J. 2006), VFI (Quinlan, Ross. 1993), J48 (Breiman, Leo. 2001) and Random Forest (Hall, M. A. 1998).

J48 (Kohavi, Ron and John, George H. 1997) is an implementation of C4.5 algorithm that produces

decision trees from a set of labeled training data using the concept of information entropy. It examines the normalized information gain (difference in entropy) that results from choosing an attribute for splitting the data into smaller subsets. To make the decision, the attribute with the highest normalized information gain is used. The KNN algorithm (Wang, Jun. and Zucker, Daniel J. 2000) compares the test sample with the available training samples and finds the ones that are more similar (“nearest”) to it. When the k -nearest training samples are found, the class label in majority is assigned to the new sample. Learning in the VFI algorithm (Quinlan, Ross. 1993) is achieved by constructing feature intervals around each class for each attribute (basically discretization) on each feature dimension. Class counts are recorded for each interval on each attribute and classification is performed by a voting scheme.

The Naïve Bayesian Classifier (George, H. and John, Pat Langley. 1995) assumes that features are independent. Given the observed feature values for an instance and the prior probabilities of classes, the a posteriori probability that an instance belongs to a class is estimated. The class prediction is the class with the highest estimated probability. The SVMs (Vapnik, V. 1995) first map the attribute vectors into a feature space (possibly with higher dimensions), either linearly or nonlinearly, according to the selected kernel function. Then, within this feature space, an optimized linear division is sought; i.e., a hyper plane is constructed which separates two classes (this can be extended to multiple classes). MultiBoosting (Webb, Geoffrey. 2000) is an extension to the highly successful AdaBoost technique for forming decision committees. MultiBoosting can be viewed as combining AdaBoost with wagging. It is able to harness both AdaBoost's high bias and variance reduction with wagging's superior variance reduction. Using C4.5 as the base learning algorithm, Multi-boosting is demonstrated to produce decision committees with lower error than either AdaBoost or wagging significantly more often than the reverse over a large representative cross-section of data set. It offers the further advantage over AdaBoost of suiting parallel execution.

7. Feature Selection

In machine learning, during the training of the classifiers, if the numbers of image features are large, it can lead to ill-posing and over fitting, and reduce the generalization of the classifier. One way to overcome this problem is to reduce the dimensionality of features. To reduce the dimensionality of the large set of features of dataset, in our study, we propose the use of three optimal attribute selection algorithms: correlation based feature selection (CFS) method (Kohavi, Ron and John, George H. 1997), which evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them, secondly an approach based on wrappers (Hughes, N.P. and Tarassenko, L. 2003) which evaluates attribute sets by using a learning scheme. Also in this study, three search methods are

also examined: the Best First, Greedy Stepwise and Scatter Search algorithms. These search algorithms are used with attribute selector's evaluators to process the greedy forward, backward and evolutionary search among attributes of significant and diverse subsets. In total, these feature selection algorithms were tested to select nearly 10 optimal and significant features out of 1024 features.

When we do PCA, we need to do an eigen-decomposition of the covariance matrix. The procedure of PCA is as follows:

1. Compute the mean:

$$\bar{x} = \frac{1}{m} \sum_{i=1}^m \tilde{x}_i$$

2. Generate the zero-mean data matrix:

$$x_i = \tilde{x}_i - \bar{x}$$

$$A = (x_1, x_2, \dots, x_m)$$

3. Construct the covariance matrix:

$$C = AA^T$$

The covariance matrix C is symmetric and positive definite. So the eigenvalues of C is real and non-negative.

4. Eigen-decomposition:

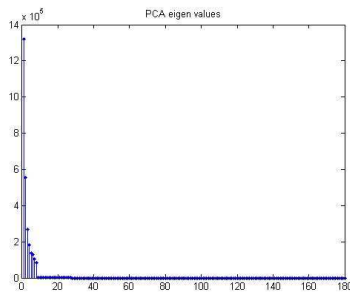
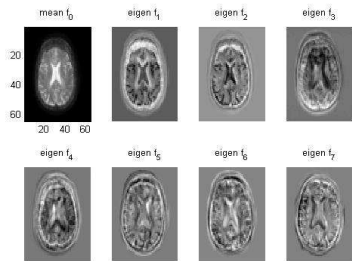
The eigenvalues λ_i and the eigenvectors v_i of C satisfy

$$Cv_i = \lambda_i v_i$$

5. So we have the eigen-decomposition of the covariance matrix:

$$C = V\Lambda V^{-1} = V\Lambda V^T$$

**Figure 3. (a) Eigen values of segmented MRI images
(b) Eigen vectors after PCA**



8. Experiments and Results

8.1 Level of wavelet decomposition

We obtained wavelet coefficients of 60 brain MR images, each of whose size is 256 X 256. Level-1 HAR wavelet decomposition of a brain MR image produces 16384 wavelet approximation coefficients; while level-2

and level-3 produce 4096 and 1024 coefficients, respectively. The third level of wavelet decomposition greatly reduces the input vector size but results in lower classification percentage. With the first level decomposition, the vector size (16384) is too large to be given as an input to a classifier. The preliminary experimental analysis of the wavelet coefficients through simulation in Matlab 7.10., we showed that level-2 features are the best suitable for different classifiers, whereas level-1 and level-3 features results in lower classification accuracy. The second level of wavelet decomposition not only gives virtually perfect results in the testing phase, but also has reasonably manageable number of features (4096) that can be handled without much hassle by the classifier. We also use the DAUB-4 (Daubachies) as mother wavelets to get decomposition coefficients of MRI images at Level 2 for comparative evaluation of two wavelets decomposition methods in terms of classification accuracy.

8.2 Attribute Selection and Classification

Table 1 shows the accuracy of classification (percentage of correctly classified samples), True Positive Rate (TP), False Positive Rate (FP) and Average Classification Accuracy (ACC) over all pair-wise combination with different feature evaluators and search algorithms with respect to multi-class classification.

Table 1 shows the performance of several learning classifiers, including K-nearest neighbor, SVM, Naive Bayes, MultiboostAB, Rotation Forest, VFI, J48 and Random Forest. Among the pair-wise classification, the lowest accuracy is observed for the classification VFI classifiers of 74.16% and the highest accuracy for the classification by Rotational forest of 97.06%. Moreover, the combination of CFS feature evaluator with the of Best First search algorithm gives the highest classification accuracy.

While Table 1 shows the performance of individual classifiers, Table 2 defines the comparative results of various combined search techniques and feature evaluators using above prescribed classifiers. Table 3 compares the proposed method against a popular dimensionality reduction method, known as Principal Component Analysis (PCA). PCA applies an orthogonal linear transformation that transforms data to a new coordinate system of uncorrelated variables called principal components. We have applied PCA to reduce the number of attributes or feature to 18 attributes and plotted the ROC curves using several above mentioned learning classifiers in terms of True Positive and False Positive Rate, as seen in figure 4. As can be seen in figure 4, ROC curves for all the trained learning classifiers examined in this study, the curves lie above the diagonal line describing the better classification rather than any other random classifiers. The optimal points of various trained classifiers are indicated by bold solid circles as False Positive rate (FP) and True Positive rate (TP). These optimal points in ROC curves

show the maximum optimal value (FP, TP) of all trained classifiers.

Table 1. Various Classifiers comparison with respect Average Classification Accuracy(%) and other parameters

Classifier	TP Rate	FP Rate	Precision	Recall	F-Measure	ACC (%)
KNN	0.935	0.917	0.826	0.853	0.839	91.04
SVM	0.912	0.912	0.831	0.912	0.87	91.17
Naive Bayes	0.868	0.916	0.828	0.868	0.847	86.76
MultiboostAB	0.91	0.91	0.829	0.91	0.868	91.04
Rotation Forest	0.971	0.285	0.971	0.971	0.968	97.06
VFI	0.742	0.049	0.93	0.742	0.796	74.16
J48	0.96	0.314	0.958	0.96	0.957	95.98
Random Forest	0.97	0.271	0.97	0.97	0.968	97.01

Table 2. Comparison of pair wise combination of various Attribute Selectors and classifiers with respect to ACC (%)

Evaluator	Search Algorithm	Classifier	N	ACC (%)
CFS	Best First	K-NN	6	91.04
CFS	Greedy Stepwise	K-NN	2	89.70
CFS	Scatter Search	K-NN	4	88.23
Wrapper	Best First	K-NN	5	89.32
Wrapper	Greedy Stepwise	K-NN	4	87.56
Wrapper	Scatter Search	K-NN	4	88.20
CFS	Best First	SVM	6	91.17
CFS	Greedy Stepwise	SVM	6	89.23
CFS	Scatter Search	SVM	4	91.04
Wrapper	Best First	SVM	2	90.65
Wrapper	Greedy Stepwise	SVM	2	90.65
Wrapper	Scatter Search	SVM	5	89.56
CFS	Best First	Naive Bayes	8	86.76
CFS	Greedy Stepwise	Naive Bayes	8	82.78
CFS	Scatter Search	Naive Bayes	7	82.12
Wrapper	Best First	Naive Bayes	4	85.44
Wrapper	Greedy Stepwise	Naive Bayes	2	85.44
Wrapper	Scatter Search	Naive Bayes	2	80.12
CFS	Best First	MultiboostAB	5	91.04
CFS	Greedy Stepwise	MultiboostAB	5	91.04
CFS	Scatter Search	MultiboostAB	4	86.54
Wrapper	Best First	MultiboostAB	5	89.39
Wrapper	Greedy Stepwise	MultiboostAB	5	90.45
Wrapper	Scatter	MultiboostAB	4	88.76

	Search			
CFS	Best First	Rotation Forest	9	97.06
CFS	Greedy Stepwise	Rotation Forest	9	96.21
CFS	Scatter Search	Rotation Forest	8	91.66
Wrapper	Best First	Rotation Forest	5	93.78
Wrapper	Greedy Stepwise	Rotation Forest	6	93.78
Wrapper	Scatter Search	Rotation Forest	6	89.54
CFS	Best First	VFI	3	74.16
CFS	Greedy Stepwise	VFI	2	71.01
CFS	Scatter Search	VFI	4	71.01
Wrapper	Best First	VFI	3	72.22
Wrapper	Greedy Stepwise	VFI	2	72.85
Wrapper	Scatter Search	VFI	4	72.85
CFS	Best First	J48	7	95.98
CFS	Greedy Stepwise	J48	7	95.98
CFS	Scatter Search	J48	6	91.41
Wrapper	Best First	J48	7	95.98
Wrapper	Greedy Stepwise	J48	7	95.98
Wrapper	Scatter Search	J48	6	91.41
CFS	Best First	Random Forest	8	97.01
CFS	Greedy Stepwise	Random Forest	8	95.47
CFS	Scatter Search	Random Forest	8	95.47
Wrapper	Best First	Random Forest	5	96.25
Wrapper	Greedy Stepwise	Random Forest	6	96.25
Wrapper	Scatter Search	Random Forest	5	90.01

Table 3. Classification Comparison using PCA and other feature attribute evaluators in terms of ACC (%)

Classifier	PCA	CFS-Best First	Wrapper-Best First
KNN	91.38	91.04	89.32
SVM	96.24	91.17	90.65
Naive Bayes	85.63	86.76	85.44
MultiboostAB	94.52	91.04	89.39
Rotation Forest	97.06	97.06	93.78
VFI	77.12	74.16	72.22
J48	95.34	95.98	95.98
Random Forest	97.34	97.01	96.25

Fig.4. Shows the ROC curve of the above mentioned trained classifiers.

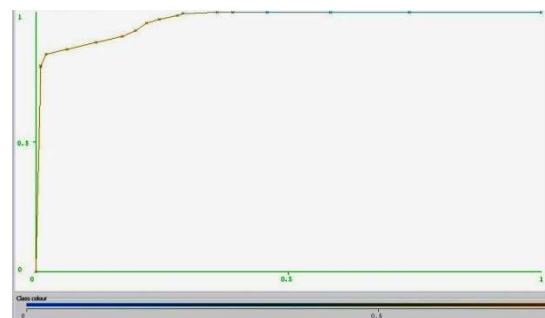


Table 4 describes the classification results using Extreme Machine Learning and Deep Machine Learning. In table 4, we compared the training time,

testing time and classification error using extreme and deep machine Learning. As we can see in the table both learning algorithms are processed to many hidden layers and their evaluations is done in terms of various factors. As depicted in Table 4, it clearly shows that deep machine learning plays a major role in reducing the classification error. As Deep and extreme machine learning are designed to work on large datasets for it is difficult to compare the performance. However, they result in acceptable accuracy levels, and we are currently examining several other publicly available large MRI datasets for enhancing the performance of these two novel approaches (Deep learning and Extreme machine learning approaches).

Table 4. describes the classification results using Extreme Machine Learning and Deep Machine Learning.

Hidden Layers	Training Time(s)			Classification Error		
	10	15	20	10	15	20
Deep Learning	0.56	0.47	0.72	0.083	0.065	0.071
Extreme Learning	0.31	0.31	0.61	0.042	0.042	0.061

The good factor in using deep and extreme learning classification algorithms is that the model is skipped by using dimension reduction evaluators and can be used on unlabeled datasets of MRI brain images where the ROI are classified as unlabeled and can be labeled and classified using these algorithms. But, still on small datasets of our current study is positive and encouraging in terms of low classification error and computation time for training and testing of data.

9. Conclusions

In this study, we have presented a principled approach for investigating brain abnormalities based on wavelet based feature extraction, PCA based feature selection and deep and extreme machine learning based classification comparative to various others classifiers. Experiments on a publicly available brain image dataset show that the proposed principled approach performs significantly better than other competing methods reported in the literature and in the experiments conducted in the study. The classification accuracy of more than 93% in case of deep machine learning and 94% in case of extreme machine learning demonstrates the utility of the proposed method. In this paper, we have applied this method only to axial T2-weighted images at a particular depth inside the brain. The same method can be employed for T1-weighted, proton density and other types of MR images. With the help of above approaches, one can develop software for a diagnostic system for the detection of brain disorders like Alzheimer's, Huntington's, Parkinson's diseases etc. Further, the proposed approach uses reduced data by incorporating feature selection algorithms in the processing loop and still provides an improved recognition and accuracy. The training and testing time for the whole study used by deep and extreme machine learning is much less as compared to SVM and other

traditional classifiers reported in the literature. Further work will be pursued to classify different type of abnormalities, and to extract new features from the MRI brain images on various parameters as age, emotional states and their feedback.

10. References

- Fletcher-Heath, L. M., Hall, L. O., Goldgof, D. B. and Murtagh, F.R. (2001): Automatic segmentation of non-enhancing brain tumors in magnetic resonance images; *Artificial Intelligence in Medicine* 21, pp. 43-63.
- Chaplot, S., Patnaik, L.M. and Jagannathan N.R. (2006): Classification of magnetic resonance brain images using wavelets as input to support vector machine and neural network; *Biomedical Signal Processing and Control* 1, pp. 86-92.
- Gorunescu, F. (2007): Data Mining Techniques in Computer-Aided Diagnosis: Non-Invasive Cancer Detection; *PWASET Volume* 25 November, ISSN 1307-6884, PP. 427-430.
- Kara, S. and Dirgenali, F. (2007): A system to diagnose atherosclerosis via wavelet transforms, principal component analysis and artificial neural networks; *Expert Systems with Applications* 32, pp. 632-640.
- Maitra, M. and Chatterjee A. (2007): Hybrid multi-resolution Slantlet transform and fuzzy c-means clustering approach for normal-pathological brain MR image segregation, *MedEng Phys*, doi:10.1016/j.medengphy.2007.06.009.
- Abdolmaleki, P., Mihara, F., Masuda, K. and DansoBuadu, Lawrence. (1997): Neural networks analysis of astrocytic gliomas from MRI appearances *Cancer Letters* 118, pp. 69-78.
- Rosenbaum, T., Engelbrecht, V., Krolls, W. and Lenard, H. (1999): MRI abnormalities in neuro-bromatosis type 1 (NF1): a study of men and mice; *Brain & Development* 21, pp. 268-273.
- Cocosco, C., Zijdenbos, Alex P. and Evans, Alan C. (2003): A fully automatic and robust brain MRI tissue classification method; *Medical Image Analysis* 7, pp. 513-527.
- Harvard Medical School (1999): The whole Brain Atlas. [ONLINE] Available at: <http://www.med.harvard.edu/aanlib/home.html>.
- Hinton, G.E. and Salakhutdinov, R.R. (2006): Reducing the dimensionality of data with neural networks. *Science*, Vol. 313. No. 5786, pp. 504-507.
- Hinton, G.E. and Osindero, S. (2006): A fast learning algorithm for deep belief nets, *Neural Computation* 18, pp 1527-1554.

- Lin, M.B., Huang, G.B., Saratchandran P. and Sudarajan N. (2005): Fully complex extreme learning machine, *Neurocomputing*, vol (68), pp 306 – 314.
- Huang, G.B., Zhu, Q.Y. and Siew, C.K. (2006): Extreme Learning Machine: Theory and Applications, *Neurocomputing*, vol (70), pp 489-501.
- Huang, G.B., Zhu, Q.Y. and Siew, C.K. (2002): Real-Time Learning Capability of Neural Networks, *IEEE Transactions on Neural Networks*, vol 17(4), pp 863-878.
- Serre, D. (2002): *Matrices: Theory and Applications*, Springer Verlag, New York Inc.
- Wang, Jun. and Zucker, Daniel J. (2000): Solving Multiple-Instance Problem: A Lazy Learning Approach. In: 17th International Conference on Machine Learning, 1119-1125.
- Vapnik, V. (1995): *The Nature of Statistical Learning Theory*, Springer, New York.
- George, H. and John, Pat Langley. (1995): Estimating Continuous Distributions in Bayesian Classifiers. In: Eleventh Conference on Uncertainty in Artificial Intelligence, San Mateo, 338-345.
- Webb, Geoffrey. (2000): *MultiBoosting: A Technique for Combining Boosting and Wagging*. Machine Learning. Vol.40 (No.2).
- Rodriguez, Juan J., Kuncheva, Ludmila I. and Alonso, Carlos J. (2006): Rotation Forest: A new classifier ensemble method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 28(10):1619-1630G.
- Quinlan, Ross. (1993): *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA.
- Breiman, Leo. (2001): Random Forests. *Machine Learning*. 45(1):5-32.
- Hall, M. A. (1998): *Correlation-based Feature Subset Selection for Machine Learning*. Hamilton, New Zealand.
- Kohavi, Ron and John, George H. (1997): Wrappers for feature subset selection. *Artificial Intelligence*. 97(1-2):273-324.
- Hughes, N.P. and Tarassenko, L. (2003): Novel signal shape descriptors through wavelet transforms and dimensionality reduction. In *Wavelet Applications in Signal and Image Processing X*, pages 763–773.
- Mishra, Anurag, Singh, Lavneet and Chetty, Girija (2012): A Novel Image Water Marking Scheme Using Extreme Learning Machine, *Proceedings of IEEE World Congress on Computational Intelligence (WCCI 2012)*, Brisbane, Australia, IEEE Explore.
- Singh, Lavneet and Chetty, Girija (2012): Hybrid Approach in Protein Folding Recognition using Support Vector Machines, *Proceedings of International Conference on Machine Learning and Data Mining (MLDM 2012)*, Berlin, Germany, LNCS, Springer.
- Singh, Lavneet and Chetty, Girija (2012): Review of Classification of Brain Abnormalities in Magnetic Resonance Images Using Pattern Recognition and Machine Learning, *Proceedings of International Conference of Neuro Computing and Evolving Intelligence, NCEI 2012*, Auckland, New-Zealand, LNCS Bioinformatics, Springer.
- Singh, Lavneet and Chetty, Girija (2012): A Novel Approach for protein Structure prediction Using Pattern Recognition and Extreme Machine Learning, *Proceedings of International Conference of Neuro Computing and Evolving Intelligence, NCEI 2012*, Auckland, New-Zealand, LNCS Bioinformatics, Springer.
- Singh, Lavneet and Chetty, Girija (2012): Using Hybrid Neural Networks for Identifying the Brain Abnormalities from MRI Structural Images, *Proceedings of International Conference of Neuro and Image Processing (ICONIP 2012)*, LNCS, Springer.
- Singh, Lavneet and Chetty, Girija (2012): A Novel Approach to Protein Structure Prediction Using PCA or LDA Based Extreme Learning Machines, *Proceedings of International Conference of Neuro and Image Processing (ICONIP 2012)*, LNCS, Springer.
- Singh, Lavneet and Chetty, Girija (2012): Investigating Brain Abnormalities from MRI Images Using Pattern Recognition and Machine Learning Techniques, *Proceedings of International Conference on Pattern Recognition in Bioinformatics (PRIB 2012)*, LNCS, Springer.
- Singh, Lavneet and Chetty, Girija (2012): A Novel Approach to Protein Structure Prediction Using PCA Based Extreme Learning Machines and Multiple Kernels, *Proceedings of The 12th International Conference on Algorithm and Architectures for Parallel Processing (ICA3PP 2012)*, LNCS, Springer.

Indirect Weighted Association Rules Mining for Academic Network Collaboration Recommendations

Yun Sing Koh¹

Gillian Dobbie²

Department of Computer Science
University of Auckland, New Zealand,
Email: ykoh@aut.ac.nz¹, rpears@aut.ac.nz²

Abstract

Collaborative research is increasingly important and popular in academic circles. However for young researchers identifying new research collaborators to form joint research and analyzing the level of cooperation of the current partners can be a very complex task. Thus recommendation of new collaborations would be important for young researchers. This paper presents a new approach to recommend collaborators in an academic social network using the co-authorship network. We propose a weighted indirect rule mining approach using a novel weighting mechanism called sociability.

Keywords: Social Network, Indirect Weighted Association Rule Mining, Sociability

1 Introduction

Rapid growth and exponential use of social digital media has led to an increase in popularity of social networks and the emergence of social network mining which combines data mining with social computing. As social networks are generally made of social entities that are linked by some specific type of interdependency such as friendship. Social networks represents social relationships in terms of nodes and links. Nodes are the individual actors within the networks, and links are the relationships between the actors. Social Network Analysis (SNA) analyses the importance relationships between actors, and is a central point to the evaluation and the analysis of social interactions.

Nowadays, this type of network is commonly used, and each network connects millions of users. Social network mining aims to discover implicit, previously unknown and potentially useful knowledge from a vast pool of data residing in the social networking sites such as Twitter (Weng et al. 2010, Ghosh et al. 2012), Facebook (Fan & Yeung 2010), Google+ (Leenes 2011), LinkedIn.

An example of the social network application is the Co-authorship Social Network which represents a scientific collaboration network (Huang et al. 2008). Increasing research collaboration amongst researchers can bring together different points of view to address a particular research issue. Furthermore, studies have shown that scholars with higher levels of collaboration tend to be more productive (Lotka 1926). Thus, it would be beneficial for new emerging researchers

to find potential successful collaborators. Yet traditional digital libraries and search engines focus on discovering relevant documents which does not make it straightforward to search for people who share similar research interests (Chen et al. 2011). There have been a few digital library platforms, such as ArnetMiner (Tang et al. 2008) and Microsoft Academic Search (Microsoft Academic Search 2011) which returns a list of experts given a particular domain. However, the list only provides a limited set of names does not consider the implicit social networks of the experts.

To help in efficiently discovering potential collaborators, we present a new approach that considers social network structure based on reachability, and sociability of a researcher as a recommendation tool for potential collaborators. Our approach weights researchers based on a sociability factor, which tries to capture how often they work with a different researcher. Using these weights we are able to generate rules to describe the connection between a researcher and a collaborator. We then use these rules to generate recommendations to other researchers who are indirectly associated to them and may be possible collaborators. In our experiments we used the collaborative network from the digital community DBLP.

The paper is organized as follows. In Section 2 we look at related work in the area of recommendations for social network. In Section 3 we present our weighted indirect association rule mining approach. In Section 4 we discuss our experimental results. Finally, Section 5 concludes the paper.

2 Related Work

Ever since the proliferation of social network research, there has been a considerable amount of research carried out to build recommendations for social networks (Ogata et al. 2001, Quercia & Capra 2009, Karagannis & Vojnovic 2009, Chen et al. 2009, Weng & Chang 2008, Cheong & Corbitt 2009, Roth et al. 2010). The related work presented in this section aims to use social networks in the context of recommendation systems for an academic network.

Aleman-Meza et al. (2006) proposed a solution to solve the conflict of interest problem using social networks. The main objective was to detect relationships of conflict of interest amongst authors of scientific papers and potential reviewers of those papers based on public sources such as DBLP and the Friend of a Friend project.

Kautz et al. (1997) proposed the ReferralWeb system to identify experts in searches by keywords and generate a path of social relationships between a user and the recommended expert. The proposed solution models and extracts existing relationships among people in the area of Computer Science using public data available on Web documents. McDonald (2003) pro-

Copyright ©2012, Australian Computer Society, Inc. This paper appeared at the 10th Australasian Data Mining Conference (AusDM 2012), Sydney, Australia, December 2012. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 134, Yanchang Zhao, Jiuyong Li, Paul Kennedy, and Peter Christen, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

posed an evaluation of two different social networks that can be used in a system to recommend individuals for possible collaborations. The system matches individuals looking for expertise within people that could have this expertise.

Zaiane et al. (2007) proposed a technique which explored a social network based on the DBLP database by using a new random walk approach to find interesting information about the research community and then recommended collaborations. The approach aims at helping the user in the process of searching for relevant conferences, similar authors and interesting research topics.

Chen et al. (2011) proposed CollabSeer which is an open system to recommend potential research collaborators for scholars and scientists. The proposed approach discovers collaborators based on the structure of the coauthor network and a users research interests. Currently, three different network structure analysis methods that use vertex similarity are supported in CollabSeer: Jaccard similarity, cosine similarity, and the relation strength similarity measure.

There has been some research in using frequent pattern mining in finding interesting patterns in an academic network (Adnan et al. 2009, Nohuddin et al. 2012). In this paper we propose a new approach, in finding recommendations for an academic network, using an indirect frequent mining approach.

3 Mining Weighted Indirect Association Rules

In this section we describe our proposed weighted indirect rule mining approach. In Section 3.1 we discuss the weighted association rule mining approach, and our weighting mechanism called sociability weight. In Section 3.2 and Section 3.3 we discuss the combination of weighted association rule mining and indirect association rule mining.

3.1 Weighted Association Rules Based On Sociability

Association rule mining is an important data mining task that discovers relationships among items in a transaction database. Most approaches to association rule mining assume that all items within a dataset have a uniform distribution with respect to support. Therefore, weighted association rule mining was introduced to provide a notion of importance to individual items.

Given a set of items, $I = \{i_1, i_2, \dots, i_n\}$, a transaction may be defined as a subset of I and a dataset as a set D of transactions. A set X of items is called an itemset. The support of X , $\text{sup}(X)$, is the proportion of transactions containing X in the dataset. An *association rule* is an implication of the form $X \rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \emptyset$. The rule $X \rightarrow Y$ has *support* of s in the transaction set D , if $s = \text{sup}(XY)$. The rule $X \rightarrow Y$ holds in the transaction set D with *confidence* c where $c = \text{conf}(X \rightarrow Y) = \text{sup}(XY)/\text{sup}(X)$. The association rules are also known as a direct association rules. Given a transaction database D , a support threshold minsup and a confidence threshold minconf , the task of association rule mining is to generate all association rules that have support and confidence above the user-specified thresholds.

In weighted association rule mining a weight w_i is assigned to each item i , reflecting the relative importance of an item over other items that it is associated with. The weighted support of an item i is $w_i \text{sup}(i)$. Similar to traditional association rule mining, a weighted support threshold and a confidence

threshold is assigned to measure the strength of the association rules produced. The weight of a k -itemset, X , is given by:

$$\left(\sum_{i \in X} w_i\right) \text{sup}(X) \quad (1)$$

Here a k -itemset, X , is considered a frequent itemset if the weighted support of this itemset is greater than the user-defined minimum weighted support ($w\text{minsup}$) threshold.

$$\left(\sum_{i \in X} w_i\right) \text{sup}(X) \geq w\text{minsup} \quad (2)$$

The weighted support of a rule $X \rightarrow Y$ is:

$$\left(\sum_{i \in X \cup Y} w_i\right) \text{sup}(XY) \quad (3)$$

In our approach we proposed a new sociability weight as the weighting mechanism.

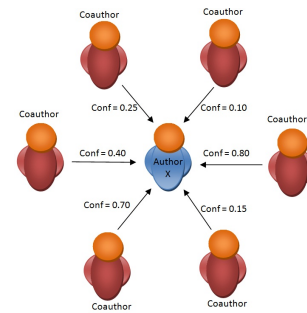


Figure 1: Author-Coauthor Graph

Definition 1 (Sociability Weight). *The sociability weight is defined based on the coauthors, i , an expert (author), k , has and the confidence of the coauthors towards the author. Given an author, k , which is connected to a set of n coauthors, the sociability weight, soc_k is defined as:*

$$\text{soc}_k = \sum_i^n \frac{\text{sup}(i, k)}{\text{sup}(i)} \quad (4)$$

which is equivalent to

$$\text{soc}_k = \sum_i^n \text{conf}(i \rightarrow k)$$

The reasoning behind this is that we are interested in promoting an expert (author) which works with a range of other researchers (coauthors). In turn the other researchers must also have a high confidence towards the author, which means that they have published frequently with the same expert.

Figure 1 shows an author-coauthors relationship. The arrows represent the rules formed between a coauthor and author X . In this example the sociability weight for the author X is $0.25 + 0.40 + 0.70 + 0.15 + 0.80 + 0.10 = 2.65$. The weights are used to float experts which are deemed to be important to the top.

Here we discuss weighted direct rules in collaboration recommendation.

Definition 2 (Weighted Direct Rule). *Let D be a dataset. A weighted direct association rule between two authors is the relationship between an author X and its coauthor Y , where $X \rightarrow Y$, where $X \subseteq D$, $Y \subseteq D$, and $X \cap Y = \emptyset$. The $w\text{sup}(X \rightarrow Y) \geq w\text{minsup}$ and $\text{conf}(X \rightarrow Y) \geq \text{minconf}$.*

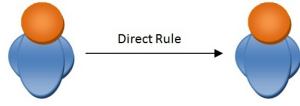


Figure 2: Direct Rule

Weighted direct association rules represent regularities discovered from a large dataset based on the weighting scheme. The problem of mining association rules is to extract rules that are strong enough and have the weighted support ($wsup$) and confidence value greater than given thresholds: minimum weighted direct support ($wminsup$) and minimum direct confidence ($minconf$).

We use the rules generated from weighted association rules in this section to form indirect rules which we will discuss in the next section.

3.2 Indirect Association Rules in Social Networking

In a classical sense, an indirect association pattern refers to a pair of items that rarely occur together but highly depend on the presence of a mediator itemset (Tan & Kumar 2002). Indirect association has been used extensively to build web recommendation systems (Kazienko 2009, Tan & Kumar 2002). In this research, we propose to use a weighted indirect association rule mining approach for collaboration recommendation in an academic social network.

Let us consider another type of associations: indirect association rules.

Definition 3 (Weighted Indirect Itemset). *An itemset (pair of researchers) $\{X, Y\}$ is indirectly associated via a mediator M , if $sup(X, Y) < wminsup$, $sup(X, M) \geq wminsup$, and $sup(Y, M) \geq wminsup$.*

Definition 4 (Weighted Indirect Rule). *Let D be a dataset. A weighted indirect association rule $X \rightarrow^{M\#} Y$ is the indirect relationship from X to Y with respect to M , for which two direct weighted association rules exist: $X \rightarrow M$ and $M \rightarrow Y$, where $X \subseteq D$, $M \subseteq D$, and $Y \subseteq D$; $X \neq M \neq Y$; and $conf^I(X \rightarrow^{M\#} Y) \geq minconf^I$.*

Each weighted indirect association rule $X \rightarrow^{M\#} Y$ has an indirect confidence $conf^I$ value which can be defined as follows:

$$conf^I(X \rightarrow^{M\#} Y) = conf(X \rightarrow M) \cdot conf(M \rightarrow Y)$$

For example given there are two rules $X \rightarrow M$ with $conf = 0.90$ and $M \rightarrow Y$ with $conf = 0.80$. Thus, $conf^I(X \rightarrow^{M\#} Y) = 0.90 \times 0.80 = 0.72$. There are two types of weighted indirect rule: partial indirect and complete indirect.

Definition 5 (Weighted Partial Indirect Rule). *Let D be a dataset. A weighted partial indirect association rule $X \rightarrow^{M\#} Y$ is the indirect relationship from X to Y with respect to M , for which two direct weighted association rules exist: $X \rightarrow M$ and $M \rightarrow Y$, where $X \subseteq D$, $M \subseteq D$, and $Y \subseteq D$; $X \neq M \neq Y$; $conf^I(X \rightarrow^{M\#} Y) \geq minconf^I$; and $X \cap Y \neq \emptyset$.*

A weighted partial indirect rule $X \rightarrow^{M\#} Y$ reflects one indirect association existing between X and Y , with no direct association $X \rightarrow Y$, even though X occurs together with Y (shown in Figure 3). In Figure 3 the solid line between X and Y represents that both the authors are co-authors but $sup(X, Y) < wminsup$, thus no weighted direct rule between these two authors are generated.

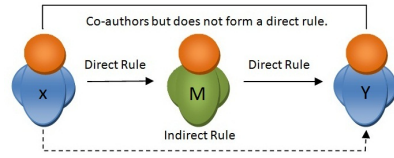


Figure 3: Partial Indirect Rule

Definition 6 (Weighted Complete Indirect Rule). *Let D be a dataset. A weighted complete indirect association rule $X \rightarrow^{M\#} Y$ is the indirect relationship from X to Y with respect to M , for which two direct weighted association rules exist: $X \rightarrow M$ and $M \rightarrow Y$, where $X \subseteq D$, $M \subseteq D$, and $Y \subseteq D$; $X \neq M \neq Y$; $conf^I(X \rightarrow^{M\#} Y) \geq minconf^I$; and $X \cap Y = \emptyset$.*

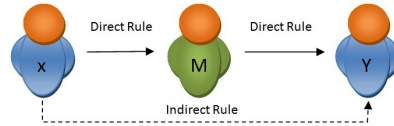


Figure 4: Complete Indirect Rule

A weighted complete indirect rule $X \rightarrow^{M\#} Y$ reflects one indirect association existing between X and Y , with no direct association $X \rightarrow Y$, and X does not occur with Y (shown in Figure 4).

3.3 Weighted Indirect Association Rule Mining

In our algorithm, we focus on finding weighted indirect rules by combining the Sociability weight in Section 3.1 and the indirect rule mining approach described in Section 3.2. In this section we describe this combined approach. The algorithm is divided into two major phases. In Phase 1 we generate all weighted frequent itemsets (Algorithm 1) using a sociability weight function shown in Algorithm 2. In Phase 2, we find all indirect associations (Algorithm 3).

A general weighted association rule mining algorithm is shown in Algorithm 1. The algorithm requires a weighted minimum support to be provided. In this algorithm L_k represents the weighted frequent itemsets and C_k represents the candidate itemsets. Candidate itemsets whose weighted support exceeds the weighted minimum support are considered large itemsets and will be included in the rule generation phase.

Algorithm 1 Weighted candidate generation algorithm

Input: Transaction database D , $wminsup$ value, universe of items I
 Output: Weighted frequent itemsets, L_k
 $k \leftarrow 1$
 $L_k \leftarrow \{\{i\} | i \in I, soc(i) * sup(i) \geq wminsup\}$
 while $L_k \neq \emptyset$ do
 $k \leftarrow k + 1$
 $C_k \leftarrow \{x \cup y | x, y \in L_{k-1}, |x \cap y| = k - 2\}$
 $L_k \leftarrow \{c | c \in C_k, soc(c) * sup(c) \geq wminsup\}$
 end while
 return $\bigcup_{t=2}^{k-1} L_t$

Algorithm 2 Sociability Weight, $\text{soc}(i)$

Input: Item i , universe of items I
Output: Sociability Weight
{Find the neighbourhood of item i .}
 $N_i \leftarrow \{\{j\} | j \in I, \text{sup}(i, j) > 0\}$
return $\sum_{j \in N_i} \frac{\text{sup}(j, i)}{\text{sup}(j)}$

Frequent itemset L_k is used to generate candidate indirect associations P . Each candidate in P is a triplet $\langle x, y, M \rangle$, where x and y are the items which are indirectly associated by mediator M . P is generated joining the frequent itemsets in L_k . During the join, a pair of frequent itemsets $\{x_1, x_2, \dots, x_k\}$ and $\{y_1, y_2, \dots, y_k\}$ are joinable if the two itemsets have exactly $k - 1$ items in common. If so, they generate a candidate indirect association $\langle x, y, M \rangle$, where x and y are the different items, one from each k -itemset, and M is the set of common items.

Algorithm 3 Indirect rule mining algorithm

Input: Itemset L_k , minconf^I value
Output: Indirect Rules
 $R \leftarrow \emptyset$
 $P \leftarrow \{x \cup y | x, y \in L_k, |x \cap y| = k - 1\}$
for $\langle x, y, M \rangle \in P$ do
 if $\text{conf}(x \rightarrow M) \cdot \text{conf}(M \rightarrow y) > \text{minconf}^I$
 then
 $R \leftarrow \{\langle x, y, M \rangle | x \in P, y \in P\}$
 end if
end for
return R

For example, two itemsets $\{a, y\}$ and $\{a, z\}$ can be joined together to generate a candidate indirect association $\langle y, z, \{a\} \rangle$. Since the candidate indirect associations are generated by joining two frequent itemsets, they certainly satisfy the mediator confidence condition, minconf^I . In this example, $\{a\}$ is the mediator.

4 Results and Evaluation

In this section we evaluate the performance of mining indirect weighted association rules for collaboration recommendations. To the best of our knowledge, our technique is the first to suggest recommendation using indirect rule mining in a social media context. However there has been some work carried out based on other techniques.

We compare our results to strength vertex similarity method used in Collabseer (Chen et al. 2011). The relation strength of two adjacent authors is proportional to the number of their coauthored articles. If author A has n_A publications, author B has n_B publications, author A and author B coauthored n_{AB} articles. The relation strength from author A to author B is defined as follows.

$$R(A, B) = \frac{n_{AB}}{n_A}$$

For two non-collaborator authors A and C , if A could reach C only through author B , then how close author A is to author C should be proportional to the relation strength of author A to author B and the relation strength of author B to author C . We define indirect relation strength from author A to author C as:

$$R'(A, C) = R(A, B) \cdot R(B, C)$$

which can be written as:

$$R'(A, C) = \frac{n_{AB}}{n_A} \cdot \frac{n_{BC}}{n_B} = \frac{\text{sup}(AB)}{\text{sup}(A)} \cdot \frac{\text{sup}(BC)}{\text{sup}(B)}.$$

Thus,

$$R'(A, C) = \text{conf}(A \rightarrow B) \cdot \text{conf}(B \rightarrow C)$$

which is similar to a standard indirect rule conf^I measure for the indirect rule $A \rightarrow^{B\#} C$ with B as the mediator where is not included.

In their approach all authors are given equal weighting. We believe some authors are more active and more likely to form collaboration. Thus we used the sociability weights to promote these collaborations.

Table 1: Characteristics of Datasets

Dataset	Trans	Items	Avg Len
DBLP Data Mining	34215	2117	2.71
DBLP Artificial Intelligence	35380	6817	2.57
DBLP Software Engineering	21628	1591	2.58
DBLP Database	11931	2922	2.59
T10I4D100K	100000	870	10.1

In our experiments we used the DBLP Computer Science Bibliography dataset (<http://dblp.uni-trier.de/xml/>), and a frequent mining dataset that is available from the Frequent Itemset Mining Implementations (FIMI) repository (<http://fimi.ua.ac.be/>). When we use the frequent mining dataset, we map each unique item as an author and the set of items it co-occurs with as their collaborators. Note that the transactions in the above dataset share similar characteristics as those in a academic collaboration network. T10I4D100K is a dataset with a large number of items and transactions. The lengths of transactions within these datasets are relatively short. These datasets represent scenarios of a social network which comprises of many people with a small group of people that they interact with. This is similar to that of a collaboration network. From the DBLP dataset, we extracted papers written from 2000-2009. From the selected papers we extracted, we partitioned the datasets into different research areas based on the publication venue. We chose the datasets from four different research areas: databases, data mining, artificial intelligence, and software engineering. Table 1 summarizes the characteristics of the datasets used in the following experiments. For each dataset, we show the number of transactions, number of items, and average length of the transactions.

4.1 Number of Indirect Rules

In the first experiment we compare the number of recommendations generated by our algorithm and the existing algorithm. The number of indirect rules represents the number of recommendations found. The results are shown in Figure 5 and Figure 6.

Figure 5 shows the complete indirect rules generated, whereas, Figure 6 shows the number of partial indirect rules generated. In all the experiments we used a w_{minsup} of 0.001. We varied the minConf^I from 0.20 to 0.50. The number of recommendations or rules generated are inversely proportional to the minConf^I threshold. When the minConf^I threshold decreases, the number of recommendations produced increases.

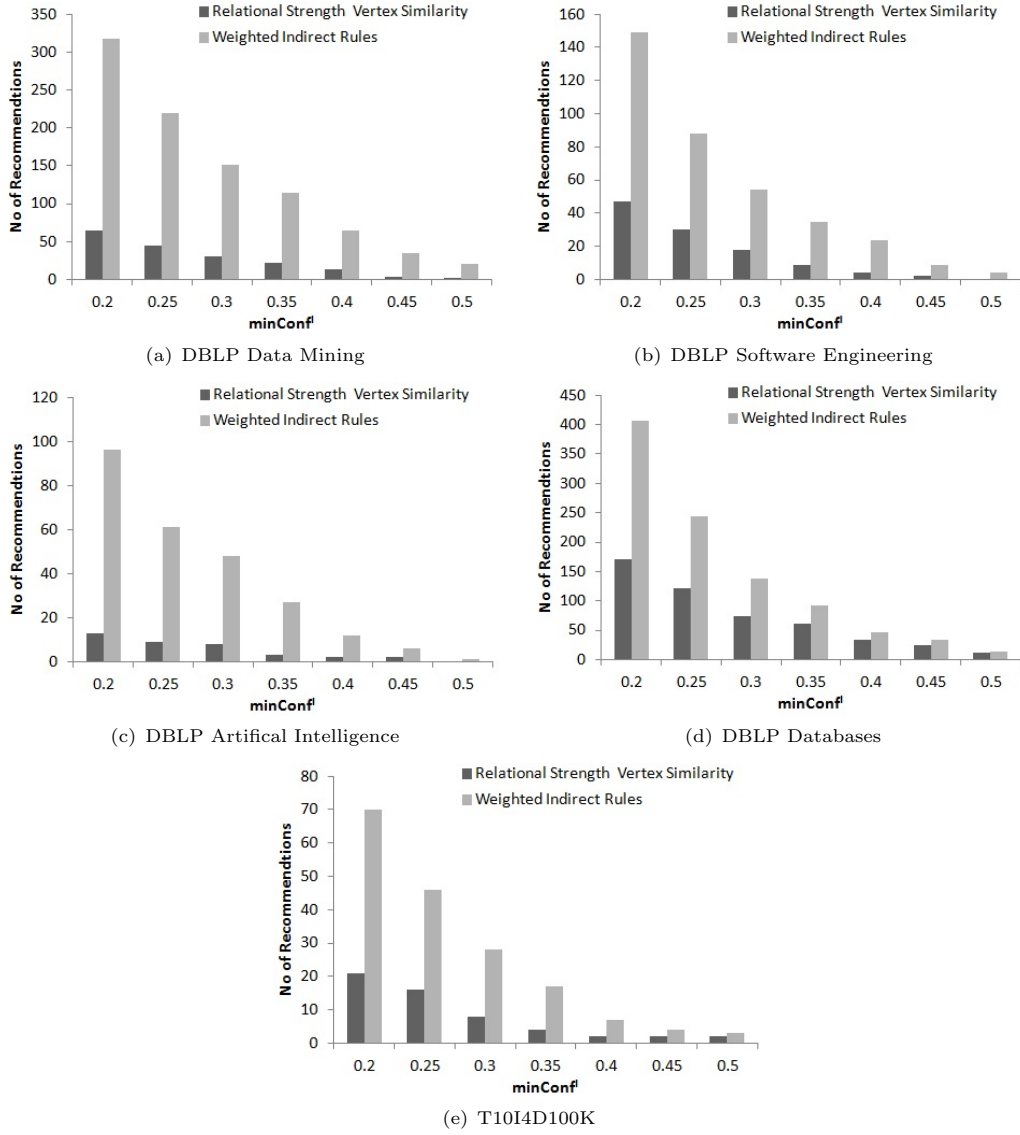


Figure 5: Number of Complete Indirect Rules

Overall the number of recommendations found by our technique is higher than the relation strength vertex similarity algorithm. The number of recommendations found by our technique is between 1 to 7 times more than the relation strength vertex similarity algorithm.

4.2 Lift Analysis

To evaluate the strength of the recommendations (rules) produced we use the lift measure. Lift is a well-known statistical measure that can be used to rank rules in IBMs Intelligent Miner (Bayardo & Agrawal 1999):

$$\text{lift}(X \rightarrow Y) = \frac{\sup(XY)}{\sup(X) \cdot \sup(Y)}$$

Note that if the occurrence of A and B are perfectly independent, the $\text{lift}(X \rightarrow Y) = 1$. If X and Y appear together more often than we would expect under independence, the lift is greater than 1, and otherwise it is less than one.

In the similar way indirect confidence, conf^I , is defined for indirect rule, we adapt the lift measure to an indirect lift measure, lift^I , for $X \rightarrow^{M\#} Y$ as:

$$\text{lift}^I(X \rightarrow^{M\#} Y) = \text{lift}(X \rightarrow M) \cdot \text{lift}(M \rightarrow Y)$$

Table 2: Average Lift Values

Dataset	Weighted Indirect Rules		
	Complete	Partial	Total
DBLP Data Mining	505033.3	2923.5	507956.7
DBLP Artificial Intelligence	142485.7	114	142599.6
DBLP Software Engineering	303300.3	180.7	303481.1
DBLP Databases	502548	107.9	502655.9
T10I4D100K	4578.4	4.5	4582.9
Dataset	Rel. Strength Vertex Similarity		
	Complete	Partial	Total
DBLP Data Mining	116442.2	5640.2	122082.4
DBLP Artificial Intelligence	121360.7	361.3	121722.0
DBLP Software Engineering	139179.6	373	139552.6
DBLP Databases	525566.7	242.3	525809.0
T10I4D100K	3495.9	178.9	3674.7

Table 2 shows the average lift values produced by our algorithm compared to the strength vertex similarity method. Overall our algorithm consistently produced rules which had a higher lift value. In this experiment, the minConf^I set at 0.20. We chose a low minConf^I value as it produces the most recommendations for both algorithms. If a higher minConf^I threshold is selected, the set of recommendations would be a subset of the recommendations generated whilst using the lower minConf^I thresh-

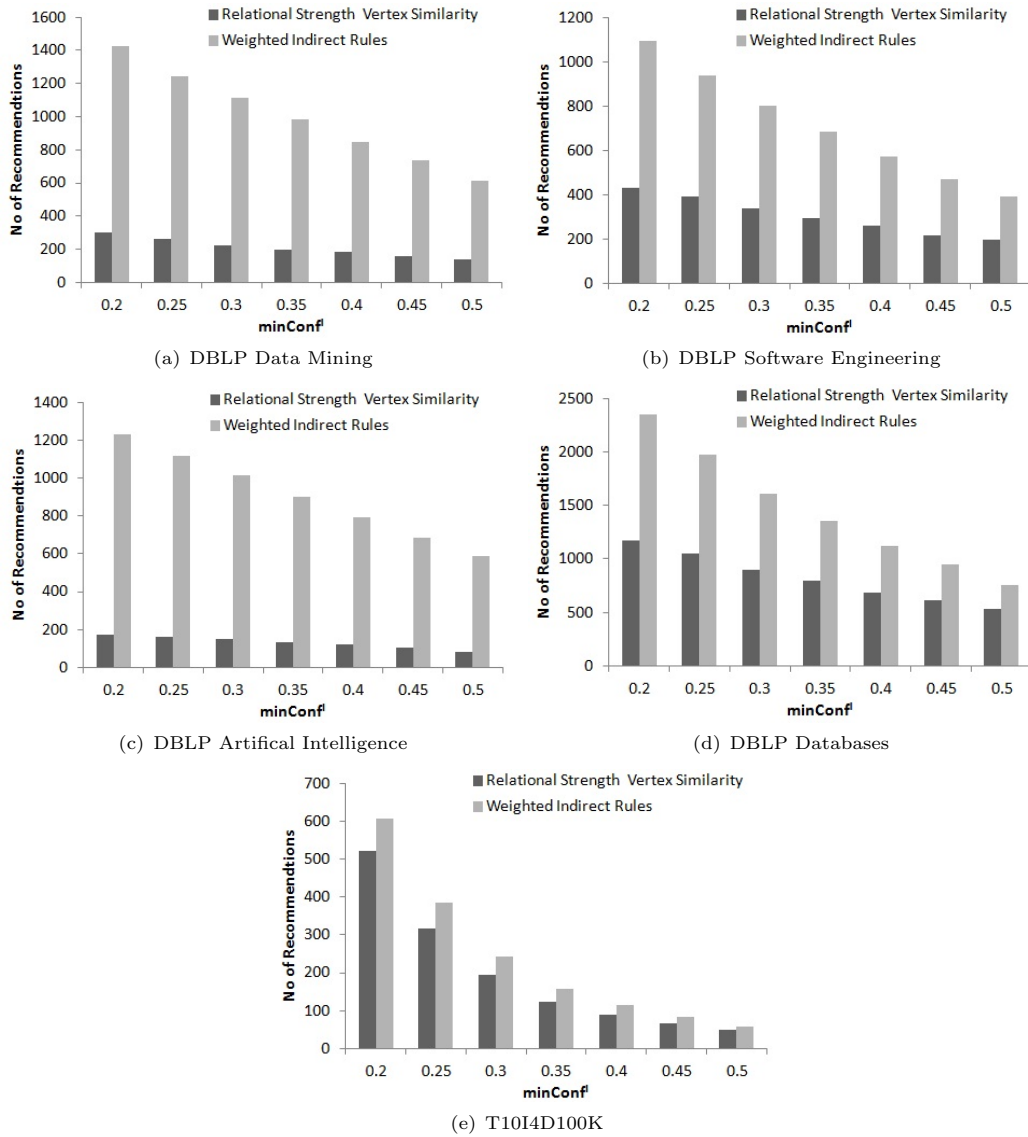


Figure 6: Number of Partial Indirect Rules

old. Thus by choosing a lower $minConf^I$ we are evaluating the superset of the recommendations generated.

4.3 Runtime Analysis

Here we compare the execution time of the two algorithm. Figure 7 shows the results of the experiment. Overall the number of recommendations influence the

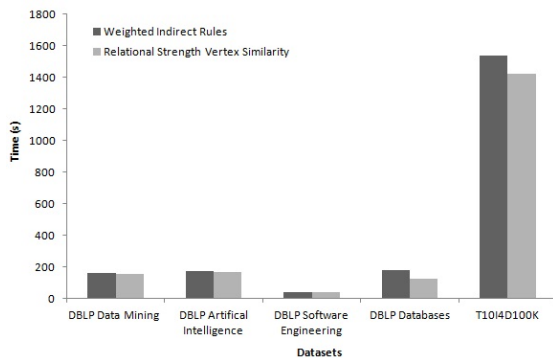


Figure 7: Runtime Analysis

runtime of the algorithms. Despite the additional

recommendations generated by the weighted indirect rule algorithm, the runtime is still comparable to the relation strength vertex similarity algorithm.

5 Conclusions

In this paper we proposed a novel algorithm to mine social networks for collaboration recommendation. Our proposed technique uses a weighted mechanism called sociability weight and combined it with indirect association rule mining. Overall our technique generated more recommendations as compared to a previous approach, relation strength vertex similarity algorithm and the additional recommendation are considered strong.

In the future we may consider other features such as citations or latent semantic analysis (abstracts or keywords for example), which better spans across academic domains.

References

Adnan, M., Alhajj, R. & Rokne, J. (2009), Identifying social communities by frequent pattern mining, *in* 'Information Visualisation, 2009 13th International Conference', pp. 413–418.

- Aleman-Meza, B., Nagarajan, M., Ramakrishnan, C., Ding, L., Kolari, P., Sheth, A. P., Arpinar, I. B., Joshi, A. & Finin, T. (2006), Semantic analytics on social networks: experiences in addressing the problem of conflict of interest detection, in 'Proceedings of the 15th international conference on World Wide Web', WWW '06, ACM, New York, NY, USA, pp. 407–416.
- Bayardo, Jr., R. J. & Agrawal, R. (1999), Mining the most interesting rules, in 'Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining', KDD '99, ACM, New York, NY, USA, pp. 145–154.
URL: <http://doi.acm.org/10.1145/312129.312219>
- Chen, H.-H., Gou, L., Zhang, X. & Giles, C. L. (2011), Collabseer: a search engine for collaboration discovery, in 'Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries', JCDL '11, ACM, New York, NY, USA, pp. 231–240.
- Chen, J., Geyer, W., Dugan, C., Muller, M. & Guy, I. (2009), Make new friends, but keep the old: recommending people on social networking sites, in 'Proceedings of the 27th international conference on Human factors in computing systems', CHI '09, ACM, New York, NY, USA, pp. 201–210.
- Cheong, F. & Corbitt, B. J. (2009), A social network analysis of the co-authorship network of the australasian conference of information systems from 1990 to 2006, in '17th European Conference on Information Systems (ECIS 2009)', pp. 292–303.
- Fan, W. & Yeung, K. (2010), Virus propagation modeling in facebook, in 'Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on', pp. 331–335.
- Ghosh, S., Viswanath, B., Kooti, F., Sharma, N. K., Korlam, G., Benevenuto, F., Ganguly, N. & Gumadi, K. P. (2012), Understanding and combating link farming in the twitter social network, in 'Proceedings of the 21st international conference on World Wide Web', WWW '12, ACM, New York, NY, USA, pp. 61–70.
URL: <http://doi.acm.org/10.1145/2187836.2187846>
- Huang, J., Zhuang, Z., Li, J. & Giles, C. L. (2008), Collaboration over time: characterizing and modeling network evolution, in 'Proceedings of the international conference on Web search and web data mining', WSDM '08, ACM, New York, NY, USA, pp. 107–116.
- Karagiannis, T. & Vojnovic, M. (2009), Behavioral profiles for advanced email features, in 'Proceedings of the 18th international conference on World wide web', WWW '09, ACM, New York, NY, USA, pp. 711–720.
- Kautz, H., Selman, B. & Shah, M. (1997), 'Referral web: combining social networks and collaborative filtering', *Commun. ACM* **40**, 63–65.
- Kazienko, P. (2009), 'Mining indirect association rules for web recommendation', *Int. J. Appl. Math. Comput. Sci.* **19**, 165–186.
- Leenes, R. (2011), Who needs facebook or google+ anyway: privacy and sociality in social network sites, in 'Proceedings of the 7th ACM workshop on Digital identity management', DIM '11, ACM, New York, NY, USA, pp. 31–32.
URL: <http://doi.acm.org/10.1145/2046642.2046650>
- Lotka, A. J. (1926), 'The frequency distribution of scientific productivity', *Journal of the Washington Academy of Sciences* **16**(12), 317–323.
- McDonald, D. W. (2003), Recommending collaboration with social networks: a comparative evaluation, in 'Proceedings of the SIGCHI conference on Human factors in computing systems', CHI '03, ACM, New York, NY, USA, pp. 593–600.
- Microsoft Academic Search (2011), <http://academic.research.microsoft.com/>.
- Nohuddin, P. N., Coenen, F., Christley, R., Setzkorn, C., Patel, Y. & Williams, S. (2012), 'Finding interesting trends in social networks using frequent pattern mining and self organizing maps', *Knowledge-Based Systems* **29**(0), 104–113.
- Ogata, H., Yano, Y., Furugori, N. & Jin, Q. (2001), 'Computer supported social networking for augmenting cooperation', *Comput. Supported Coop. Work* **10**, 189–209.
- Quercia, D. & Capra, L. (2009), Friendsensing: recommending friends using mobile phones, in 'Proceedings of the third ACM conference on Recommender systems', RecSys '09, ACM, New York, NY, USA, pp. 273–276.
- Roth, M., Ben-David, A., Deutscher, D., Flysher, G., Horn, I., Leichtberg, A., Leiser, N., Matias, Y. & Merom, R. (2010), Suggesting friends using the implicit social graph, in 'Proceedings of the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining'.
- Tan, P.-N. & Kumar, V. (2002), Mining indirect associations in web data, in R. Kohavi, B. M. Masand, M. Spiliopoulou & J. Srivastava, eds, 'WEBKDD', Vol. 2356 of *Lecture Notes in Computer Science*, Springer, pp. 145–166.
- Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L. & Su, Z. (2008), Arnetminer: extraction and mining of academic social networks, in 'Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining', KDD '08, ACM, New York, NY, USA, pp. 990–998.
URL: <http://doi.acm.org/10.1145/1401890.1402008>
- Weng, J., Lim, E.-P., Jiang, J. & He, Q. (2010), Twiterrank: finding topic-sensitive influential twitterers, in 'Proceedings of the third ACM international conference on Web search and data mining', WSDM '10, ACM, New York, NY, USA, pp. 261–270.
URL: <http://doi.acm.org/10.1145/1718487.1718520>
- Weng, S.-S. & Chang, H.-L. (2008), 'Using ontology network analysis for research document recommendation', *Expert Syst. Appl.* **34**, 1857–1869.
- Zaiane, O. R., Chen, J. & Goebel, R. (2007), Dbconnect: mining research community on dblp data, in 'Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis', WebKDD/SNA-KDD '07, ACM, New York, NY, USA, pp. 74–81.

Using network evolution theory and singular value decomposition method to improve accuracy of link prediction in social networks

Qinxue Meng

Paul J. Kennedy

Centre for Quantum Computation and Intelligent Systems
 Faculty of Engineering and Information Technology
 University of Technology, Sydney
 PO Box 123 Broadway NSW 2007 Australia
 Email: Qinxue.Meng@student.uts.edu.au
 Paul.Kennedy@uts.edu.au

Abstract

Link prediction in large networks, especially social networks, has received significant recent attention. Although there are many papers contributing methods for link prediction, the accuracy of most predictors is generally low as they treat all nodes equally. We propose an effective approach to identifying the level of activities of nodes in networks by observing their behaviour during network evolution. It is clear that nodes that have been active previously contribute more to the changes in a network than stable nodes, which have low activity. We apply truncated singular value decomposition (SVD) to exclude the interference of stable nodes by treating them as noise in our dataset. Finally, in order to test the effectiveness of our proposed method, we use co-authorship networks from an Australian university from between 2006 and 2011 as an experimental dataset. The results show that our proposed method achieves higher accuracy in link prediction than previous methods, especially in predicting new links.

Keywords: Link prediction, network evolution, Katz measure, singular value decomposition, social network analysis.

1 Introduction

Social networks are a type of graph structure whose nodes represent people or other entities embedded in a social context and relationships are indicated by links between nodes. There are many natural examples of social networks (Scott & Carrington 2011). This paper considers academic collaboration networks. In this type of network, researchers are linked via media, for example co-authored publications, projects or grants. Meanwhile, abstract concepts can also work as media to establish connections and relationships in social networks. Examples of useful abstract concepts to illustrate this point are common interests and friendship.

A common feature of social networks is that they are generally dynamic rather than static compared with other networks, such as gene networks. They grow and change quickly over time by adding, strengthening or removing links and nodes. Under these circumstances and since examples of social networks abound in the real world, identifying patterns

of evolution and predicting changes in links becomes an urgent and fundamental question that is still not well understood.

Traditionally, the analysis of social networks (Carrington et al. 2005) focuses on a single snapshot of a network and generally makes predictions based on local neighbourhood algorithms: the more common neighbours two nodes have, the higher the possibility they have to develop a link in the future. An obvious weakness of this is in treating every node equally. As a result, accuracy of prediction is not very satisfactory (Liben-Nowell & Kleinberg 2007).

The reason for this is straightforward: link prediction in social networks is a non-stationary problem. The underlying relationships between people are ever changing. Changes in relationships in social networks are the result of many contributions as the nodes are people. Consequently, during the evolution of networks, not all nodes tend to generate new links over time. For example, nodes labeled as “stable” may maintain their old relationships for a long time without any changes while “active” nodes tend to establish new links with others even if there is a long path between the nodes in the existing network. Finally, “regular” nodes have a tendency to expand their social circle, but more slowly.

Based on these considerations, we propose a new method to improve the accuracy of link prediction. We group and label nodes in a social network based on their level of behaviour during network evolution. Then singular value decomposition (SVD) is applied to assess the contribution of different groups to the results of link prediction.

In our experiment, we choose co-authorship networks as our experimental dataset. We form prediction problems in two categories: (i) predicting *all* links and (ii) predicting *new* links. Co-authorship networks are applicable to both of these kinds of problem. Unlike friendship networks, co-authorship networks can be seen as a kind of temporary network, where coauthored publications are transitory. Whether two authors are connected depends on predicting their co-publications in the future. Meanwhile, co-authorship networks represent academic collaboration and therefore it is meaningful to predict new links.

The experimental dataset is gathered from the research management database of an Australian university, namely the University of Technology, Sydney, over the five year period from 2006 to 2010. The experimental results show that the accuracies of predictors using a succession of years of data for network evolution exceed those on a single snapshot and that smoothing of the network using truncated singular value decomposition improves prediction.

The rest of this paper is organized as follows. Section 2 gives a brief literature review of some previous research on link prediction. The experimental dataset and methodology are given in sections 3 and 4, followed by the experimental results in section 5 and conclusion and future work in section 6.

2 Related work

2.1 Models of network evolution

There have been a number of methods proposed and analyzed in the area of network evolution. The majority focus on changes in topological structure of networks. As graphs can be abstracted as a probability distribution, one typical approach (Guo et al. 2007) models dynamic graphs with a sequential linear model, namely Markov chains, to form a series of probability distributions (Snijders 2002). Another similar method (Song et al. 2009) introduces a Bayesian network to model the process of network evolution. Probabilistic methods of building network models such as those outlined above are generally based on a single snapshot of a network. Cortes et al. (2001) proposed the idea of summarizing a series of networks from different time points to represent the network model. This model is extended by Sharan and Neville (2008) to develop a representation that captures the changes in social networks by adding different kernel functions to summarize the weights of links. In our experiment, we follow their idea of a summary representation but propose a practical algorithm to summarize the weights of links from different time periods.

2.2 Link prediction

The link-prediction problem for social networks can be described from a data-mining point of view in the following way: given a snapshot of a social network at time t , the goal is to predict new links that will be added to the network during the interval from time t to a given future time t' .

This problem can be viewed as a simple binary classification problem. That is, for any two potentially linked objects o_i and o_j , predict whether l_{ij} is 1 or 0. However, it is still difficult to completely solve this binary classification problem. Current research aims to measure the degree of similarity and closeness between two target nodes. In terms of the networks this means that not only should they be similar to each other, but they must also be reachable through the network. In other words, the closer and more similar are they, the higher the possibility they have to be connected in the future. Generally, approaches to addressing this problem come from two sides: the attribute information of nodes and the structural properties of social networks.

In the first of these approaches, attribute information is used for link prediction. Popsecul and Ungar (2003) introduces a structured logistic regression model that can make use of relational features to predict the existence of links on citation datasets from CiteSeer. In that experiment, link prediction on citation networks is cast into a citation recommendation problem by gauging the similarities between target publications and existing publications. However, those methods have the limitation that attributes of nodes can only reveal similarity with other nodes, but fail to take into account the concept of “distance”. For example, two people may be quite similar to each other in terms of habits, interests and backgrounds.

However, they cannot be friends if they are located far from each other in geography as they have no chance to meet.

Links are predicted on the basis of graph proximity measures. Among different proximity measures, nearest neighbourhood algorithms have been widely applied. In the experiment of Murata and Moriyasu (2008), they introduced a weighted common neighbour approach and compared its prediction results with common neighbour and the Jaccard coefficient method. Unfortunately, their experimental results have low accuracies, all under 50%. The reason for this is because neighbourhood algorithms can make correct predictions when two nodes are quite close to each other. Some relationships, such as friendship, co-authorship and co-citation are transitive: nodes may be connected in the future if there is a path among them, but these methods may ignore this.

The second approach takes into account the structure of social networks, namely the “distance” between nodes, when devising measures of closeness. One famous approach is the Katz measure (1997) which defines a measure that directly sums over the collection of paths, exponentially damped by their length so as to count short paths more heavily. Another approach uses random walk (Rudnick & Gaspari 2004), to calculate the number of steps from a start point s to the end point e . Because the time to arrive is not in general symmetric, a common way to detect closeness from this probabilistic approach is to consider the commute time $C_{s,e} = T_{s,e} + T_{e,s}$, where $T_{s,e}$ and $T_{e,s}$ are times to move from start to end and end to start respectively. Liben-Nowell and Kleinberg (2007) present an experiment comparing predictors on large co-authorship networks. Their work suggests that information about future interactions can be extracted from the network topology alone and that subtle measures for detecting node proximity can outperform more direct measures. However, the results show that among all predictors, the best is the Katz measure, combining neighbourhood and distance concepts. However, the derived accuracy of 16% is still quite low.

The above research confirms that current methods of link prediction have much room for improvement. In our experiment, we gather information concerning the evolution of a co-authorship network at an Australian university so as to treat nodes differently. Through observing their past actions, we label nodes by their activity and those nodes with high values in activity have a high possibility to connect with others. In order to improve the accuracy further, matrix theory is applied to detect major patterns for predicting results.

3 Dataset

The dataset used for analysis, visualization and explanation of our approach is from the research master enterprise (RME) database of the University of Technology, Sydney. It is a collection of over 60000 records covering all faculties. It provides information about all publications from UTS researchers during the recent six years (2006-2011) including journals, conference papers and proceedings, chapters and books.

The dataset consists of researcher, publication and researcher-publication relationship files for each year, containing 56,535 records in total (Table 1). In our experiments, datasets from 2006 to 2010 are used to summarize the evolution of the co-authorship network on which link prediction is based and the data from

Table 1: Numbers of authors, publications and links per year from the research database of an Australian university.

Year	Authors	Publications	Co-authorship links
2006	2763	1941	5389
2007	2972	1971	5739
2008	3148	2010	5981
2009	3231	1985	5893
2010	3548	2047	6336
2011	3992	2052	6963

Table 2: Structure of the data sets. Column 1 lists file, column 2 the role it plays in the social network and column 3, the type of entries.

Name	Type	Sources
Authors	Node	Professors, lecturers, researchers
Publications	Node	Books, Articles, Chapters, Conference Papers and Reports
Co-authorship	Link	

2011 is used to estimate the accuracy of predictors.

The advantage of this data source over other scientific bibliographic databases such as CiteSeer and the computer sciences bibliographic data source, DBLP, is its integrity. Although the number of records in our dataset is relatively small compared to those scientific bibliographic databases, it contains all types of authors and publications (see Table 2). This means it is easier to capture the process of network evolution. With this information, it is straightforward to detect the activities of researchers.

4 Methodology

This section describes our experimental methodology. The whole process consists of four parts: building co-authorship networks, involvement of network evolution, link prediction and determining the activities of nodes.

4.1 Building co-authorship networks

Firstly, we apply a simple model to build the co-authorship network so as to reveal research collaboration throughout the university. Two researchers are connected if they work on the same publication (see Figure 1). In our graph, co-authorship is a type of mutual relationship and therefore the network is undirected and weighted. The weight for a link represents the number of publications coauthored by the two researchers. Publications are not explicitly represented in the co-authorship network.

4.2 Involvement of network evolution

According to the above model, let us consider a given network $G = (N, L, W)$ where N is the set of nodes while links are in the set $L \subseteq N \times N$ and W contains the weights of links. Conceptually, network evolution can be represented by a series of networks $\Omega = (G_1, \dots, G_t)$, so that $G_t = (N_t, L_t, W_t)$ represents the network at time t . Then the link prediction problem is to predict G_{t+1} based on networks

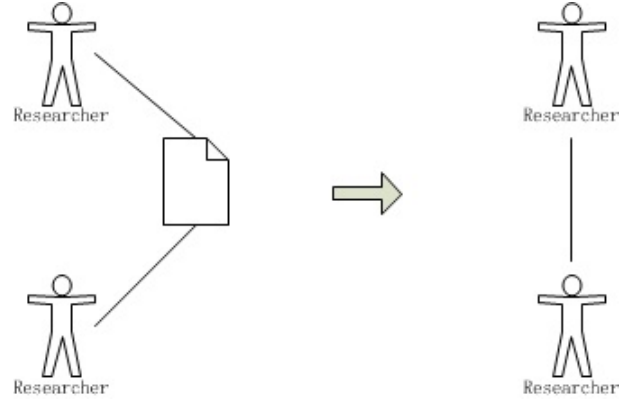


Figure 1: The co-authorship network model. Two researchers coauthoring a publication (left) are directly linked in the network model (right).

G_1, \dots, G_t . Each graph can be represented by an adjacency matrix X of size $n \times n$ where n is the number of nodes and $X_{ij} = X_{ji}$ is equal to the weight of the link between node i and j .

In order to involve the whole history of network evolution into the link prediction problem, the series of matrices associated with the network evolution should be collapsed into a single matrix. The simplest way to do this is to add the matrices together, taking care to ensure that the nodes are the same over the entire time series. However, this does not generally work well in most cases because recent links should have a stronger influence on prediction of future links compared to older links. Here, we propose an approach to collapsing adjacency matrices of networks, motivated by Sharan and Neville's research (2008) on network evolution, where the link structure is damped backward in time according to the following model

$$\begin{aligned} N_{t+1} &= N_1 \cup N_2 \cup \dots \cup N_t \\ L_{t+1} &= L_1 \cup L_2 \cup \dots \cup L_t \\ W_{t+1} &= \sum_{i=1}^t \left(\frac{i}{t}\right)^{t-i} W_i \end{aligned} \quad (1)$$

where W_i is the adjacency matrix for network G_i . Clearly, this method gives greater weight to more recent links.

4.3 Link prediction

Predicting future links is generally based on closeness between nodes. In order to evaluate the effectiveness of different lengths of sequences on network evolution, we adopt several approaches for measuring closeness between nodes.

1. Common neighbours (CN) (Liben-Nowell & Kleinberg 2007) ranks pairs using the number of common neighbours.

$$closeness_{CN}(x, y) = |\Gamma(x) \cap \Gamma(y)| \quad (2)$$

where $\Gamma(x)$ denotes the set of neighbours of node x in a given network and $|\Gamma(x)|$ is the number of elements in the set.

2. Adamic/Adar (AA) (Adamic & Adar 2003) is an extension of common neighbours that adds weights to the neighbours.

$$closeness_{AA}(x, y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log |\Gamma(z)|} \quad (3)$$

where z indexes the common neighbours of nodes x and y .

3. Katz measure (1953) uses a combination of both neighbourhood and distance.

$$closeness_K(x, y) = \sum_{l=1}^{\infty} \beta^l \cdot |paths_{x,y}^l| \quad (4)$$

where $paths_{x,y}^l$ is the set of all l -length paths from x to y and $\beta > 0$ is a scale parameter for the function. β can be regarded as a radius around the target node and predictors can only fetch neighbours from inside the circle formed by this radius. A very small β yields predictions much like common neighbours as the long paths contribute very little to the sum. Due to the fact that a network without node attributes can be represented by its adjacency matrix, the corresponding matrix for Katz's closeness is defined, using the approach in Liben-Nowell & Kleinberg (2007), as

$$P = (I - \beta A_{t+1})^{-1} - I \quad (5)$$

where A_{t+1} is the matrix containing information for the evolution of the network and I is the identity matrix.

However, values of some elements in P are quite small, which means that they have low probability of happening in the future. Sometimes, links associated with large values in P may not happen either as some activity nodes have many close relationships with others. Here we propose a method to improve the prediction accuracy.

The link prediction matrix P can be written as $P = UDV^T$ by the singular value decomposition (SVD). In this formula, R is the rank of P . Matrices U and V are orthogonal matrices of size $M \times R$ and D is a diagonal matrix of singular values $\sigma_1 > \sigma_2 > \dots > \sigma_R > 0$. According to truncated matrix theory

$$P \approx U_k D_k V_k^T \quad (6)$$

where U_k and V_k comprise the first k columns of U and V and D is the $k \times k$ principal submatrix of D .

As a result, a matrix of predicted links can be written as $P' = U_k D_k V_k^T$ which not only contains the main features of matrix P but also excludes the noise that is entailed in the components greater than k .

4.4 Determining activities of nodes

In our experiment, the parameter k is calculated in the following way. We classify nodes in the networks into one of three groups: stable, active and regular. This is done based on their contribution to network evolution as follows.

Stable nodes tend to maintain their relationships as time goes. They may be viewed as inactive or nodes that have already left the network. This means they have low probability of generating new connections with others. Identifying them in advance can promote the efficiency of link prediction.

Active nodes represent those who developed new connections more often during past growth of

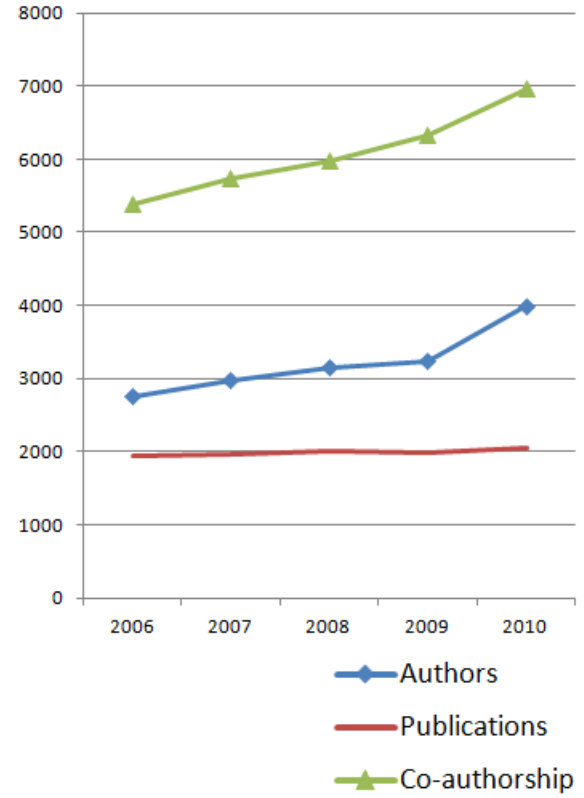


Figure 2: Numbers of authors, publications and co-authorship links per year in the UTS co-authorship network evolution.

networks. They are sometimes the main contributors to network evolution. Therefore, they are of extreme importance in link prediction. Connection to isolated groups may start from them.

Regular nodes are the remainders, containing the rest.

With this definition, the activities of researchers in the network are defined using

$$F(x_k) = \sum_{i=1, j=i+1}^n \left(\frac{L(x_k, t_j) - L(x_k, t_i)}{n} \right) \quad (7)$$

where x_k is a node in the network and the function $L(x_k, t_j)$ is the number of edges of node x_k at time point t_j . If the value of activity is not positive, nodes are labeled as "stable". The value of k used to form the truncated matrix should be the sum of active nodes and regular nodes or the sum of the active nodes only.

5 Results

We use the University of Technology, Sydney, research dataset to assess the performance of the link predictors discussed in section 4.3. All experiments were performed using R (Ripley 2001).

5.1 Building co-authorship networks

We first build the initial co-authorship networks which contain both authors and publications with connections through co-authorship. From Figure 2, it is clear that the numbers of authors and co-authorship

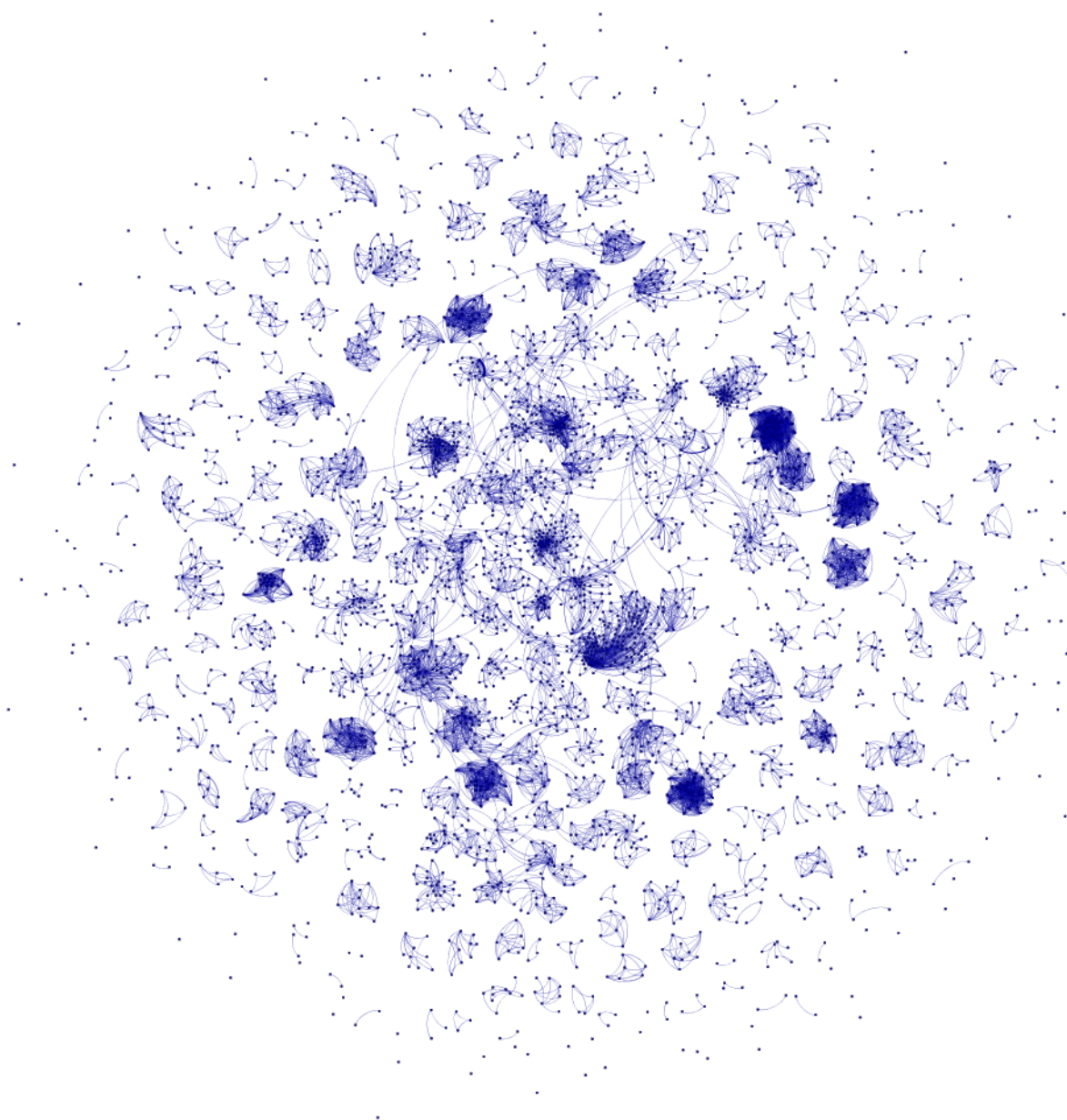


Figure 3: The compacted co-authorship network at UTS from years 2006 to 2010

Table 3: Co-authorship Networks from 2006 to 2010

Year	Nodes	Links	Density of Networks
2006	2763	7115	0.0019
2007	2972	8196	0.0019
2008	3148	9486	0.0019
2009	3231	9243	0.0018
2010	3548	12267	0.0019

relationships grew significantly especially from 2009 to 2010. However the number of publications remained stable, ranging from between 1941 and 2047.

In order to investigate academic collaboration further, we link authors directly by perceiving publications as media according to the network transformation model (see Figure 1). Summary information for the researcher-only co-authorship networks is listed in Table 3. In this table, network densities are calculated as the ratio of the number of edges to the number of possible edges. Although the number of nodes and links increases annually, the network densities stay at almost the same value. This suggests that academic collaboration as measured in co-authorship is relatively stable at UTS.

5.2 Involvement of network evolution

Compacting network evolution into a single matrix is a key step in our experiments. As links from different times contribute differently in link prediction, we assume that the weights, or importance, of links should decrease as time goes on. Using the evolution model presented in Section 4.2, we compacted co-authorship networks from 2006 to 2010 and visualized the resulting network in Figure 3. It shows clear research groups or clusters although there are some isolated authors as well.

5.3 Determining activities of nodes

Determining the activities of nodes in a compacted co-authorship network is a prediction problem that uses the truncated SVD method to exclude noise and improve the accuracies of link prediction. Activities are calculated based on the proposed method described in Section 4.4 and are normalized to lie in $[-1, 1]$. The distribution of activities is illustrated in Figure 4, which plots the activity of each node ranked from lowest to highest. Stable nodes are those where the tangent of the activity function is less than 0 and takes up more than 50% of the authors (3643 active nodes). These are the nodes who left the network during 2006–2010 or who maintain relationships with the same set of partners. Regular authors are those nodes where the tangent is positive but below the red line which is set at an activity level of 0.74. This threshold was set empirically at the point where the curve grew quickly. The nodes above the threshold are active authors who have the highest activities among the three groups.

5.4 Link prediction

Unlike friendship, co-authorship is a transitory relationship. As a result, link prediction in co-authorship networks can be thought of as two problems: predicting new links and predicting all links (including the existing ones). The first problem aims to find whether two authors who have never cooperated with

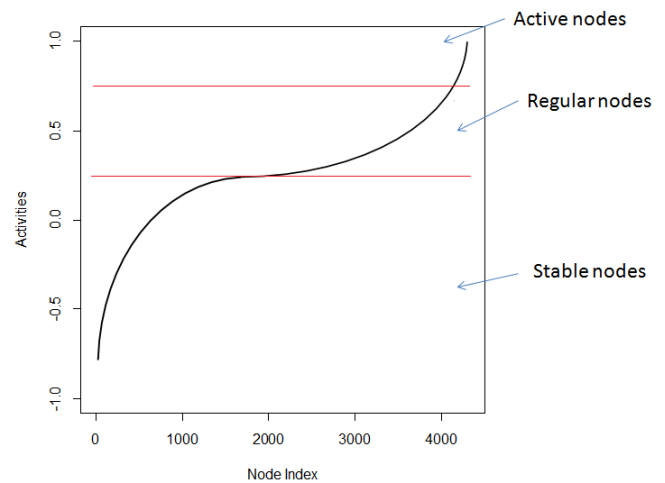


Figure 4: Node activity by node index in increasing order of activity value.

each other will work together in the future based on their closeness. The second problem not only detects the possible new links but also checks whether co-authored nodes will maintain their current relationships into the future. Here, we first apply several baseline algorithms: common neighbourhood (CN), Adamic/Adar (AA) and the Katz method, and compare their predicted results with and without involvement of network evolution and the truncated SVD method. We have observed that different values of β can affect the performance of the Katz measure greatly. Based on preliminary tests on our datasets with $\beta \in [0.005, 0.5]$, we empirically set $\beta = 0.001$ in equations (4) and (5).

We perform link prediction using the three link predictors on different snapshots of the co-authorship networks and compare the predicted results with the real situation in 2011. Accuracies measured by the ratio of correct predictions to all predictions are given in Table 4. These results clearly show, as would be expected, that the more recent networks achieve higher prediction accuracies. Furthermore, involving the evolution of the network by using a compacted network leads to higher accuracies both for predicting new links and all links. Predicting new links is still a challenge as the accuracies are significantly lower than those for predicting all links.

Next, the truncated SVD method is applied to further improve prediction accuracies for the compacted co-authorship network. Results, presented in Table 5, confirm that using the truncated SVD method after identifying the activities of authors can enhance the prediction accuracies in general. For the problem of predicting new links in particular, determining the number of active nodes boosts the prediction accuracies greatly. For prediction of all links, using both active and regular nodes, but not the stable ones, also enhances the prediction accuracy but not by as much. In addition, stable nodes contribute less to the network evolution.

6 Conclusions

This paper describes a novel method for improving the accuracy of link prediction. On the one hand, involving network evolution rather than using a single snapshot of networks provides a more suitable input

Table 4: Performance of predictors on different co-authorship networks. Figures are the ratio of correct predictions to all predictions. CN = common neighbours, AA = Adamic/Adar, Katz = Katz closeness measure.

Co-authorship network	New Links			All Links		
	CN	AA	Katz	CN	AA	Katz
Network of 2006	0.09	0.08	0.11	0.56	0.54	0.61
Network of 2007	0.13	0.11	0.15	0.61	0.62	0.65
Network of 2008	0.14	0.15	0.19	0.69	0.66	0.67
Network of 2009	0.17	0.18	0.23	0.72	0.68	0.71
Network of 2010	0.21	0.20	0.26	0.78	0.76	0.76
Compacted network (from 2006 to 2010)	0.33	0.39	0.42	0.84	0.82	0.88

 Table 5: Performance of predictors using truncated SVD method on compacted co-authorship network from 2006 to 2010. CN, AA and Katz are the closeness measures used, node type is the nodes used and k is the parameter for truncated SVD.

Predictor	Node Type	k	New Links	All Links
CN	Active	278	0.72	0.86
	Active and regular	2873	0.49	0.90
AA	Active nodes	278	0.75	0.82
	Active and regular	2873	0.53	0.88
Katz	Active nodes	278	0.91	0.81
	Active and regular	2873	0.77	0.92

for the prediction methods. On the other hand, the truncated SVD approach can exclude noise and effectively improve prediction accuracies for new links after identifying the amount of activity of authors in networks. The methods were evaluated on six years of research collaboration data from an Australian university. Results demonstrated significant improvements in accuracy using our approach.

In the future, we would like to extend our experimental datasets to larger ones from other sources to test the effectiveness and robustness of our proposed method further. We plan also to develop quantitative approaches to set the threshold between active and regular nodes.

References

- Adamic, L. & Adar, E. (2003), ‘Friends and neighbors on the web’, *Social networks* **25**(3), 211–230.
- Carrington, P., Scott, J. & Wasserman, S. (2005), *Models and methods in social network analysis*, Cambridge University Press.
- Cortes, C., Pregibon, D. & Volinsky, C. (2001), ‘Communities of interest’, *Advances in Intelligent Data Analysis* pp. 105–114.
- Guo, F., Hanneke, S., Fu, W. & Xing, E. (2007), Recovering temporally rewiring networks: A model-based approach, in ‘Proceedings of the 24th International Conference on Machine Learning’, ACM, pp. 321–328.
- Katz, H., Selman, B. & Shah, M. (1997), ‘Referral web: combining social networks and collaborative filtering’, *Communications of the ACM* **40**(3), 63–65.
- Katz, L. (1953), ‘A new status index derived from sociometric analysis’, *Psychometrika* **18**(1), 39–43.
- Liben-Nowell, D. & Kleinberg, J. (2007), ‘The link-prediction problem for social networks’, *Journal of the American Society for Information Science and Technology* **58**(7), 1019–1031.
- Murata, T. & Moriyasu, S. (2008), ‘Link prediction based on structural properties of online social networks’, *New Generation Computing* **26**(3), 245–257.
- Popescul, A. & Ungar, L. (2003), Statistical relational learning for link prediction, in ‘IJCAI workshop on learning statistical models from relational data’, Vol. 2003.
- Ripley, B. (2001), ‘The R project in statistical computing’, *MSOR Connections. The Newsletter of the LTSN Maths, Stats & OR Network* **1**(1), 23–25.
- Rudnick, J. A. & Gaspari, G. D. (2004), *Elements of the random walk: an introduction for advanced students and researchers*, Cambridge University Press.
- Scott, J. & Carrington, P. (2011), *The SAGE Handbook of Social Network Analysis*, SAGE Publications Ltd.
- Sharan, U. & Neville, J. (2008), Temporal-relational classifiers for prediction in evolving domains, in ‘Data Mining, 2008. ICDM’08. Eighth IEEE International Conference on’, IEEE, pp. 540–549.
- Snijders, T. (2002), ‘Markov chain Monte Carlo estimation of exponential random graph models’, *Journal of Social Structure* **3**(2), 1–40.
- Song, L., Kolar, M. & Xing, E. (2009), ‘Time-varying dynamic Bayesian networks’, *Advances in Neural Information Processing Systems* **22**, 1732–1740.

Learning Personalized Tag Ontology from User Tagging Information

Endang Djuana

Yue Xu

Yuefeng Li

School of Electrical Engineering and Computer Science
Queensland University of Technology
GPO Box 2434, Brisbane, QLD 4001

{e.djuanatjhwa, yue.xu, y2.li}@qut.edu.au

Abstract

The cross-sections of the Social Web and the Semantic Web has put *folksonomy* in the spot light for its potential in overcoming knowledge acquisition bottleneck and providing insight for "wisdom of the crowds". *Folksonomy* which comes as the results of collaborative tagging activities has provided insight into user's understanding about Web resources which might be useful for searching and organizing purposes. However, collaborative tagging vocabulary poses some challenges since tags are freely chosen by users and may exhibit *synonymy* and *polysemy* problem. In order to overcome these challenges and boost the potential of *folksonomy* as emergence semantics we propose to consolidate the diverse vocabulary into a consolidated entities and concepts. We propose to extract a tag ontology by ontology learning process to represent the semantics of a tagging community. This paper presents a novel approach to learn the ontology based on the widely used lexical database WordNet. We present personalization strategies to disambiguate the semantics of tags by combining the opinion of WordNet lexicographers and users' tagging behavior together. We provide empirical evaluations by using the semantic information contained in the ontology in a tag recommendation experiment. The results show that by using the semantic relationships on the ontology the accuracy of the tag recommender has been improved.

Keywords: collaborative tagging, *folksonomy*, ontology learning, personalization, tag recommendation

1 Introduction

The development of World Wide Web has led the research activities into cross-sections of two worlds: the Social Web and the Semantic Web. The Social Web is represented by a class of web sites and applications in which user participation is the primary driver of value which often referred by the phrase "collective intelligence" or "wisdom of crowds" to refer to the value created by the collective contributions of all these people (Gruber 2008). This trend was firstly mentioned in article by O'Reilly (2005) as Web 2.0.

The Semantic Web is an extension of the existing World Wide Web. It provides a standardized way of expressing the relationships between web pages, to allow machines to understand the meaning of hyperlinked information (Berners-Lee 2001). This may create the "web of data" in which metadata in the form of ontology, explicit specification of the conceptualization of a domain (Gruber 1993), plays important role in achieving this vision.

However, after several years on, this vision still has challenges due to knowledge acquisition bottleneck such as development and maintenance of ontologies. Ontology learning has been developed to overcome this barrier (Maedche and Staab 2001). Ontology learning or semi-automatic way of constructing ontology relies on machine learning and automated language-processing techniques to extract concepts and ontological relations from structured or unstructured data such as database and text (Navigli, Velardi and Gangemi 2003).

Folksonomy (Vander Wal 2005) which is emerging from collaborative tagging activities has been acknowledged as potential source for constructing ontology, as they capture the vocabulary of the users which may be aggregated to produce emergent semantics, from which people may develop lightweight ontologies (Mika 2007). The growing availability of *folksonomies* has motivated the work introduced in this paper for constructing lightweight ontology from collaborative tagging data.

User tagging or collaborative tagging describes the process by which many users add metadata in the form of keywords to Internet resources with a freely chosen set of keywords (tags) (Marlow et al 2006, Golder and Huberman 2006).

Research works have been conducted in utilizing tagging information to improve searching, clustering, and recommendation making. However, collaborative tagging vocabulary poses some challenges since tags are freely chosen by users and may exhibit *synonymy* and *polysemy* problem. Moreover, the relationships among tags haven't been maximally utilized, which could provide valuable information us to better understand users since there exists rich relationships among tags.

In this paper we present our approach to construct personalized tag ontology based on user tagging information and the widely used general knowledge ontology WordNet (Fellbaum 1998). We begin by introducing the background of user tagging collection and the main motivation for this work in Section 2. We then review related works in Section 3. In Section 4 we introduce our ontology learning approach including the ontology personalization approach. In Section 5 we present novel methods for improving tag recommendation based on the proposed tag ontology. In Section 6 we

present an experiment and the initial results. Section 7 concludes this paper and gives some ideas for further work.

2 Key Concept and Motivation

2.1 User Tagging

A user tagging collection involves three entities: items, tags, and users, which are described below:

- Users $U = \{u_1, u_2, \dots, u_{|U|}\}$ contains all users in an online community who have used tags to organize their items.
- Tags $T = \{t_1, t_2, \dots, t_{|T|}\}$ contains all tags used by the users in U . Tags are typically arbitrary strings which could be a single word or short phrase.

In this paper, a tag is defined as a sequence of terms.

For $t \in T$, $t = \langle term_1, term_2, \dots, term_m \rangle$. A function is defined to return the terms in a tag:

$$tagset(t) = \{term_1, term_2, \dots, term_m\}$$

- Items $I = \{i_1, i_2, \dots, i_{|I|}\}$ contains all domain-relevant items or resources. What is considered by an item depends on the type of user tagging collection, for instance, in Amazon.com the items are mainly books.

Based on the three entities, a user tagging collection or a collaborative tagging system is formulated as 4-tuple: $F = (U, T, I, Y)$ (Jaschke et al 2008) where U, T, I are finite sets, whose elements are the users, tags and items, respectively. Y is a ternary relation between them, i.e., $Y \subseteq U \times T \times I$, whose elements are called tag assignments or taggings. An element $(u, t, i) \in Y$ represents that user u collected item i using tag t .

Tags in a tag collection may exhibit many variations such as *synonymy* where different tags may have the same or closely related meanings. Different users may tag an item using different tags which have similar meaning. The other variation is *polysemy* where one tag has multiple meanings. A tag may be used by different users to tag different items that are not related to each other at all. Moreover, one tag may have semantic relationship to other tags, e.g. “inn” is a kind of “hotel” which shows the two tags are related with each other and “inn” has “more specific” meaning. This condition may not be utilized to relate items collected under these two tags because they are simply treated as two different tags.

2.2 Motivation

Many methods have been proposed to deal with the problems of synonymy and polysemy (Bischoff et al 2008, Suchanek and Vojnovic and Gunawardena 2008, Liang et al 2010). There are several works which try to infer relationship between tags (Tang et al 2009, Liu, Fang and Zhang 2010). However, these works mostly didn't base the inference on semantic measure but on statistical measure which may fail to capture the semantic relationships among tags. Also, the semantic relationships between tags need to be exploited more by existing tagging based applications including tag based recommenders.

In order to tackle these problems, it becomes desirable to find a way to consolidate the multiple facets and the relationships of tags into a consolidated entity which will help better understand the tags used by users. There are several possible solutions include using

classification systems such as taxonomy or using conceptualization systems such as ontology. In this work we consider to use ontology to represent the semantics in tags collection because of the flexibility of an ontology and possibility of emerging semantics from the ontology learning process (Mika 2007, Lin, Davis and Zhou 2009).

3 Related Works

Work by Garcia-Silva et al (2012) compares most relevant approaches for associating tags with semantics in order to make explicit the meaning of those tags. They have identified three group of approaches which are based on 1) clustering techniques i.e. to cluster tags according to some relations among them (statistical techniques); 2) ontologies i.e. aiming at associating semantic entities e.g. WordNet, Wikipedia, to tags as a way to formally define their meaning; 3) hybrid approach i.e. mixing clustering techniques and ontologies. Our work falls into the second group which is based on ontologies.

Beside our work there are several works which tried to extract ontological structures from user tagging systems. Lin, Davis and Zhou (2009) extracted ontological structures by exploiting low support association rule mining supplemented by WordNet. Trabelsi, Jrad and Yahia (2010) focused more on extracting non-taxonomic relationships from folksonomies using triadic concepts with external resources: WordNet, Wikipedia and Google.

Tang et al (2009) and Liu, Fang and Zhang (2010) represents state of the art work for generating ontology from folksonomy based on generative probabilistic models i.e. tag-topic model and set-theoretical approach i.e. to produce tag subsumption graph respectively. Most of this works did not provide applications for the ontology such as tag recommendation.

As for the work in collaborative tag recommendation there are several notable works such as work by Sigurbjornsson, van Zwol and D'Silva (2008) which is based on tag co-occurrences. Although this work has achieved good result, it didn't rely on the actual meaning of tags which may miss the semantic relationships among tags.

Beside our work there are several works which utilize some format of ontology to assist in tag recommendation task. Baruzzo et al (2009) used existing domain ontology to recommend new tags by analyzing textual content of a resource needed to be tagged. They relied on existing domain ontology which is not always available for a particular domain and also they didn't provide quantitative evaluation.

Tag recommendation approach by Tatu, Srikanth, D'Silva (2008) by mapping textual contents in Bibsonomy bookmarks, not just the tags to form conflated tags to normalized concepts in WordNet and similar approach by Lipczak et al (2009) which explored resource content as well as resource and user profiles are comprehensive. There is a drawback that they relied on extended textual contents provided by Bibsonomy which are not always available in other user tagging systems.

4 Ontology Learning from User Tagging

One stream of approach to the ontology construction relies on machine learning and automated language-processing techniques to extract concepts and ontological relations from structured or unstructured data such as database and text (Navigli, Velardi and Gangemi 2003).

In this work we propose to construct the tag ontology based on some existing ontology, which we call backbone ontology. The basic idea is to take advantage of hierarchies of concepts in the backbone ontology and to form the tag ontology by mapping the tags in the tag collection to the concepts on the backbone ontology and extracting the available relationships among concepts in the backbone ontology.

The lexical knowledge base WordNet (Fellbaum 1998) was chosen in this paper as the backbone ontology as it has wide coverage of concepts (over 200,000) and richness of relationships such as semantic relationships “is-a”, “part-of”, lexical relationships “synonymy” and “antonymy” as well as availability of accompanying corpus and other facility for disambiguation process. The backbone ontology is defined below.

Definition 1 (Backbone ontology): The backbone ontology is defined as a 2-tuple $BackboneONTO = (C, R)$ where $C = \{c_1, c_2, \dots, c_{|C|}\}$ is a set of concepts; $R = \{r_1, r_2, \dots, r_{|R|}\}$ is a set of relations representing the relationships between concepts.

A concept c in C is a 3-tuple $c = (id, synset, category)$ where id is a unique identification assigned by WordNet system to the concept c ; $synset$ is a synonym set containing synonymic terms which represent the meaning of the concept c ; and $category$ is a lexical category assigned by WordNet lexicographers to classify this concept c into a general category. A relation r in the relation set R is a 3-tuple $r = (type, x, y)$, where $type \in \{is_a, part_of, \dots\}$; $x, y \in C$ are the concepts that hold the relation r .

For easy to describe the work, we denote the set of synonyms representing c by $synset(c)$ and the category of c by $category(c)$. For each term w in $synset(c)$, w is represented as a 2-tuple $(w, freq_c(w))$ where w is a synonym term of the concept c ; $freq_c(w)$ is the frequency assigned by WordNet lexicographers to the term as an indication of how frequently this term has been used to represent the meaning of the concept c based on the accompanying WordNet corpus. For a term w , the set of concepts for which w is a synonymic term is defined as $con(w) = \{c | (w, f) \in synset(c)\}$.

4.1 Mapping Tags to Concepts

One tag may contain one or more terms. It is possible that a tag can be mapped directly to one or more concepts in the backbone ontology. It is also possible that only part of a tag may map to one or more concepts. We propose the following mappings to deal with different cases.

There are 3 different cases for finding possible mappings for a given tag, which are: (1) mapping the full tag to one or more concepts; (2) mapping part of the tag to one or more concepts; and (3) splitting the tag into a list of single words, then mapping each of the words to concepts separately. Readers are referred for a more detailed discussion for each case from previous publications in Djuana, Xu and Li (2011).

1. Direct Mapping

We define the following function to represent the whole mapping from a tag to concepts:

$$Tag_Concept_{whole}: T \rightarrow 2^C$$

$$\begin{aligned} \forall t \in T, Tag_Concept_{whole}(t) \\ = \{c | \forall c \in C, \exists (w, f) \in synset(c), t == w\} \end{aligned}$$

$Tag_Concept_{whole}(t)$ is a set of concepts for each of which t is synset term.

2. Partial Mapping

The following function represents the partial mapping from a tag to concepts: $Tag_Concept_{partial}: T \rightarrow 2^C$

$$\begin{aligned} \forall t \in T, Tag_Concept_{partial}(t) \\ = \{c | \forall c \in C, \exists (w, f) \in synset(c), MaxPostfix(t) == w\} \end{aligned}$$

$MaxPostfix(t)$ stands for the largest postfix of t .

3. Term Mapping

The following function represents the term mapping from a tag to concepts: $Tag_Concept_{term}: T \rightarrow 2^C$

$$\begin{aligned} \forall t \in T, Tag_Concept_{term}(t) \\ = \sum_{a \in tagset(t)} Tag_Concept_{whole}(a) \end{aligned}$$

Overall, $\forall t \in T$, the tag to concept mapping is defined as follows:

$$Tag_Concept(t) = \begin{cases} Tab_Concept_{whole}(t) & \text{if } t \text{ is directly mapped} \\ Tag_Concept_{partial}(t) & \text{partially mapped} \\ Tag_Concept_{term}(t) & \text{term mapped} \end{cases} \quad (1)$$

4.2 Mapping Disambiguation

A tag can be mapped to multiple concepts. After all the possible mappings are found, we need to choose the most appropriate concept from the mapped concepts to represent the meaning of the tag for this particular tag collection.

For disambiguating the concepts, we propose to measure the strength of the mapping by using the word frequency provided by WordNet. A matrix $T_C[t_i, c_j]_{m \times n}$ is defined to represent the strength of the mapping between tags and concepts, where $m=|T|$ and $n=|C|$. In order to make the frequency comparable between different concepts, we normalize the frequency value to a scale of [0, 1]. The mapping strength based on frequency is defined below:

$$T_C_{frequency}[t_i, c_j] = \begin{cases} \frac{f_{c_j}(t_i)}{\sum_{c_k \in Tag_Concept(t_i)} f_{c_k}(t_i)} & c_j \in Tag_Concept(t_i) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

For a tag t_i , the concept c_j should be chosen as t_i 's concept if $T_C[t_i, c_j]$ is the highest value for all $c_j \in Tag_Concept(t_i)$. After the disambiguation, each tag t will be mapped to one and only one concept. This can be defined by a one to one disambiguation mapping $M_{frequency}: T \rightarrow C$

$$M_{frequency}(t) = \underset{c \in Tag_Concept(t)}{\operatorname{argmax}} (T_C_{frequency}[t, c]) \quad (3)$$

On the other hand, multiple tags may be mapped to one concept. The following function defines the mapping from a concept to tags: $Concept_Tag: C \rightarrow 2^T$
 $Concept_Tag(c) = \{t | \forall t \in T, M_{frequency}(t) == c\}$

4.3 Relationship Extraction Process

After the mapping and disambiguation processes, each tag will be mapped to a concept on the backbone ontology. Based on the mappings, we retrieve the available relationships (“is-a” relations) from the mapped concept c consecutively until we reach the top of the hierarchy. This operation is the same operation as finding an ancestor in a tree-based structure. The top of the hierarchy in the backbone ontology is a general category defined by WordNet.

We can then extract the mapped concepts together with the relationships in the backbone ontology to form the tag ontology. As the result of the tag to concept mapping and the relationships extraction, we can construct the tag ontology which is defined as below:

Definition 2 (Tag Ontology): The tag ontology is defined as 2-tuple $TagOnto = (TC, TR)$ where $TC = \{tc_1, tc_2, \dots, tc_{|TC|}\}$ is a set of tag-concepts, i.e., $\subseteq C \times 2^T$, and $TR = \{tr_1, tr_2, \dots, tr_{|TR|}\}$ is a set of tag relations. Each element in TC is a pair of a concept c and a set of tags $\{t_1, t_2, \dots, t_n\}$, i.e., $tc = (c, \{t_1, t_2, \dots, t_n\}) \in TC$, which represents that each tag in $\{t_1, t_2, \dots, t_n\}$ can be mapped to concept c . TR is defined as:

$$TR = \left\{ r = (type, c_1, c_2) \left| \begin{array}{l} r \in R, \\ Concept_Tag(c_1) \neq \emptyset, \\ Concept_Tag(c_2) \neq \emptyset \end{array} \right. \right\}$$

4.4 Personalization in Mapping Disambiguation

The tag ontology constructed using the approach described in previous sections mainly utilizes the structural information between concepts and the frequencies of synset terms provided by WordNet. The tag-to-concept mapping is mainly determined based on the synset term frequencies which are derived based on WordNet corpus.

However, for a given tagging collection, the synset term frequencies may not adequately reflect the interests of the users in this particular collection. To reduce the bias caused by solely using the synset term frequency, we propose to take user tagging information into consideration in disambiguating the mapping from tags to concepts.

Let (U, T, I, Y) be a tagging system, the following strategy is proposed to generate personalized tag ontology for users in U . The personalization in the context of this paper is for a tagging community rather than for individual users. The idea here is to find tag relevance based on the tagging information of users in a tagging community and then map tags onto the backbone ontology based on the tag relevance.

In WordNet, each concept is assigned into one and only one category. Let CA denote the set of categories in WordNet ontology, for a concept $c \in C$, $\varepsilon(c) \in CA$ is defined as the only category assigned to the concept c . Different concepts can be categorized into one category.

On the other hand, for a category Ca , it may have multiple concepts. A function $concept(Ca) = \{c | \forall c \in C, \varepsilon(c) == Ca\}$, is defined to return all the concepts that belong to category Ca .

Moreover, the categories of a tag t can be obtained from the category of t 's concepts (i.e., $Tag_Concept(t)$).

The set of categories of a given tag t is defined as: $category(t) = \{\varepsilon(c) | c \in Tag_Concept(t)\}$. A category can have multiple concepts. Similarly, a category Ca can have multiple tags which belong to Ca . A function $tag(Ca) = \{t | \forall t \in T, Ca \in category(t)\}$ is defined to return all the tags that belong to category Ca .

For an item, different users may collect it using different tags and these tags must have something in common which reflects some characteristic of the item. Therefore, by looking at the tags that have been used by users in U to tag the same items, we can find related tags with respect to the users in U . For a given tag $t \in T$, the related tags of t is defined by the following equation:

$$t_related(t) = \{t_j | \forall i \in I_t, \exists t_j \in T_i, \exists u \in U, (u, t_j i) \in Y\} \quad (4)$$

where I_t is a set of items that are collected by users with tag t , T_i is a set of tags that are used by users to tag item i .

In this paper, we propose to estimate the relevance between a tag t_i and a concept c_j by exploiting the relevance between the tag and its t -related tags that belong to the same category of c_j to measure the strength from t_i to the concept c_j . Let $p(t_i | t_k)$ represent the probability of using t_i to tag some items given that t_k has been used to tag the items. If $p(t_i | t_k)$ is high, it can be considered that t_i is highly relevant to t_k .

We propose the following equation to measure the relevance of a tag to a concept based on the relevance of the tag to its related tags that belong to the same category of this concept:

$$\begin{aligned} t_relevance(t_i, c_j) \\ = \sum_{t_k \in t_related(t_i) \cap tag(category(c_j))} p(t_i | t_k) \end{aligned} \quad (5)$$

Given tags t_i and t_k , the probability of using t_i and t_k to tag an item a can be calculated by the equation:

$$p(a | t_i, t_k) = \frac{p(t_i | a, t_k) p(a | t_k)}{p(t_i | t_k)}, \text{ from which, we can get}$$

the following equation to calculate $p(t_i | t_k)$:

$$p(t_i | t_k) = \sum_{a \in I} p(t_i | a, t_k) p(a | t_k) \quad (6)$$

Let $UI_{t_j} = \{(u_i, i_k) | \forall u_i \in U, \forall i_k \in I, (u_i, t_j, i_k) \in Y\}$ be a set of user-item pairs each of which represents that a user tags an item using tag t_j (i.e., the tag assignments using t_j); $U_{t_j, i_k} = \{u_i | \forall u_i \in U, (u_i, t_j, i_k) \in Y\}$ be a set of users who have used tag t_j to tag item i_k .

For a given tag t , the probability of using t by any user to tag any item, denoted as $p(t)$, can be defined as the ratio between the number of tag assignments using t and the total number of tag assignments, i.e., $p(t) = \frac{|UI_t|}{|Y|}$.

The probability of using tag t to tag item a by any users can be defined as the ratio between the number of users

who used t to tag a and the total number of tag assignments, i.e., $p(t, a) = \frac{|U_{t,a}|}{|Y|}$.

Similarly, $p(t_1, t_2, a) = \frac{|U_{t_1,a} \cap U_{t_2,a}|}{|Y|}$, it is the ratio between the number of users who have used both t_1 and t_2 to tag item a and the total number of tag assignments.

Based on these probabilities, we can calculate the two probabilities, $p(a|t)$ and $p(t_1|a, t_2)$, as:

$$p(a|t) = \frac{p(t,a)}{p(t)} = \frac{|U_{t,a}|}{|U_{t,*}|}$$

$$p(t_1|a, t_2) = \frac{p(t_1, t_2, a)}{p(t_2, a)} = \frac{|U_{t_1,a} \cap U_{t_2,a}|}{|U_{t_2,a}|}$$

Thus, equation (6) becomes:

$$p(t_i|t_k) = \sum_{a \in I} \frac{|U_{t_i,a} \cap U_{t_k,a}|}{|U_{t_k,a}|} \quad (7)$$

With Equation (7), we can calculate the relevance between a tag and a concept using Equation (5). The normalized tag relevance is used to measure the relevancy from a tag to a concept. $T_C_{relevance}[t_i, c_j]_{m \times n}$ is defined as below:

$$T_C_{relevance}[t_i, c_j] = \frac{t_relevance(t_i, c_j)}{\sum_{c \in \text{Tag_Concept}(t_i)} t_relevance(t_i, c)} \quad (8)$$

For different sets of users, $T_C_{relevance}[t_i, c_j]$ can be different because they are based on user tagging information, while $T_C_{frequency}[t_i, c_j]$ will be the same for all user sets because it is based on the term frequency provided by WordNet.

The mapping disambiguation based on tag relevancy can be defined as $M_{relevance} : T \rightarrow C$

$$M_{relevance}(t) = \underset{c \in \text{Tag_concept}(t)}{\text{argmax}} (T_C_{relevance}[t, c]) \quad (9)$$

5 Tag Recommendation based on Tag Ontology

A tag recommender is a specific kind of recommender systems in which the goal is to suggest a set of tags for a user to use for tagging a particular item. One of our goals in this paper is to investigate whether the semantic information captured in the constructed tog ontology can be utilized to improve the accuracy of tag recommendation.

The task of a tag recommender system is to recommend, for a given user $u \in U$ and a given item $i \in I$ which has not been tagged by the user, a set $\tilde{T}(u, i) \subseteq T$ of tags. In many cases $\tilde{T}(u, i)$ is computed by first generating a ranking on the set of tags according to some criterion, from which then the top n tags are selected.

5.1 CF based Tag Recommendation

A tag recommender has been proposed in (Jaschke et al 2008) which is based on the user-based CF method. To recommend tags to a target user for tagging an item, it first finds the neighbor users of the target user, then generates a set of candidate tags which have been used by the

neighbor users to tag the item and finally rank the candidate tags based on the similarity between the target user and neighbor users to decide the top n tags as the final recommendations.

Let $CT(u, i)$ be a set of candidate tags which have been used by u 's neighbors to tag item i . For a candidate tag t in $CT(u, i)$, its ranking can be calculated by the following equation:

$$w(u, t, i) = \sum_{v \in N_u^k} \text{sim}(\tilde{x}_u, \tilde{x}_v) * \delta(v, t, i),$$

$$\delta(v, t, i) = \begin{cases} 1 & (v, t, i) \in Y \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

where $\text{sim}(\tilde{x}_u, \tilde{x}_v)$ is the similarity of users, N_u^k is user u 's neighborhood containing k similar users, $\delta(v, t, i)=1$ indicates the user v has used this tag t to tag the item i . The top n tags, denoted as $T(u, i)$, can be determined based on the ranking:

$$T(u, i) = \underset{t \in T}{\text{argmax}}^n w(u, t, i) \quad (11)$$

5.2 Tag Recommendation based on Tag Ontology

Having the tag ontology in place we can explore the concept representation of a tag, its placement in the hierarchy and its relationships to other concepts. This brought us an idea to improve the recommendations in $T(u, i)$ based on the semantic information in the extracted ontology to see if the ontology can directly improve tag recommendations.

In the proposed method, we generate candidate tags based on neighbour users' preference and the synset information captured in the tag ontology as well, and rank the candidate tags based on both user similarity and tag popularity.

5.2.1 Candidate tag expansion

Let $CT(u, i)$ be the set of candidate tags generated based on neighbor users' preferences. For each candidate tag t in $CT(u, i)$, by using the disambiguation mapping methods given in Equation (3) or (9), t can be mapped to concepts $M_{frequency}(t)$ or $M_{relevance}(t)$ in the tag ontology, respectively. From the synset terms of the mapped concepts, two expanded sets of candidate tags can be generated based on the two methods:

$$CT_{frequency}(u, i) = \bigcup_{t \in CT(u, i)} \text{synset}(M_{frequency}(t))$$

$$CT_{relevance}(u, i) = \bigcup_{t \in CT(u, i)} \text{synset}(M_{relevance}(t))$$

5.2.2 Recommendation ranking

For each of the candidate tag t in $CT_{frequency}(u, i)$ or $CT_{relevance}(u, i)$, its ranking is calculated by the following equation:

$$w_y(u, t, i) = \begin{cases} \sum_{v \in N_u^k} \text{sim}(\tilde{x}_u, \tilde{x}_v) * \delta(v, t, i) & t \in CT(u, i) \\ \sum_{v \in N_u^k} \text{sim}(\tilde{x}_u, \tilde{x}_v) * \delta(v, t, i) * \mathcal{P}(t) & t \notin CT(u, i), t \in CT_y(u, i) \end{cases} \quad (12)$$

where $\gamma \in \{frequency, relevance\}$ and $\mathcal{P}(t)$ is the popularity of tag t , which is calculated as: $\mathcal{P}(t) = |UI_t| / \max_{t_i \in T} |UI_{t_i}|$.

As defined in Section 4.4, UI_t contains (user, item) pairs representing the tag assignments using tag t . $|UI_t|$ is the number of times that t has been used to tag items. The higher the $|UI_t|$, the more popular the tag t is.

$\mathcal{P}(t)$ is the ratio between $|UI_t|$ and the maximum number of times that a tag has been used to tag items in this tagging community.

Based on the two disambiguation methods, we can generate two lists of tags ranked by using Equation (12). Thus, two lists of top n tags can be determined based on the ranking:

$$T_{frequency}(u, i) = \operatorname{argmax}_{t \in T} w_{frequency}(u, t, i) \quad (13)$$

$$T_{relevance}(u, i) = \operatorname{argmax}_{t \in T} w_{relevance}(u, t, i) \quad (13)$$

In our experiments to be discussed below, the accuracy of recommendations using the result in (13), (14), or the combination of the two has been compared.

6 Evaluation

6.1 Experiment Setup

We have conducted experiments to evaluate the usefulness of the proposed tag ontology in making tag recommendations. Two datasets are used in the experiments:

(1). The Bibsonomy dataset used in ECML PKDD Discover Challenge 2009 (<http://www.kde.cs.uni-kassel.de/ws/dc09/>). The dataset contains public bookmarks and publication posts of Bibsonomy. The dataset that used in this experiment contains 1122 users, 19682 items and 6517 tags.

(2). The publicly available Delicious dataset (Wetzker, Zimmermann and Bauckhage 2008). The dataset contains all public bookmarks of users posted on delicious.com between September 2003 and December 2007. In this paper a portion of the data set is used which contains bookmarks from January to March 2004. This portion contains 1289 users, 863 items (URLs) and 215 tags.

Each of the datasets is split into a testing dataset and a training dataset based on posting date. The split percentage is 25% for testing dataset which is taken from newer posts and 75% for training dataset which is taken from older posts. This is to simulate the actual tag recommendation scenario in which users are normally given a recommendation list based on what tags previously stored in the system.

In the experiments we conducted 5 folds cross validation for all the users in the dataset. In each run of the experiment, we randomly take 20% portion as the target users while the remaining 80% is taken as the training users from whom we calculate similarities to the target users to find neighbors. The top n tags are recommended to each target user for each of the user's items in the testing set. The recommended tags are compared to the target user's actual tags of the items in the testing dataset. If a recommended tag matches with an actual tag, we calculate this as a hit. The standard precision and recall are used to evaluate the accuracy of tag recommendations.

6.2 Results

We have conducted the following runs to compare the performance between the baseline recommender, the user based CF method, and the proposed methods:

- User-CF: this is the user based CF tag recommender system proposed in (Jäschke et al 2008).
- Exp_Freq: this is the proposed method to expand candidate tags by using synset terms of the tag ontology mapped based on synset term frequency.
- Exp_Rel: this is the proposed method to expand candidate tags by using synset terms of the tag ontology mapped based on tag relevance.
- Freq&Rel: this method generates the tag recommendations by combining the results of Exp_Freq and Exp_Rel and selecting the top n tags.

The results of the experiments are presented in Table I to Table IV for Bibsonomy and Delicious datasets, respectively. As shown in these tables, the use of the ontology has improved the precision and recall for all the two datasets. From the results, we can see that, the Exp_Rel run achieved better results than that of Exp_Freq run, which means that the tag relevance generated based on user tagging behavior of the users in this tagging community is more useful than the term frequency given by WordNet lexicographers. The former reflects the specific perspective of the users in this particular community, while the latter reflects the general viewpoint of lexicographers. Especially, the combination of the two methods outperforms all the other methods. From the results of this experiment, we can say that the tag ontology can be used to improve the performance of recommendation.

N	5	10	15	20
User-CF	0.183	0.103	0.070	0.052
Exp_Freq	0.191	0.109	0.075	0.056
Exp_Rel	0.191	0.110	0.075	0.056
Freq&Rel	0.201	0.126	0.091	0.072

Table 1: Precision for Bibsonomy dataset

N	5	10	15	20
User-CF	0.435	0.474	0.479	0.479
Exp_Freq	0.445	0.489	0.491	0.50
Exp_Rel	0.445	0.491	0.50	0.52
Freq&Rel	0.481	0.513	0.531	0.561

Table 2: Recall for Bibsonomy dataset

N	5	10	15	20
User-CF	0.169	0.081	0.072	0.054
Exp_Freq	0.176	0.095	0.063	0.047
Exp_Rel	0.176	0.096	0.065	0.047
Freq&Rel	0.183	0.104	0.072	0.049

Table 3: Precision for Delicious dataset

N	5	10	15	20
User-CF	0.609	0.655	0.656	0.655
Exp_Freq	0.639	0.681	0.682	0.680
Exp_Rel	0.639	0.683	0.685	0.689
Freq&Rel	0.641	0.697	0.703	0.711

Table 4: Recall for Delicious dataset

7 Conclusion

Tagging is getting more and more popular in many Web sites. It provides useful data for better understanding users' information needs. The user self-defined tags not only reflect users' understanding to the content of the tagged items, but also provide rich information about item hierarchical classification.

In this paper, we proposed a novel approach to construct tag ontology from user tagging information to represent the semantic meaning and hierarchical relationship among tags. We believe the constructed tag ontology can be used in many applications such as item classification, item recommendation, and tag recommendation. In this paper, we presented a primary experiment to show the improvement to tag recommendation based on the tag ontology. There is room to further improve the recommendation by applying further the extracted ontology structural information in the process of generating recommendation.

8 References

- Baruzzo, A., Dattolo, A., Pudota, N. and Tasso, C. (2009), Recommending new tags using domain-ontologies, *Proc. of IEEE/WIC/ACM Web Intelligence and Intelligent Agent Technology-Workshops*, 409-412, IEEE.
- Berners-Lee, T. (2001), The Semantic Web. Scientific American, <http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21>. Accessed 19 Oct 2012.
- Bischoff, K., Firan, C.S., Nejd, W., and Paiu, R. (2008), Can all tags be used for search? In *Proc. ACM Conference on Information and Knowledge Management*, 193-202, ACM Press.
- Djuana, E., Xu, Y. and Li, Y. (2011), Constructing tag ontology from folksonomy based on WordNet, In *Proc. IADIS International Conference on Internet Technologies and Society*, IADIS. (In press)
- Fellbaum, C. (ed.) (1998), *WordNet: An Electronic Lexical Database*, Cambridge, MA: MIT Press.
- García-Silva, A., Corcho, O., Alani, H., Gómez-Pérez, A. (2012), Review of the state of the art: Discovering and associating semantics to tags in folksonomies, *The Knowledge Engineering Review*, Vol 27(01) 57-85.
- Golder, S. and Huberman, B. (2006), The structure of collaborative tagging systems, HP Labs Tech. Report, <http://www.hpl.hp.com/research/scl/papers/tags/tags.pdf>, Accessed 19 Oct 2012.
- Gruber, T.R. (1993), Towards principles for the design of ontologies used for knowledge sharing, In *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Guarino, N., Poli, R. (Eds.), Kluwer Academic Publishers.
- Jaschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L., and Stumme, G. (2008), Tag recommendations in social bookmarking systems, *AI Communications*, vol 21, 231-247, IOS Press.
- Liang, H., Xu, Y., Li, Y., Nayak, R., Tao, X. (2010), Connecting users and items with weighted tags for personalized item recommendations. In *Proc. 21st ACM Conference on Hypertext and Hypermedia*, 51-60, ACM.
- Lin, H., Davis, J., and Zhou, Y. (2009), An integrated approach to extracting ontological structures from folksonomies, *The Semantic Web: Research and Applications*, 654-668, Springer.
- Lipczak, M., Hu, Y., Kollet, Y., Milios, E. (2009), Tag sources for recommendation in collaborative tagging systems, In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases Discovery Challenge*.
- Liu, K., Fang, B., Zhang, W. (2010), Ontology emergence from folksonomies, In *Proc. ACM International Conference on Information and Knowledge Management*, 1109-1118, ACM Press.
- Maedche, A. and Staab, S. (2001), Ontology learning for the Semantic Web. *IEEE Intelligent Systems* 16(2), 72-79, IEEE.
- Marlow, C., Naaman, M., Boyd, D. and Davis, M. (2006), HT06, tagging paper, taxonomy, Flickr, academic article, to read, In *Proc. ACM Hypertext and Hypermedia*, 31-40, ACM Press.
- Mika, P. (2007), Ontologies are us: A unified model of social networks and semantics. *Web Semantics*. 5(1), 5-15, Elsevier.
- Navigli, R., Velardi, P., and Gangemi, A. (2003), Ontology learning and its Application to automated terminology translation", In *IEEE Intelligent Systems*, vol 18(1), 22-31, IEEE.
- O'Reilly, T. (2005), What is Web2.0: Design patterns and business models for the next generation of software, <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>, Accessed 19 Oct 2012.
- Sigurbjörnsson, B., van Zwol, R. (2008), Flickr tag recommendation based on collective knowledge, In *Proc. World Wide Web Conference*, 327-336, W3C.
- Suchanek, F. M., Vojnovic, M., and Gunawardena, D. (2008), Social tags: meaning and suggestions. In *Proc. ACM Conference on Information and Knowledge Management*, 223-232, ACM Press.
- Tang, J., Leung, H., Luo, Q., Chen, D., Gong, J. (2009), Towards ontology learning from folksonomies, In *Proc. 21st International Joint Conference on Artificial Intelligence*, 2089-2094, AAAI Press.
- Tatu, M., Srikanth, M. and D'Silva, T. (2008): Tag recommendations using bookmark content. In *Proceedings of RSDC'08*, 96-107.
- Trabelsi, C., Jrad, A.B., and Yahia, S.B. (2010), Bridging folksonomies and domain ontologies: Getting out non-taxonomic relations, *Proc. IEEE International Conference on Data Mining Workshops*, 369-379 IEEE.
- Vander Wal, T. (2005), Folksonomy coinage and definition, <http://vanderwal.net/folksonomy.html>. Accessed 19 Oct 2012.
- Wetzker, R., Zimmermann, C. and Bauckhage, C. (2008), Analyzing social bookmarking systems: A del.icio.us cookbook", *Proc. of European Conference on Artificial Intelligence*.

A Collaborative Filtering Recommendation System Combining Semantics and Bayesian Reasoning

Jialing Li

Li Li*

Xiao Wen

Jianwei Liao

Faculty of Computer and Information Science
Southwest University, Chongqing 400715, China

Email: {lj1333, lily, swnuwx}@swu.edu.cn

Abstract

Tag-based recommendation systems aim to improve the search experience of the end users. However, due to different backgrounds of the end users, descriptions of the same resources may be totally different in particle size and degree of specialization, which raises the question of how to tackle the growing discrepancy of public taxonomies (Folksonomy) in the social networks. In line with this, WordNet-based similarity is used to obtain semantic distance between tags and topic categories in order to reduce the divergence of tags. This in turn improves the search accuracy. The Bayesian reasoning is introduced to infer users' preferences through mining users' comments towards particular categories. Users' interaction behavior, which may facilitate preference estimation, is considered as well to enhance search efficiency. A series of experiments are conducted based on Flickr and Delicious datasets. The results show that the proposed recommendation algorithm can effectively improve search precision and provide a greater level of user satisfaction.

Keywords: Recommendation system, Collaborative filtering, Bayesian theory, Semantic similarity

1 Introduction

Overabundant growth of information on the Internet has raised many issues such as how to effectively manage vast amounts of heterogenous data, and how to help end users find resources according to their tastes with both accuracy and efficiency. Tags, as a kind of metadata, are used to address data heterogenous issues. Folksonomy formed by user tagging in social networking services such as Flickr¹, Delicious², etc., helps to classify and manage social data repositories in a uniform way. Tags also build a bridge between users and items, and can be used to improve search mechanisms (Karen et al. 2008). For taste matching issues, in addition to the familiar keyword searching technique, recommendation systems appear to complement potential interesting information with corresponding recipients (Resnick & Varian 1997).

The key point of recommendation is to capture users' preference as precisely as possible. However, in-

complete user preference modeling, preference learning and result presentation may hinder the system from achieving its full potential. In fact, there exists several discussions about what factors should be considered when modeling users' preference. Xu (Xu et al. 2011) model the user preference on various topics in a Topic Oriented Graph, and devise a topic-oriented tag-based recommendation system by preference propagation. Experiments show the approach outperforms several state-of-the-art collaborative filtering methods, yet it does not take users' contacts into account. Lemman's work (Lerman 2006) clearly shows that users' contact information can effectively improve search precision. In this paper, we build user's personalized file according to the content item liked by the user and topics liked by his/her friends, and then sort query results according to the file to improve user satisfaction.

Three research questions need to be answered in order to achieve a better recommendation result regarding the user's taste against a topic of an appointed contact, they are: 1) How to attain the topic-item relation? 2) How to learn the user's preference from his/her past rating behaviour? 3) How to cater the user by returning his/her most favourite items? This paper aims to address the above problems. The contributions of the paper include:

- Category identification problems tackled using semantics and category repositories;
- Bayesian probabilistic reasoning used to learn user's preference in social network tagging systems;
- Resulting items ranked accordingly to meet user's taste mostly.

The remainder of the paper is organized as follows. Section 2 is the problem definition. Section 3 details the proposed mining method, including Bayesian reasoning and item category identification with similarity calculation. Section 4 presents experimental results based on two social network datasets. Finally, Section 5 is the related works and Section 6 concludes the paper.

2 Terminology Definition

Firstly, suppose u_a is the centric user. Let $U = \{u_1, u_2, \dots, u_n\}$ denote the contact cluster of u_a . Then the items of u_i are presented as $I(u_i) = \{i_1, i_2, \dots, i_m\}$, the tags of i_s , $i_s \in I(u_i)$, are expressed as $T(u_i, i_s) = \{t_1, t_2, \dots, t_k\}$, topics of items are categorized into $C = \{c_1, c_2, \dots, c_r\}$. The organization relationship among them is shown in Fig. 1. Let us focus on the "interaction behaviour". Each item can be marked, commented or be liked by other users, which may include u_a . Thus we can mine u_a 's taste from its rating history, taking into account that the items u_a rated

*Corresponding author

Copyright ©2012, Australasian Computer Society, Inc. This paper appeared at the 10th Australasian Data Mining Conference (AusDM 2012), Sydney, Australia, December 2012. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 134, Yanchang Zhao, Jiuyong Li, Paul Kennedy, and Peter Christen, Ed. Reproduction for academic, not-for-profit purposes permitted provided this text is included.

¹<http://www.flickr.com>

²<http://delicious.com>

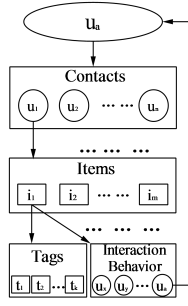


Figure 1: Tree-like structure in social tagging network

does not always come from his/her contacts' collections.

Our task is to track the centric user's indirect past ratings to learn his/her topic-specific preferences according to the appointed friend, which is represented as topic-contact matrix. The size of the matrix is $|C| \times |U|$, where $|C|$ is the number of topic categories. Let $Matrix(i, j) = f(c_i, u_j)$, so (c_i, u_j) are the items posted by u_j , and the concrete value represents the digitized extent that u_a shares the likes of u_b about topic category c_x , $c_x \in C$. The value is obtained by the method explained in Section 3.

Since the topic user cares about is different among websites, we initialize C in Flickr as {Animal, Art, City, Entertainment, Nature, News and Politics, People, Science and Technology, Sports, Travel and Places} on Flickr, and C in Delicious as {Animal, Art, City, Entertainment, Education, Fashion, Comedy, Lifestyle, Music, Nature, New, People, Science, Sport, Travel} on Delicious, these categories are obtained from the navigation classified on each website.

3 Development of Efficient Mining Method

Bayesian reasoning and item category identification method followed by similarity calculation are discussed in the following section. We also outline the mining algorithm.

3.1 Bayesian Reasoning

Since our aim is to analyze user's behaviours in different situations, we can figure out the distribution of user preferences for each friend on each topic category through calculating the conditional probability of collected statistics.

Let $likes(u_a, i)$ stand for items liked by the centric user u_a , and $posted(u_b)$ represent item collection posted by u_b . Let $topic(i)$ on behalf of the category that item i should belong to, and $topic(i) \in C$. The probability of item i posted by u_b , belonging to a certain category c_x and liked by u_a can be expressed as $p(likes(u_a, i) | topic(i) = c_x, i \in posted(u_b))$.

According to the Bayesian conditional theorem and Grsel et al.'work (Grsel & Sen 2009), the above probability can be rewritten as equation (1):

$$\frac{p(i \in posted(u_b) | topic(i) = c_x, likes(u_a, i)) \times p(likes(u_a, i) | topic(i) = c_x) \times p(topic(i) = c_x)}{p(i \in posted(u_b) | topic(i) = c_x) \times p(topic(i) = c_x)} \quad (1)$$

Factors in the above formula can be calculated by the following methods:

$p(i \in posted(u_b) | topic(i) = c_x, likes(u_a, i)) =$
(number of items proposed by u_b and belonging to

the category c_x and also liked by u_a)/(number of items belonging to the category c_x and also liked by u_a).

$p(likes(u_a, i) | topic(i) = c_x) =$ (number of items belonging to the category c_x and also liked by u_a)/(number of items belonging to the category c_x).

$p(i \in posted(u_b) | topic(i) = c_x) =$ (number of items proposed by u_b and belonging to the category c_x)/(number of items belonging to the category c_x).

3.2 Item Category Identification

In folksonomy, the key advantage of a tag is its straightforward way in describe the content of heterogeneous data like music data file, blog website, etc. Usually, through calculating the degree of its tags that also belongs to C , we can attain which topic the item should be indirectly be categorized into. Let the $topic(i)$ be the category of item i , t_k^i be the k -th element in its tag set, and $w(t_k^i)$ be the weight of the tag, the probability item i belonging to category c_x is computed as follows:

$$Pr(topic(i) = c_x) = \frac{\sum_{k=1}^{l_i} w(t_k^i) \times Level(c_x, t_k^i)}{\sum_{c_y \in C} \sum_{k=1}^{l_i} w(t_k^i) \times Level(c_y, t_k^i)} \quad (2)$$

When item i has only one tag, the weight of the tag equals one. Otherwise the $w(t_k^i)$ can be obtained by formula (3), in which the variable v denotes the number of tags of item i . And $Level(c_x, t_k^i)$ used to calculate the degree tag t_k^i belonging to c_x is calculated as formula (4).

$$w(t_k^i) = \begin{cases} \frac{1}{2^k} + \frac{1}{2^{v(v-1)}} & k < v \\ \frac{1}{2^k} & k = j \end{cases} \quad (3)$$

$$Level(c_x, t_k^i) = (1 - \theta) \times IsCategory(c_x, t_i) + \theta \times similarity(c_x, t_i) \quad (4)$$

The extent of t_i belonging to c_x is calculated in formula (4) composed of 2 parts. As preprocessing, we build category repositories to establish "belong" relation between tags and topic categories, through clustering tags which co-occurrence with the topic, namely, if $t_j \in c_x$, $t_k \in T(u_i, i_s)$, and also $t_j \in T(u_i, i_s)$, then $t_k \in c_x$. On startup, category name used as the 1-st one belongs to the corresponding category. The obtained category files reserve irregular form of tags on Flickr and Delicious, and we also combine it with similarity calculation based on WordNet built by domain experts to keep semantic information between tags and category names.

In formula (4), $IsCategory(c_x, t_i)$ returns the 0-1 value which identify whether the tag exists in the repositories of c_x . And $similarity(c_x, t_i)$ returns the similarity value of t_i and c_x , where $0 < \theta < 1$ is used to coordinate the contribution of two parts.

3.3 WordNet-based similarity calculation

Many works devote to find out the semantic meaning wrapped by WordNet content of least common subsumer(LCS) (Jiang, & Conrath 1998), path length (Leacock, & Chodorow 1998) and relation structure (Wu, & Palmer 1994). Resnik (Resnik 1992) deems

that the common information the concept pairs share is the criterion to measure similarity value between them. Lin (Lin 1998) believes that similarity value between two concepts, like, concept 1 and concept 2, can be expressed as information content ratios of the common information between them. It is computed as follows:

$$\text{similarity}(\text{concept}_1, \text{concept}_2) = \frac{2 \times \log p(\text{lso}(\text{concept}_1, \text{concept}_2))}{\log p(\text{concept}_1) + \log p(\text{concept}_2)} \quad (5)$$

In which, $\text{lso}(\text{concept}_1, \text{concept}_2)$ represents the lowest common ancestor node in the classification tree shared by concept_1 and concept_2 . $p(c)$ denotes the probability of encountering an instance of concept c . In this paper, we use Lin's method to compute the similarity between tags and categories because of its good performance (Budanitsky & Hirst 2006).

3.4 Main Processes

Below are the main processes of the proposed recommendation system in the form of pseudo code. Symbols u_a , u_b , c_x , $T(u_b, i)$ are defined in Section 2, in which, u_a denoted the centric user, u_b denoted friends of u_a , c_x was the topic category and $T(u_b, i)$ represented the tag collection of item i which was posted by u_b . $\text{Contact}(u_a)$ represents the friends of u_a . Symbol a is used to count the number of items posted by u_b , as well as being liked by u_a , which belongs to category c_x . Symbol b denotes the number of items which are liked by u_a , and also belonging to c_x . Symbol c represents the number of items belonging to c_x . Symbol d denotes the number of items which are posted by u_b and also belonging to c_x .

Require: the topic-contact matrix of u_a and the keywords to be queried

Ensure: recommendation items

```

1: for each  $u_b \in \text{Contact}(u_a)$  do
2:   for each category  $c_x$  do
3:     for each item  $i \in I(u_b)$  do
4:        $\text{WeightList} \leftarrow \text{WeightDistribution}(T(u_b, i));$ 
5:        $\text{CategoryList} \leftarrow$ 
6:          $\text{SimilarityCalculation}(T(u_b, i));$ 
7:        $\text{Topic}(i) \leftarrow$ 
8:          $\text{TopicIdentify}(\text{WeightList}, \text{CategoryList});$ 
9:       if  $i \in \text{Posted}(u_b)$  AND  $i \in \text{Interacted}(u_a)$  AND
10:         $\text{Topic}(i) = c_x$  then
11:          $a + 1 \leftarrow a;$ 
12:       end if
13:       if  $\text{Topic}(i) = c_x$  AND  $i \in \text{Interacted}(u_a)$  then
14:          $b + 1 \leftarrow b;$ 
15:       end if
16:       if  $i \in \text{Posted}(u_b)$  AND  $\text{Topic}(i) = c_x$  then
17:          $c + 1 \leftarrow c;$ 
18:       end if
19:       if  $\text{Topic}(i) = c_x$  then
20:          $d + 1 \leftarrow d;$ 
21:       end if
22:        $\text{Topic-ContactMatrix}(a, b, c, d);$ 
23:     end for
24:   end for
25: end for
26:  $\text{KeywordsItems} \leftarrow \text{keywords};$ 
27:  $\text{Sort}(\text{KeywordsItems}, \text{Topic-ContactMatrix})$ 

```

4 Experiments and Evaluations

4.1 Implementation

Using Visual Studio 2010 as IDE, the system is coded in C# language, and run on Microsoft Windows XP, .NET Framework 3.0 and above. The framework is

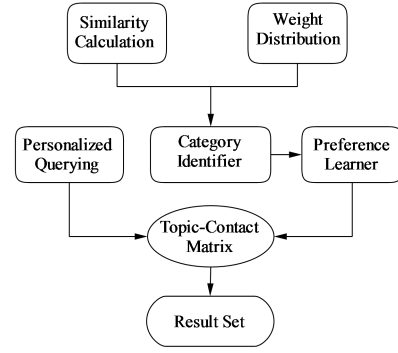


Figure 2: The framework of recommendation system

shown in Fig. 2. It includes modules such as Similarity Calculation, Weight Distribution, Category Identifier, Preference Learner and Personalized Querying.

Similarity Calculation: returning the semantic similarity between the lexical form of category and a certain tag.

Weight Distribution: returning weights of items tags sequentially computed by formula (3).

Category Identifier: returning the decreasing list of degrees of the item belonging to each category by formula (4).

Preference Learner: learning preference values from statistical data yield how much the centric user likes the items posted by one of his/her contacts. From statistical data in collections, we can see although u_a has a lot of contacts, the amount of items which u_a interacted is very little. Therefore, the topic-contact matrix is faced with a data sparsity problem. To improve efficiency of the preference matrix querying, we record the personalized information in triple formalized as $\langle \text{contact ID}, \text{topic category and preference value} \rangle$ stored in XML Schema after the first time preference has been learned to be reused later.

Personalized Querying: receiving a list of keywords as input, we obtain item collection with keyword searching. Then we recommend the result collection ranked by the topic-contact matrix in descending order according to the category sequence the keywords belong to.

4.2 Data Sets and Evaluations

The system was evaluated with two real-life datasets.

DataSet I: the dataset was crawled from Flickr during 2009.12-2010.1. We record the contact ID for the centric user and the photo information posted by each contact in two files. The photo information includes photo ID, contact ID (who posted the photo), sets of user IDs who had commented on it, and also its tag list. Before crawling, we check that the centric user is active enough to mine its preference.

DataSet II: It is the bookmark repository in Delicious, provided by the 2-nd International Workshop on Information Heterogeneity and Fusion in Recommendation system³ in 2011. The information was organized into 7 files. The statistics of the above two datasets are listed in Table 1.

In the experiment, for a given user, we select the top 20 items as recommendations ranked by the user's topic-contact matrix after firstly selected through keyword matching. The candidate items collected are assumed occur in the database. Suppose that I_1 are the top 20 items ranked by the matrix, or the top 20

³<http://ir.ii.uam.es/hetrec2011>

Table 1: Statistics of Experimental Datasets

Property	Flickr	Delicious
The number of users	10	1867
The number of contacts	3026	7668
The number of items	22969	69227
The number of tag assignments	124845	437593
Average tag number per user	45	48
Average tag number per resource	5	6

of the item collection returned by keywords searching as baseline. I_2 are the items in database. For each user, I_1 is different according to its topic-contact matrix and searching keywords each time. I_2 is the same on each website. It is meaningful to return the top 20 items to cater to user's taste preferences. Evaluation criterions are precision, recall and F-measure in this paper. Our criteria for the user "like" this photo is that the user's ID is contained in the photo's comment userID list.

$$precision = \frac{|number\ of\ items\ liked\ by\ u_a\ in\ I_1|}{|number\ of\ items\ in\ I_1|}$$

$$recall = \frac{|number\ of\ items\ liked\ by\ u_a\ in\ I_1|}{|number\ of\ items\ liked\ by\ u_a\ in\ I_2|}$$

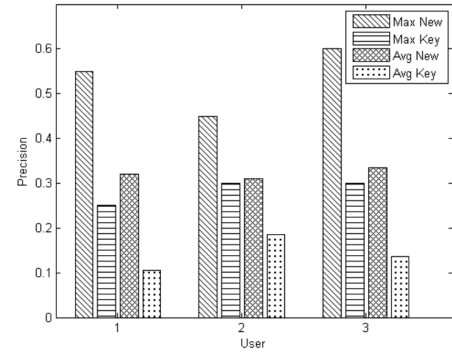
$$F - measure = \frac{2 \times precision \times recall}{precision + recall}$$

4.3 Results and Discussion

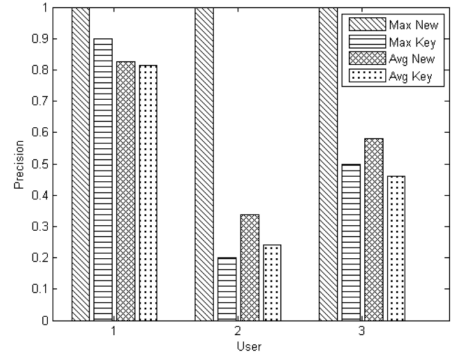
Firstly, we evaluate the proposed approach against the method of keyword searching only. There are 100 keyword tags for test randomly selected from tags of items in I_2 . Based on WordNet 2.0, we respectively learn 3 user preference in each dataset, we choose them for the reason that they are the most active users in each website. The precisions are compared in Fig. 3(a) and Fig. 3(b).

In Fig. 3(a) and Fig. 3(b), axis x presents users, axis y presents precision value corresponding to each user. Since we had tested 100 different tags for each user, we record their max precision and average precision value. Since user's purposes vary on different websites, the 100 testing tags are not the same between Flickr and Delicious, only the same in each. Legends sequentially denote max precision value of the proposed method (*New*), max precision value of keyword method (*Key*), average precision value of *New*, average precision value of *Key*. Method *New* utilize similarity calculation, where $\theta = 0.4$ in formula (4), the usage of θ will be discussed in Fig. 7.

The results show that the proposed method could influence ranking result items to make them better to the centric user's tastes. Let us focus on Flickr in Fig. 3(a). Our proposed method outperforms the baseline method in some cases. The results vary among users in Flickr and Delicious for difference of keywords for searching each time and each user's specialized topic-contact matrix. With respect to Delicious user in Fig. 3(b), the contact number of each Delicious user is smaller than the Flickr user in dataset we collected, and the biggest number is 90. Taking the 1-st Delicious user for example, it only contains one record in its preference matrix. We studied it as a limiting case for regarding proposed method. The number of contacts increases from user 1 to user 3 in Delicious in Fig. 3(b). The more the system learns, the higher the precision it achieves. However we can also see that the



(a) Flickr



(b) Delicious

Figure 3: Precision comparisons on two methods

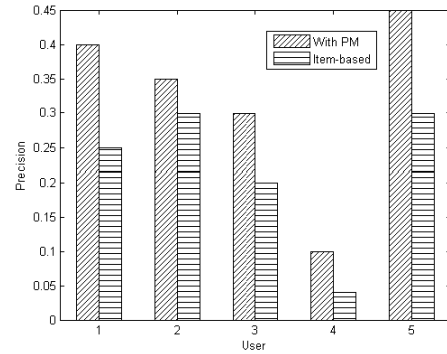
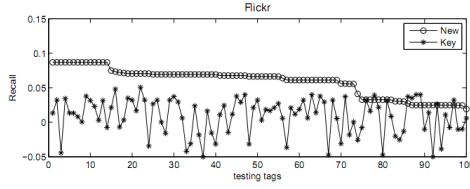


Figure 4: Precision comparisons on two methods

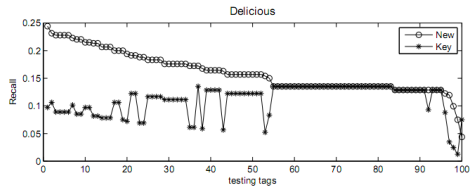
gap between the proposed method and the baseline method augments little. According with that average contact number of each user in Flickr can reach 300, it is evident that users in Flickr more frequently contact with their friends since their friends contribute more to their searching needs. The data also show that the proposed system improves precision by nearly 56% in Flickr and 21% in Delicious datasets.

Secondly, we evaluate the proposed approach against the traditional item-based method. The similarity between items is calculated according to the similarity among each item's tag vector. Using Latent Dirichlet Allocation(LDA) we easily dig the weights of which each item belongs to the latent topics in a certain number, and on the basis of the common among weight vector, we calculate the cosine similarity between weight vector.

Fig.4 is the comparison of the item-based method and the proposed method. It shows that the later one is better in our discussion. The more we learn from user's rating behaviour the more precise we can recommend items for him/her. As shown in Fig.4,

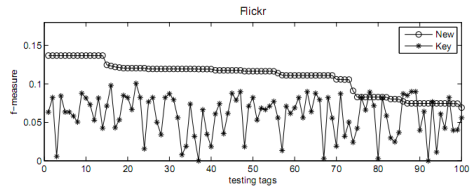


(a) Recall on Flickr

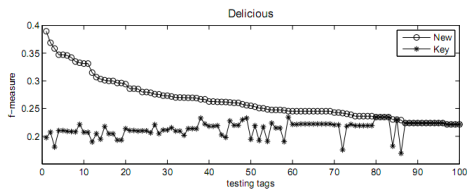


(b) Recall on Delicious

Figure 5: Recall comparisons on two methods



(a) F-measure on Flickr



(b) F-measure on Delicious

Figure 6: F-measure comparisons on two methods

the proposed method could gain more better result for the more active ones like user 1 and user 5 are. So the proposed method has certain learning ability.

Thirdly, two other criteria, the Recalls and F-measures are measured and the results are shown in Fig. 5 and Fig. 6, respectively. In Fig. 5 and Fig. 6, the axis x presents testing tags, the axis y presents recall measurement according to each tag in Fig. 5 and the corresponding f-measure. The legends “New” and “Key” respectively denotes the method with PM and the item-based method. The two methods are compared with the obtained data from Flickr (Fig. 5(a) and Fig. 6(a)) and Delicious (Fig. 5(b) and Fig. 6(b)) datasets, respectively. The results are sorted by the value of “New” in descending order. It shows that the proposed recommendation method outperforms those in which only keyword searching is used.

Finally, we study the impact of WordNet similarity calculation on recommendation precision as shown in Fig. 7. In Fig. 7 the axis x presents testing tags, the axis y presents precision value corresponding to each tag, legends represent proposed method with WordNet-based similarity calculation (*Sim*) and proposed method without similarity calculation (*NoSim*). In *Sim*, let $\theta = 1$ in formula (4) and in *NoSim*, let $\theta = 0$ in the same formula.

We find that contribution of similarity to the precision rate is not large, since when testing, the tags we choose is mostly already “known” in the tag repository,

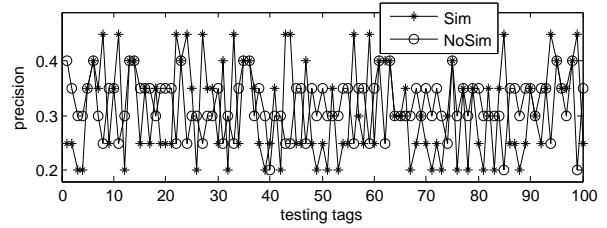


Figure 7: Precision comparisons on two methods

ry, this ensures that result can be found by keyword searching. Take searching keyword “cat” for example, no similarity-based method can identify the word since it already exists in category repositories, while using its synonym “kitty” as a replacement in the query, the proposed method without similarity failed to identify the word while the proposed similarity-based method can identify the word not in the tag repository. According to the results in Fig. 7, the proposed method with similarity calculation is preferable to the non-similarity method.

5 Related Work

Topic-based, tag-based and user-based methods are the dominant techniques applied in recommendation system (Shepitsen et al. 2010)(Rashid et al. 2002). Jin et al. (Jin et al. 2011) use Latent Dirichlet Allocation(LDA) to model topics probability distribution over tags and resources of multiple topics. It is uncommon that each tag has the same topic vector. Thus we consider different topics for different website users. Ziegler et al. (Ziegler et al. 2005) present topic diversification method and introduce a intra-list similarity metric to assess the topical diversity of recommendation lists. Although the proposed method can improve user satisfaction with recommendation lists, it is also detrimental to the average accuracy. Abel et al. (Abel et al. 2011) find that temporal profile build by hashtag-based, entity-based, and topic-based benefit from semantic enrichment improve recommendation quality. Results demonstrate that topic-based methods perform better than the hashtag-based strategy and requires less run-time and memory.

Cantador et al. (Cantador et al. 2011) argue that in some cases, tags are used to depict subjective qualities of a item or be related to organisational aspects. They map the concepts of filtered tags and classified by their mechanism to semantic entities like WordNet and Wikipedia. The obtained concepts are then transformed into semantic classes that can be uniquely assigned to context-based categories. They should dig into the semantic relations between the obtained tag clusters. Krestel et al. (Krestel & Fankhauser 2012) explore an approach to personalized tag recommendation that combining a probabilistic model of tags from the resource with tags from the user. They use LDA to estimate the topic-tag distribution and the resource topic distribution from an unlabeled corpus of documents using Dirichlet priors for the distributions and a fixed number of topics. Shepitsen et al. (Shepitsen et al. 2010) present a personalization algorithm for recommendation in folksonomies which rely on hierarchical tag clusters. Experiments show that clusters of tags can be effectively used as a means to ascertain the user’s interest to determine the topic of a resource. Durao et al. (Durao & Dolog 2010) present a tag-based recommendation system which suggests similar resources based on cosine similarity

calculus with additional factors such as tag popularity, tag representativeness and affinity between user and tag. Semantic similarity can be used to overcome ambiguity problems and further improve recommendation accuracy.

Nocera et al. (Nocera & Ursino 2011) cluster users based on the tags they share, and provide a user with recommendations of similar users and resources. While Nakatsuji et al. (Nakatsuji et al. 2009) allow user profiles to be constructed as a hierarchy of classes, user's interest weight is assigned to each class and instance, then generates user with the highest similarity into group. Yin et al. (Yin et al. 2012) deem that it is worthwhile to emphasize the significance of trust information providing reliable personal friend relationships, and provide a simple framework for the design of the prediction model making use of both listening and trust information. Firan et al. (Firan et al. 2007) focus on how tags characterized the user and enable personalized recommendations. Through analysing tag usage in contrast to conventional user profiles, they specify recommendation algorithms based on tag user profiles.

6 Conclusions

In order to improve users' searching experience by returning items most catering to his/her tastes on social networking websites, we propose a recommendation system by utilizing the Bayesian rule and WordNet-based similarity calculation to address problems such as user profile modeling, item category identification, preference learning and result presentation. We have achieved an improved average recommending precision by nearly 56% and 21% with the obtained data from Flickr and Delicious, respectively. It also helps to diversify users' needs and may foster new interests around the original search tags with the proposed semantic extension. The methods proposed in this paper can be applied to other social tagging networks such as Blog systems. Finally, our method shows potential in providing a personalized spin on the social network experience. In the future, we will focus on the improvement of preference calculations.

Acknowledgements

This work was supported by National Natural Science Foundations of China(61170192), and Natural Science Foundations of CQ.

References

- Karen,H.L., Tso-Sutter, Leandro,B.M.,(2008), Tag-aware Recommender Systems by Fusion of Collaborative Filtering Algorithms. *SAC'08: Proceedings 23rd Annual ACM Symposium on Applied Computing*, Fortaleza, Brazil.
- Resnick,P., Varian,H.R. (1997), Recommender Systems, *Communications of the ACM*, Vol. 40, pp. 56-58.
- Xu,G.D., Gu,Y.H., Zhang,Y.C. (2011), TOAST: A Topic-oriented Tag-based Recommender System, *WISE'11: Proceedings of the 12th international conference on Web information system engineering*, Berlin, Heidelberg.
- Lerman,K. (2006), Social Networks and Social Information Filtering on Digg, *ICWSM2007 Boulder*, Colorado, USA.
- Gursel,A., Sen,S. (2009), Improving Search in Social Networks by Agent based Mining, *IJCAI'09: Proceedings of the 21st international joint conference on Artificial intelligence*, San Francisco, USA.
- Jiang,J.J., Conrath,D.W. (1998), Semantic Similarity based on Corpus Statistics and Lexical Taxonomy, In *Proceedings of the International Conference on Research in Computational Linguistics*, TaiWan, China.
- Leacock,C., Chodorow,M. (1998), Combining Local Context and WordNet Similarity for Word Sense Identification. In *WordNet: An Electronic Lexical Database*,Christiane Fellbaum, MIT Press, Vol. 49,pp. 265-283.
- Wu,Z.B., Palmer,M. (1994), Verb Semantics and Lexical Selection, *ACL-94: Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, Las Cruces, NM.
- Resnik,P. (1992), WordNet and Distributional Analysis: A Class-based Approach to Lexical Discover. in *Proceedings of the AAAI Symposium on Probabilistic Approaches to Natural Language*,San Joe, CA.
- Lin,D.K. (1998), An Information-theoretic Definition of Similarity. In *Proceedings of the 15th International Conf. on Machine Learning*, San Francisco, CA.
- Budanitsky,A., Hirst,G. (2006), Evaluating WordNet-based Measures of Lexical Semantic Relatedness, *Computational Linguistics*, Vol. 32, pp. 13-48.
- Shepitsen,A., Gemmell,J., Mobasher,B. (2010), Personalized Recommendation in Social Tagging Systems Using Hierarchical Clustering. In *RecSys'10: Proceedings of the fourth ACM conference on Recommender systems*, New York, USA.
- Rashid,A.M., Albert,I., Cosley,D. (2002), Getting to Know You:Learning New User Preferences in Recommender Systems, In *Proceedings of the 7th International conference on Intelligent user interfaces*, San Francisco, USA.
- Jin,Y.A., Li,R., Wen,K. (2011), Topic-based Ranking in Folksonomy via Probabilistic Model, *Artificial Intelligence Review*, Vol. 36, pp. 139-151.
- Ziegler,C.N., McNee,S.M., Konstan,J.A. (2005), Improving Recommendation Lists through Topic Diversification, In *Proceedings of the 14-th international conference on World Wide Web*, New York, USA.
- Abel,F., Gao,Q., Houben,G.J., et al. (2011), Analyzing User Modeling on Twitter for Personalized News Recommendations, *Lecture Notes in Computer Science*, Vol. 6787,pp. 1-12.
- Cantador,I., Konstant,I., Jose,J.M. (2011), Categorising Social Tags to Improve Folksonomy-based Recommendations, *Web Semantics:Science,Services and Agents on the World Wide Web*, Vol. 9,pp. 1-15.
- Krestel,R., Fankhauser,P. (2012), Personalized Topic-based Tag Recommendation, *Neurocomputing*, Vol. 76, pp. 61-70.
- Durao,F., Dolog,P. (2010), A Personalized Tag-Based Recommendation in Social Web Systems. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, New York, USA.

- Nocera,A., Ursino,D. (2011), An Approach to Providing a User of a social folksonomy with Recommendations of Similar Users and Potentially Interesting Resources, *Knowledge-Based Systems*, Vol. 24, pp. 1277-1296.
- Nakatsuji,M., Yoshida,M., Ishida,T. (2009), Detecting Innovative Topics based on User-interest Ontology, *Web Semantics:Science, Services and Agents on the World Wide Web*, Vol. 7, pp. 107-120.
- Yin,C.X., Peng,Q.K., Chu,T. (2012), Personal Artist Recommendation via a Listening and Trust Preference Network, *Physica A:Statistical Mechanics and its Applications*, Vol. 391, pp. 1991-1999.
- Firan,C.S., Nejdl,W., Paiu,R. (2007), The Benefit of Using Tag-Based Profiles, *In Proceeding of the 2007 Latin American Web Conference IEEE Computer Society*, Washington, USA.

Evaluating the Performance of Several Data Mining Methods for Predicting Irrigation Water Requirement

Mahmood A. Khan¹, Md Zahidul Islam^{2,3}, Mohsin Hafeez^{1,4}

¹ School of Environmental Sciences, Charles Sturt University, Wagga Wagga 2678, NSW, Australia

² School of Computing and Mathematics, Charles Sturt University, Bathurst 2795, NSW, Australia

³ Centre for Research in Complex Systems (CRiCS), Charles Sturt University, Bathurst 2795, NSW, Australia

⁴ GHD Pty Ltd, Brisbane 4000, QLD, Australia

makhan@csu.edu.au, zislam@csu.edu.au, mhafeez@csu.edu.au

Abstract

Recent drought and population growth are planting unprecedented demand for the use of available limited water resources. Irrigated agriculture is one of the major consumers of fresh water. Huge amount of water in irrigated agriculture is wasted due to poor water management practices. To improve water management in irrigated areas, models for estimation of future water requirements are needed. Developing a model for Irrigation water demand forecasting based on historical data is critical to effectively improve the water management practices and maximise water productivity. Data mining can be used effectively to build such models. Data mining is capable of extracting and interpreting the hidden patterns from a large amount of hydrological data. In recent years, use of data mining has become more common in hydrological modelling.

In this paper, we compare the effectiveness of six different data mining methods namely decision tree (DT), artificial neural networks (ANNs), systematically developed forest (SysFor) for multiple trees, support vector machine (SVM), logistic regression and the traditional Evapotranspiration (ET_c) methods and evaluate the performance of these models to predict irrigation water demand using pre-processed dataset. The pre-processed dataset we use in this study and SysFor were never used before to compare with any other classification techniques. Our experimental result indicates SysFor produces the best prediction with 97.5% accuracy followed by decision tree with 96% and ANN with 95% respectively by closely matching the predictions for water demand with actual water usage. Therefore, we recommend using SysFor and DT models for irrigation water demand forecasting.

Keywords: Irrigation water demand forecasting, Data mining, Decision tree, ANN, Multiple trees and Water management.

1 Introduction

Water scarcity is rapidly becoming a major issue for many developed and developing countries of the world,

which is a serious threat and leads to emergence of food crisis (IWMI 2009). As the scarcity of the water increases, the demand for managing available water resources becomes crucial. In particular, a recent drought in Australia has made prominent the need to manage agriculture water more wisely. It is reported that, more than 70% of available water in Australia and 70% to 80% of water Worldwide is currently being used by irrigated agriculture (Khan et al. 2009, Khan et al. 2011, IWMI 2009). Due to recent drought, climate change, population growth and increasing demand for domestic and industrial water requirement, preserving sufficient amount of freshwater for agricultural production will become increasingly difficult. Since all the existing water resources are fully utilised and drawing of more water is impracticable, therefore the best alternative is to increase the water productivity (Khan et al. 2011). Studies report that, the water delivered for irrigation is not always efficiently used for crop production, on an average 25% of water is wasted due to inefficient water management practices (FAO 1994, Smith 2000).

In order to improve water management and maximise water productivity application of various hydrological and data driven models using data mining methods have become very essential. In the current situation, models to predict future water requirements based on data mining techniques can be useful. Ullah et al. (2011) suggests that, to developing a model for water demand forecast, it is essential to understand the behaviour of the irrigation system in the past, the current land use trends and the behaviour of future hydrological attributes such as (rainfall, Evapotranspiration, seepage, etc.). Having an accurate and reliable Irrigation water demand forecasting model based on hydrological, meteorological and remote sensing data can provide important information to agriculture water users and managers (Pulido-Calvo et al. 2009, Zhou et al. 2002, Alvisi et al. 2007).

Recently, data mining techniques are increasingly being applied in the field of hydrology for developing models to predict various hydrological attributes such as rainfall, pan evapotranspiration, flood forecasting, weather forecasting etc (Pulido-Calvo *et al.* 2003). However, these techniques are not used for irrigation water demand forecasting. Knowledge discovery from any data set can be obtained through data mining. It discovers new and practically meaningful information from large datasets. Unlike any typical statistical methods, data mining techniques explores interesting and useful information without having any pre set hypotheses. These techniques are more powerful, flexible and capable

of performing investigative analysis (Olaiya et al. 2012). Zurada et al. (2005) says, data mining uses a number of analytical tools such as decision trees, neural networks, fuzzy logic, rough sets, and genetic algorithms to perform classification, prediction, clustering, summarisation, and optimisation. The most common tasks among these are classification and prediction which we carry out in this study.

The aim of this study is to explore and compare the effectiveness of accuracies of different data mining models on predicted water usage. We build models based on five data mining techniques namely decision trees, artificial neural networks, systematically developed forest (SysFor), support vector machine, logistic regression, and traditional ET_c based method. To best of our knowledge SysFor is compared with other classification techniques for the first time.

To develop an effective irrigation water demand forecasting model using data mining techniques adequate historical data for the attributes having high influence on water usage are required. We use the dataset which was collected from three different sources and pre-processed by Khan et al. (2011). The data pre-processing was carried out using a novel approach called Reference Evapotranspiration Based Estimate, which is based on Reference Evapotranspiration (ET_c), a comprehensive explanation can be found in Khan et al. (2011).

Once the models are built, we use the models to predict the water requirements for the unseen data. Our experimental results indicate a minor difference in the prediction accuracies of different data mining techniques. However, among the five different techniques/models the prediction performance of multiple decision tree technique Sysfor is found to be the best followed by Decision Tree and ANN.

This paper is organised as follows, section 2 describes the methods/models used in this study, followed by the description of study area and dataset in section 3. Experimental results are explained in the Section 4, Section 5 concludes the paper with some suggestions for future work.

2 Description of methods

All the methods/techniques used to predict water demand forecast in this study are well known and well established. Therefore, we explain only the basic functionalities of each method, without explaining the mathematical descriptions of the underlying algorithms. For more information relating to any specific algorithm on decision tree, artificial neural networks, support vector machine, systematically developed forest (SysFor) and logistic regression refer to (Quinlan 1993, Islam 2010, Khan et al. 2011; Cancelliere et al. 2002, Yang et al. 2006, Han & Kamber 2001; Vapnik 1995; Islam & Giggins 2011; Christensen, R. 1997). We explain the methods one by one as follows.

2.1. Decision Tree (DT)

Decision trees are a powerful tool for data classification. Decision tree learns from the training dataset and apply the learned knowledge on the testing dataset to find the hidden relationships between the classifying (class) and

classifier (non class) attributes. A class attribute is an attribute of 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). By using the rules generated by a decision tree the relationship between the attributes of a dataset can be extracted. Each rule represents a unique path from the root node to each leaf of the tree.

Decision trees are made of nodes and leaves, as shown in Figure 1 where each node in the tree represents an attribute and each leaf represents the value for the records belonging to the leaf (Khan et al. 2011, Han & Kamber 2001). The concept of information gain is used in deciding the best suitable attribute for a node. The functionality of the decision tree is based on C4.5 algorithm (Quinlan 1993).

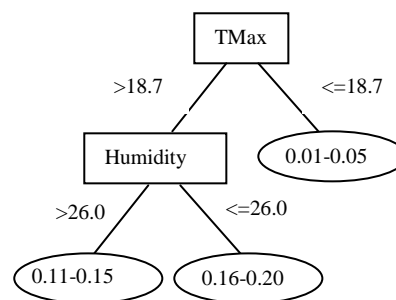


Figure1: An Example of a Decision tree generated from our dataset

2.2. Artificial Neural Networks (ANN)

Artificial Neural Network (ANN) is a data processing and classification model that is inspired by the biological neural network. ANN learns the non-linear relationships, trends and patterns from training dataset and uses the knowledge for predicting the class values of unseen datasets (Cancelliere et al. 2002, Yang et al. 2006).

Interconnection strengths known as weights are used to store the gained knowledge. Weights of the neurons in ANN are computed during the training process. Based on the nature of the datasets an appropriate network can be selected, where a user/data miner can choose number of layers and number of nodes in each layer of the network. In hydrological modelling most ANNs are trained with single hidden layer (Dawson & Wilby 2001, de Vos & Rientjes 2005) as reported by Wu et al. (2010). The ANN model is based on error minimisation principle. Training of the model can be carried out in two ways; supervised and unsupervised learning (Craven & Shavlik 1998, Han & Kamber 2001).

One of the most popular and commonly used ANN architectures is multilayer feed-forward neural network as shown in Figure 2, which is also called as multilayer perceptron (Muttill & Chau 2006). In a multilayer perceptron network there is an input layer, an output layer and one or more hidden layers. These layers extract patterns from a dataset and use the learned patterns to predict class values of new records. The nodes in the input layer pass the processed information to the computational nodes in a forward direction (Wang et al.

2009). The hidden layer is also responsible for resolving the nonlinearity between the input and output attributes of the data set (Ambrozic & Turk 2003, Cancelliere *et al.* 2002, Safer 2003).

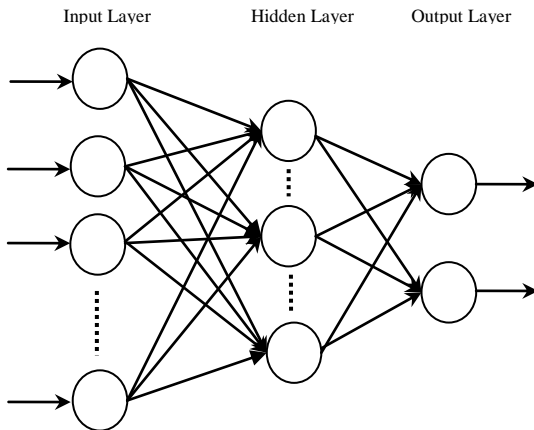


Figure 2: Architecture of three - tier feed forward neural network.

2.3. Systematically Developed Forest of Multiple Trees (SysFor)

SysFor is a multiple tree building technique based on the concept of gain ratio. This technique is developed by Islam & Giggins (2011). The purpose of building multiple trees is to gain better knowledge through the extraction of multiple patterns. We explain this technique in a step by step fashion.

In the first step, a set of good attributes and their split points are identified based on user defined goodness (gain ratio) and separation values. Islam & Giggins (2011) says, a numerical attribute can be chosen more than once within the set of good attributes, if it has higher gain ratios with different split points that are not close to each other. After the set of good attributes are selected and if the size of the good attributes is less than a user defined number of tree, then in the next step (step 2) SysFor builds the tree using each good attribute as the root attribute of the tree, and build as many trees as number of good attributes. Else it builds user defined number of trees from the set of good attributes as the root attribute.

If the number of trees build in this step are still less than the user defined number of trees, then SysFor in the next step (step 3) build more trees until user defined number is met by using alternative good attributes at the next level of the tree i.e. at level 1 of the tree generated in the previous step (step2). In this step (step 3) the algorithm first uses the root attribute of the first tree built in step 2 in order to split dataset into horizontal partition. The algorithm, then selects a new set of good attributes, their respective split points and a set of gain ratios for each horizontal partition. Based on these set of good attributes the algorithm builds a tree from each partition and the trees are joined by connecting their roots (at level 1) to the root (at level 0) of first tree build in step 2. This process of building more trees continues until user defined number of trees are generated/build. Example trees generated in SysFor are shown in Figure 3a, 3b.

After Systematic forest of multiple trees is generated as to predict the class values of unseen records we follow

voting system proposed by Islam and Giggins (2011) called SysFor Voting-2. In this voting system, we find all the leaves from all the trees the record falls into. Then the leaf with highest accuracy is determined (based on maximum number records with same class values to total number of records) and finally the majority class value of the leaf is chosen as the predicted class value of the record.

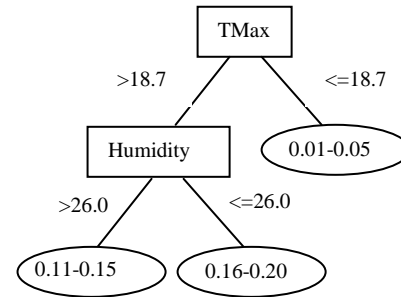


Figure 3a: Tree generated in SysFor based on first good attribute

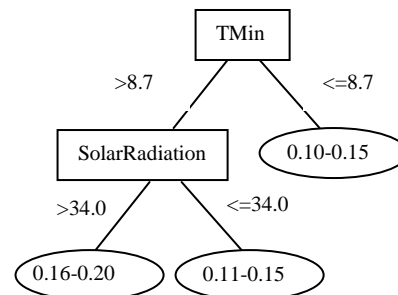


Figure 3b: Tree generated in SysFor based on second good attribute

2.4. Support Vector Machine (SVM)

Support vector machine is a state of the art neural network methodology based on statistical learning (Vapnik 1995, Wang et al. 2009). An SVM is an algorithm for maximizing a particular mathematical function with respect to a given dataset. The basic concepts behind the SVM algorithm are i) the separating hyperplane, ii) the maximum-margin hyperplane, iii) the soft margin and iv) the kernel function. A support vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification. In general, a good separation is achieved by the hyperplane that has the largest distance to the nearest training data point of any class as shown in Figure 4 which exhibits the basic concept of support vector machine. From Figure 4 it can be seen that the optimal hyperplane separates the positive and negative points from the dataset with a maximum margin, indicating the maximum distance to hyperplane from closest positive and negative data points.

2.5. Evapotranspiration (ET_c) based Prediction

ET_c can be broadly defined as crop water usage. 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 56), and

this is commonly used globally to estimate water demand. The crop coefficient method was developed for the agriculture users to calculate ET_c which helps them in making irrigation management decisions.

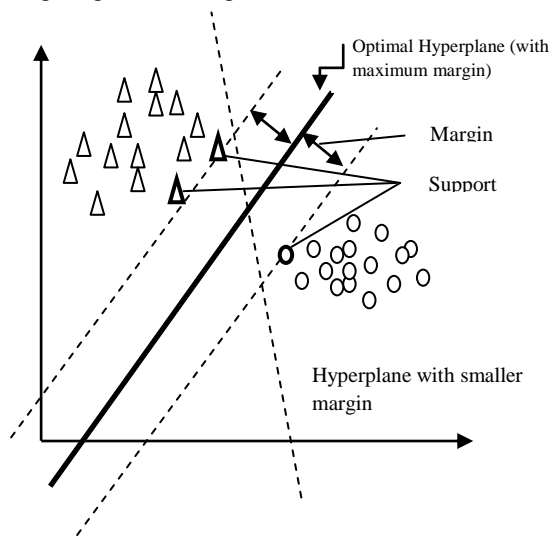


Figure 4: Basic concept of support vector machine

2.6. Logistic Regression

The main goal of logistic regression model is to predict the label t of a new given data point x based on the learning from the training data set. Logistic regression can be of two types 1) Simple Logistic Regression and ii) Multiple Logistic regression. Simple logistic regression is used to predict the class value, given it is categorical and has only two possible outcomes such as (male/female). Whereas, the multiple logistic regression can be used to predict the class value consisting of three or more possible outcomes.

Logistic regression is a capable probabilistic binary classifier (Christensen 1997). A logistic regression model helps us assess probability from which the outcomes will be chosen.

It is evident from the literature that the logistic regression is used extensively in numerous disciplines such as, in the field of medical and social sciences, marketing applications etc (Pearce & Ferrier, 2000). Zurada 2005, states that logistic regression models are designed to predict one class value at a time and they are assumed as simplest feed forward neural networks containing only two layers input and output.

3 Study area and Dataset

In this study, Coleambally Irrigation Area (CIA) is selected as our study area. CIA is one of the most modernized irrigation areas in the Murray and Murrumbidgee river basins of Australia. CIA is situated approximately 650km south-west of Sydney in the Riverina District of New South Wales which falls under lower part of Murrumbidgee River catchment as shown in Figure 5. CIA contains approximately 79,000ha of intensive irrigation area and 325,000ha of the Outfall District area, supplying water to 495 irrigation farms (CICL, 2011). Because of the recent drought in the last decade, there is a significant decline in the average water allocation to the farmers of CIA. Due to declining water

allocation and changing weather patterns, CIA requires new management measures for water use efficiency and increase water productivity.

The data for the years 2007/08 and 2009/10 is used to train the models and the data for summer season of 2008/09 is used to test the models. We use the same dataset which was collected from three different sources namely Water delivery statements, Meteorological data, and Remote sensing data and pre-processed by Khan et al. (2011) consisting of 1500 records. Khan et al. (2011), claims the dataset was pre-processed using a novel method which is based on Reference Evapotranspiration (ET_o) and is the combination of knowledge in irrigation engineering and data mining. The main goal to pre-process that dataset was to estimate daily crop water usage more accurately based on the data collected from water delivery statements.

The dataset consist of historical data on weather parameters such as Maximum and Minimum temperature, wind speed, humidity, rainfall and solar radiation in combination with soil type, crop type and crop water usage. Attributes crop type and soil type are categorical and the rest are numerical. In our experiments, we consider crop water usage as the class attribute and the rest as non-class attributes, also crop water usage is considered as a categorical attribute.

4 Experimental Results

The main purpose of this experiment is to compare the prediction performances of different data mining models on water demand forecasting.

We first built a decision tree from our training dataset to extract the relationship between the non-class and class attributes. We implement C4.5 algorithm to generate a decision tree. C4.5 takes a divide and conquers approach to build a decision tree from a training dataset using the principle of information gain (Quinlan 1993). Here we divide our dataset into two parts training and testing, the tree is built on training dataset and applied on testing dataset to check the prediction accuracy of unseen records.

In this study, an ANN is built using the three tier feed-forward architecture with back propagation. In order to build an ANN, we divide the datasets into three parts; 70%, 20% and 10% for training, validating and testing, respectively. Training of the network is performed using two different network topologies, firstly by using 1 hidden layer having 8 nodes, and secondly by using 1 hidden layer having 6 nodes. Both the networks are trained for 30000, 50000 and 70000 learning iterations. The network produced by 1 hidden layer with 8 nodes for 30000 learning iterations produces better results. The ANN is built using EasyNN plus V14.0 software (available from <http://www.easynn.com/>).

We also build SysFor on our dataset, by considering user defined number of trees to be 5 and follow SysFor voting 2 for predicting the unseen records.

Finally we train and test SVM and Logistic regression using WEKA 3.6.2 which is available at <http://www.cs.waikato.ac.nz/~ml/weka/> and very popularly used tool for performing different data mining tasks.

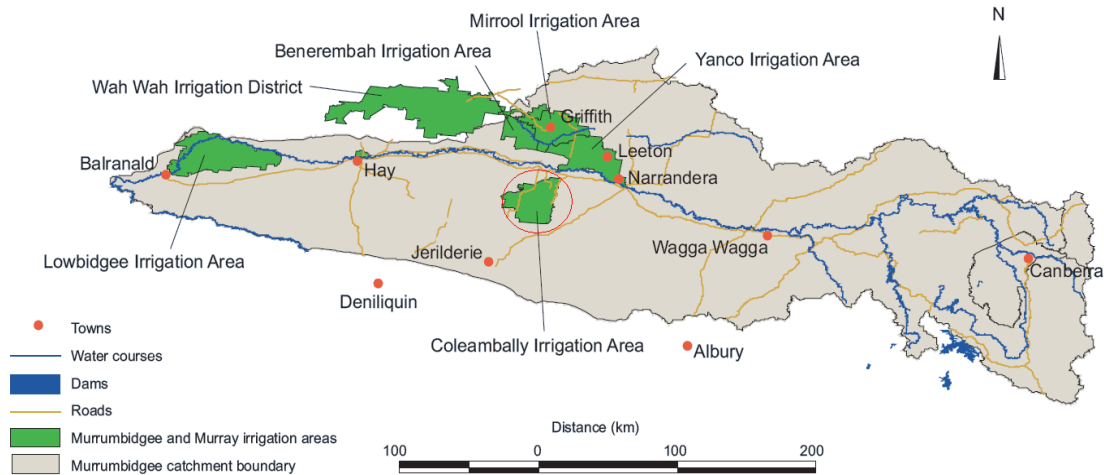


Figure 5: Location of Coleambally Irrigation Area and Other Major Irrigation Areas in Murrumbidgee Catchment

The performance evaluation of the models is carried out by comparing the prediction accuracies. The prediction accuracy check is performed using a 3 fold cross validation method. This is a method of testing the accuracy by dividing the dataset in three equal parts also called as folds, where two parts of the dataset are used for training and the third part is used for testing. This process is continued 3 times so that each part of the dataset is used once for testing. Table 1 displays the prediction accuracies of all the models used in our experiment.

Table 1 indicates that the performance of multiple decision tree technique Sysfor is better among all the other techniques, followed by decision tree and SVM. SysFor records 78% prediction accuracy while DT and SVM exhibit an accuracy of 74% and 64% respectively. The accuracy of ANN and logistic regression were recorded low. We also compare the accuracies of the experimented models with the accuracy of traditional approach which is based on actual crop evapotranspiration (ET_c).

Folds	Model				
	DT	ANN	SysFor	SVM	Logistic Regression
1	72.6	59.3	75.5	63.2	57.1
2	74.5	60.7	83	62.1	53.7
3	73.8	62	77.9	67.1	56.7
Average (%)	74	61	78	64	56

Table 1: Prediction accuracies of different models based on 3 folds cross validation

Apart from accuracy test we also compare the closeness of actual water consumed by the crop to the water predicted by the above mentioned models for summer season of the year 2008/09. Table 2 shows a comparison between the actual water usage, water usage predicted by the decision tree, ANN, SysFor, SVM, Logistic regression and traditional ET_c based approach for all the 22 nodes of CIA.

All the models are applied on every farm of CIA to obtain the water demand for a whole cropping season. The water demand for each node is calculated by adding the water demand predicted for the farms belonging to the node. The accuracy of closeness for actual and predicted water is calculated as follows

$$\text{Accuracy} = 1 - \left(\frac{|\text{Actual-Predicted Water Usage}|}{\text{Actual Water Usage}} \right) \times 100\%$$

From Table 2 it is evident that the water demand predicted by SysFor is more closely matching the actual water consumed. The accuracy of closeness is found to be 97.5% which suggest a high closeness of prediction made by the model. The accuracy of SysFor is followed by decision tree and ANN whose closeness is found to be 96% and 95% which is also considered to be very high. However, in few nodes such as Yammal and Boona 2 the prediction of SysFor was worse than decision tree and ANN. In majority of the nodes the performance of SVM, Logistic regression and ET_c was behind the performance of Sysfor, decision tree and ANN.

Moreover, 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 all the models. This is because only a few farms of the nodes were irrigating during the season. 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 exclude results of these nodes while calculating the accuracy of the models. In Table 2 the rows representing the above said nodes are shaded to highlight the exclusion of these nodes.

Figure 6 and Figure 7 displays the basic comparison between actual and predicted water usage. Figure 6 show the positive (predicted more) and negative (predicted less) predictions to actual water usage for all 22 nodes of CIA from all six models. It is evident from Figure 6 that the bars representing SysFor and DT are shorter for all nodes compared to the longer bars representing other models.

Node	Predicted Water Usage						
	Actual Water Usage (ML)	Decision Tree (ML)	ANN (ML)	SysFor (ML)	SVM (ML)	Regression (ML)	ET _c (ML)
Coly 1_2	407	344	316	379	417	428	284
Coly 3	1292	1203	1210	1155	1278	1417	777
Coly 4	800	746	1262	759	841	931	570
Coly 5	879	945	1383	1001	1110	1228	666
Coly 6	4359	4158	3807	4464	4891	5266	3235
Coly 7	82	220.5	245	231	256	283	157
Coly 8	785	802	830	850	1084	1139	875
Coly 9	4501	4297	4394	4317	4801	5211	3232
Coly 10	0	0	0	0	0	0	0
Coly 11	2262	2877.5	3104	2581	2996	3139	2264
Tubbo	696	630	814	645.7	716	792	444
Boona 1	1201	1069	1692	1189	1323	1424	791
Boona 2	418	429	531	550	720	797	259
Boona 3	2438	2101	2268	2341	2585	2713	1652
Yamma Main	4299	3732	4542	4375	4921	4966	3098
Yamma 1	3333	3364	3100	3940	5558	5558	3085
Yamma 2_3_4	2926	3045	3207	3180	4479	4370	2772
Bundure Main	87	493	650	646	726	745	419
Bundure 3	763	768	636	798	897	901	653
Bundure 4	1597	1384	1421	1387	1560	1532	897
Bundure 5_6	961	798	660	836	935	941	677
Bundure 7_8	133	378	504	396	440	486	268.5
Coleambally Irrigation Area	33917	32692.5	35177	34747.7	41112	42753	26231

Table 2: Comparison of water usage predicted by different models to actual water usage for all nodes of CIA

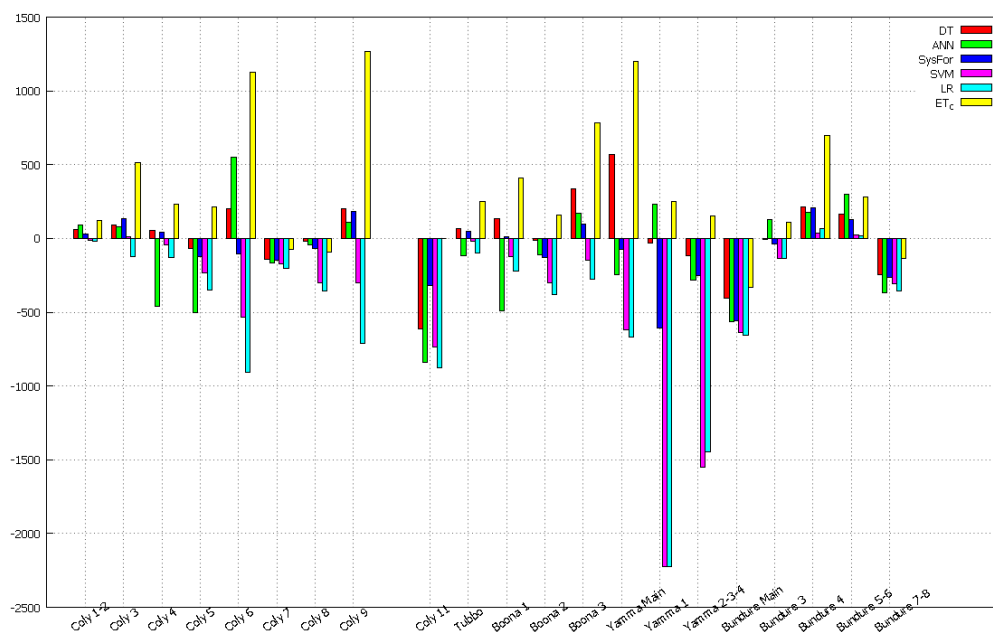


Figure 6: Positive and Negative difference between actual and predicted water usage made by different models

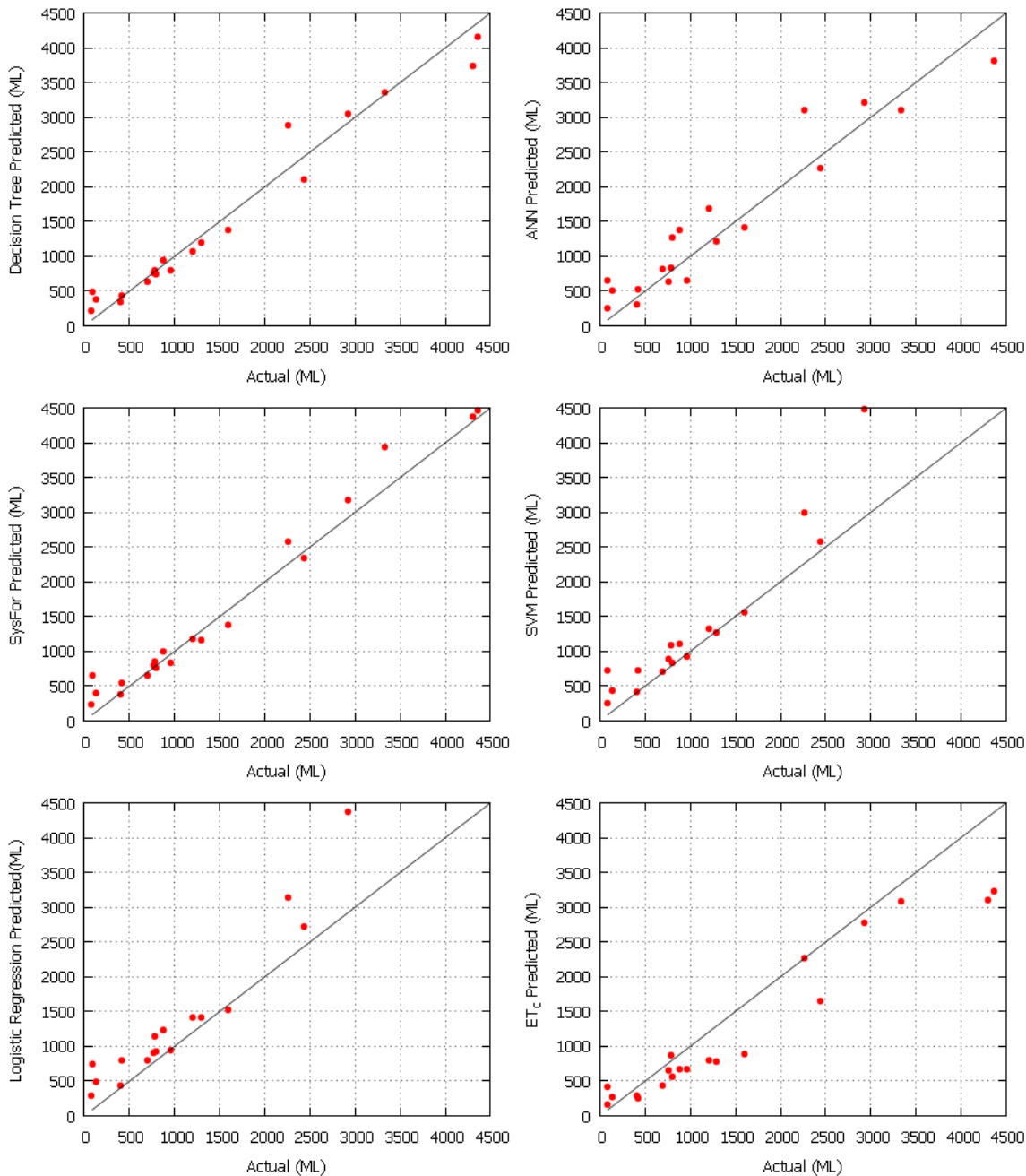


Figure 7: Actual Vs Predicted Water Usage made by six different models on 22 nodes of CIA

Therefore, we can say that the predictions made by SysFor and DT are close to actual water usage. Similarly, the scatter plots in Figure 7 shows the actual and predicted water usage made by all the models experimented in this study.

We also developed a web based Decision Support System (DSS) called Coleambally IRIS which consists of a database and collection of various models. Users (farmers and irrigation managers) access various data from DSS including water predictions made by our model as shown in Figure8. Based on our previous study we incorporated Decision Tree model in our DSS for predicting future water requirements. By using demand forecast results users will learn the water requirement for their particular farm for 7days in advance and can order the exact amount of water they require, this will increase the percentage of water savings and improve water use efficiency.

5 Conclusion

This study compares the effectiveness and performances of several data mining techniques such as decision tree, ANN, SysFor, SVM and logistic regression in predicting irrigation water demand. The novelty of this study is comparison of SysFor with other classification techniques which to our knowledge was carried out for the first time, and the application of pre-processed dataset on different classifier models.

Our experimental results indicate a minor difference in the prediction accuracies achieved by different data mining techniques mainly SysFor, Decision tree and ANN. Computational results demonstrate that based on 3 folds cross validation method multiple decision tree technique SysFor produce the best prediction accuracy of

78% followed by decision tree and SVM with 74% and 64% respectively.

We also compare the prediction accuracies of the models with the actual water consumed by the crop. The closeness of prediction accuracy of SysFor performs the best with 97.5% followed by decision tree with 96% accuracy. Interestingly, ANN performs better than SVM by closely predicting the water demand to actual water used with 95% accuracy. The accuracy predictions made by SVM, logistic regression and traditional ET_c method are found to be 78%, 75% and 77% respectively.

Therefore, from the above results we recommend that SysFor, decision tree and ANN techniques are most suitable for predicting irrigation water demand. By developing and implementing a demand forecasting model using these techniques the farmers and irrigation

managers of CIA can learn the future water requirement in advance accurately. Hence, this tool is crucial for effectively improving existing water management practices and maximising water productivity. Although the results obtained from this study are more significant for predicting water demand, the limitation would be use of less number of influential attributes in the dataset. This can be further improved by adding more attributes having high influence on crop water usage such as seepage, soil moisture, etc. In addition it would be interesting to explore the influence of cropping stage on crop water use. Furthermore, based on our results from this study we plan to incorporate SysFor model into our DSS to make the water predictions more accurate and reliable.

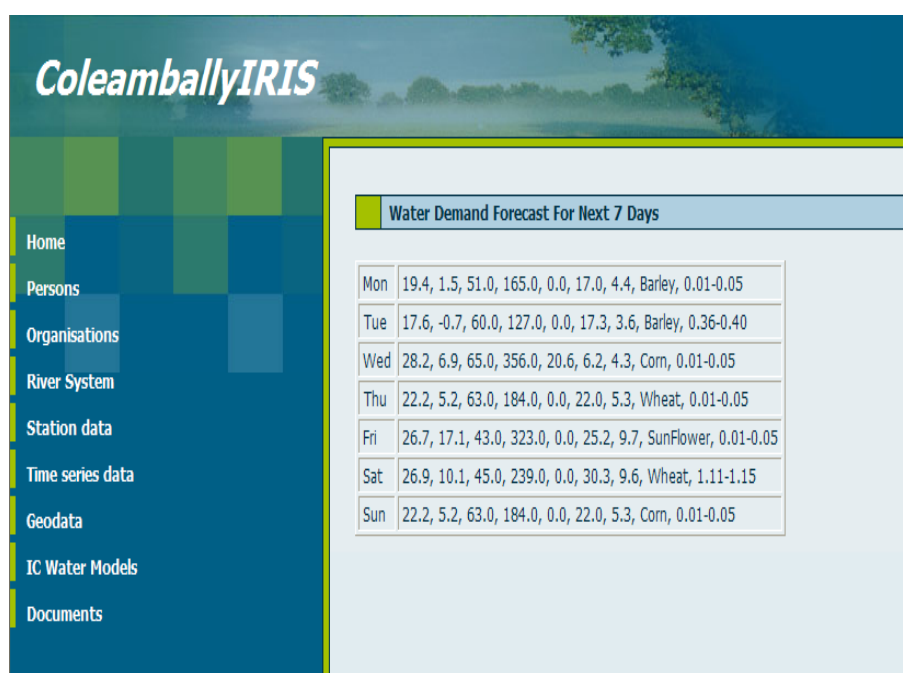


Figure 8: Irrigation water demand forecasting for 7 days

6 References

- 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.
- Ambrozic, T & Turk, G. (2003): Prediction of subsidence due to underground mining by artificial neural networks. *Computers and Geosciences*, 29, 627-637
- Cancelliere, A., Giuliano, G., Ancarani, A. & Rossi, G. (2002): A Neural Networks Approach for Deriving Irrigation Reservoir Operating Rules. *Water Resource Management*, 16, 71-88
- Christensen, R. (1997): Log-Linear Models and Logistic Regression. Springer.
- Coleambally Irrigation Company Limited (2011): Annual Compliance Report.
- Dawson, C.W. & Wilby, R.L. (2001): Hydrological modelling using artificial neural networks, *Progress in Physical Geography*, 25 (1), 80-108.
- De Vos, N.J. & Rientjes, T.H.M. (2005): Constraints of artificial neural networks for rainfall-runoff modelling: trade-offs in hydrological state representation and model evaluation. *Hydrology and Earth Systems Sciences*, 9, 111-126.
- FAO56, FAO Irrigation and Drainage Paper, <http://www.kimberly.uidaho.edu/ref-et/fao56.pdf> accessed on 25/7/2012.
- FAO, 1994 Water for Life. World Food Day 1994, Rome.
- Han, J. & Kamber, M. (2001): Data Mining: Concepts and Techniques. A Harcourt Science and Technology company. 525 B Street, Suite 1900, San Diego, CA 92101-4495, USA.
- IWMI (2009): Water for a food secure world. International Water management Institute (IWMI) Strategic Plan 2009-2013.
- 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. and Giggins, H. (2011): Knowledge Discovery through SysFor: A Systematically Developed Forest of Multiple Decision Trees. *In*

- Proceedings of the 9th Australasian Data Mining Conference (AusDM 11)*, Ballarat, Australia. Dec 01 - Dec 02, 2011. CRPIT, 121, 195-204.
- 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.
- Khan, M., Islam, M. Z., Hafeez, M. (2011): Irrigation Water Demand Forecasting – A Data Pre-Processing and Data Mining Approach based on Spatio-Temporal Data, In *Proceedings of 9th Australasian Data Mining Conference (AusDm11)*. Ballarat, Australia. Dec 01-Dec 02, CRPIT, 121, 183-194.
- Muttil, N. & Chau, K.W. (2006): Neural Network and Genetic Programming for Modelling Coastal Algal Blooms. *International Journal of Environment and Pollution*, Vol. 28, NO.3-4, 223-238.
- Pearce, J. and Ferrier, S. (2000): Evaluating the predictive performance of habitat models developed using logistic regression, *Ecological Modelin*, 133(3), 225-245
- 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.
- Quinlan, J. R. (1993) C4.5: *Programs for machine Learning*. Morgan Kaufmann Publishers, San Mateo, California, USA.
- Quinlan, J. R. (1996): Learning Decision Tree Classifiers, *ACM Computing Surveys*, 28:1
- Safer, A. M. (2003): A Comparison of two data mining techniques to predict abnormal stock market returns, *Intelligent Data Analysis*, 7, 3-13.
- 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.
- Olaiya, F., Adeyemo, A. B. (2012): Application of Data Mining Techniques in Weather Prediction and Climate Change Studies, *I.J. Information Engineering and Electronic Business*, 1, 51-59
- Ullah, K. and Hafeez, M. (2011): Irrigation Demand forecasting using remote sensing and meteorological data in semi-arid regions. In *Proceedings of Symposium J-H01 held during IUGG2011*, Melbourne, Australia, July 2011, 157-162.
- Vapnik, V. (1995): The Nature of Statistical learning Theory, Springer, New York.
- Wang, W.C., Chau, K.W., Cheng, C.T. & Qiu, L. (2009): A comparison of performance of several artificial intelligence methods for forecasting monthly discharge time series, *Journal of Hydrology*, 374 (2009), 294-306.
- Wu, C.L., Chau, K.W. & Fan, C. (2010): Prediction of rainfall time series using modular artificial neural networks coupled with data-preprocessing techniques, *Journal of Hydrology*, 389 (2010), 146-167.
- Yang, L., Dawson, C.W., Brown, M.R. & Gell, M. (2006): Neural network and GA approaches for dwelling fire occurrence prediction. *Knowledge-Based Systems*, 19, 213-219.
- Zhou, S.L. and McMohan, T.A., Walton, A. and Lewis, J. (2002): Forecasting operational demand for an urban water supply zone, *Journal of Hydrolog*, 259, 189-202.
- Zurada, J. and Lonial, S. (2005): Comparison of the Performnce of Several Data Mining Methods for bad Debt Recovery In The Health Industry, *The Journal of Applied Business research*, 21, 37-54.

Anytime Algorithms for Mining Groups with Maximum Coverage

Satya Gautam Vadlamudi

Partha Pratim Chakrabarti

Sudeshna Sarkar

Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
Kharagpur, West Bengal 721302, India
Email: {satya,ppchak,sudeshna}@cse.iitkgp.ernet.in

Abstract

Mining maximal groups from spatio-temporal data of mobile users is a well known problem. However, number of such groups mined can be very large, demanding further processing to come up with a readily usable set of groups. In this paper, we introduce the problem of mining a set of K maximal groups which covers maximum number of users. Such a set of groups can be useful for businesses which plan to distribute a set of K offers targeting groups of users such that a large number of users are covered. We propose efficient methods to solve this hard problem, which do not mine the total set of groups apriori (avoiding the large amount of time consumed upfront), instead intelligently decide during their execution as to which area of the search space is to be explored to mine the next group so that a set of K groups covering large number of users is quickly produced, and then improve the K -set as time progresses (anytime nature). Experimental results on several synthetic spatio-temporal datasets as well as real datasets that are publicly available show the efficacy and scalability of the proposed methods across various parametric inputs.

Keywords: Anytime Algorithms, Coverage, Spatio-Temporal Data Mining, Top K , Valid Groups

1 Introduction

Spatio-temporal mobility data of users consisting of $\langle \text{time}, \text{location} \rangle$ values for each member is of great value and several kinds of interesting information has been mined using such data like groups (Wang et al. 2003), trajectories (Lee et al. 2009), events and social-networks (Lauw et al. 2005), association rules and significant locations (Verhein & Chawla 2006), moving clusters (Kalnis et al. 2005), etc. Specifically, mining group patterns from mobility data can be very useful in target marketing (Schafer et al. 2001), social network analysis (Forsyth 2006), crime investigation (Xu & Chen 2005), habitat monitoring (Davis et al. 2004), and various other applications.

A criterion for a set of users to be designated as a *valid group* is proposed in the work (Wang et al. 2003), based on the proximity in their spatio-temporal mobility (see Section 2 for full details). Later, methods for mining all *maximal* valid groups

are proposed in order to avoid redundancy (Wang et al. 2008) which arises because all subsets of a valid group also become valid groups. Although the amount of redundancy is greatly reduced by mining only the maximal valid groups instead of mining all the valid groups, the number of groups mined still remains very large to be used readily (order of *one million* for an instance with 6,000 users, see (Wang et al. 2008)). Also, mining all maximal valid groups can be extremely time consuming; in our experiments, for an instance with 5,000 users, VGBK (Wang et al. 2008), an efficient algorithm, could not terminate with the set of all maximal valid groups even after sixty hours.

In this scenario, we propose the problem of finding a set (of cardinality K) of maximal valid groups which covers maximum number of users.

Such a set could be extremely useful in reaching out to a large number of users through a limited set of groups. For example, a business owner or a political outfit can advertise to a large number of people through a limited set of attractive offers. Note that, even if the offer itself does not involve all the members of a group rather only one person per group is chosen, it is still extremely likely that the advertisement reaches/influences maximum number of people by virtue of the groupings. Given the set of all maximal valid groups, the problem is well known as maximum coverage problem which is NP-hard (Hochbaum 1997) (the hardness of the maximum coverage problem holds in our case as well since there exist no generic relations between the maximal valid groups which can trivialize its complexity). Therefore, the search space for finding the set of groups with maximum coverage is exponential in the size of the set of all maximal valid groups.

Such a huge search space poses challenge in all aspects: time, memory, and solution quality. Therefore methods which can work within the given amount of memory and can give a good quality solution within the given amount of time are needed. None of the existing methods can be directly used for this purpose and substantial adaptation is needed if they were to be used. In this work, we explore several algorithms which are adaptations of existing mining techniques, and also develop new methods based on depth-first and best-first search techniques which are of low memory footprint and yet give good anytime performance. *Anytime* (Dean & Boddy 1988), meaning, they readily produce a K -set of groups as soon as they can and improve upon the K -set as time progresses, thereby making a good quality K -set available to use at any time.

Note that, our problem has two aspects to it: 1) mining the maximal valid groups, and 2) choosing the K -set efficiently. For addressing the first aspect, efficient algorithms were proposed in the litera-

Copyright ©2012, Australian Computer Society, Inc. This paper appeared at the 10th Australasian Data Mining Conference (AusDM 2012), Sydney, Australia, December 2012. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 134, Yanchang Zhao, Jiuyong Li, Paul Kennedy, and Peter Christen, Ed. Reproduction for academic, not-for-profit purposes permitted provided this text is included.

ture (Wang et al. 2008), and for addressing the second aspect, there are no clear recommendations in the literature, which need to be explored. In this paper, we identify that best-first search algorithms such as the ones based on A* (Hart et al. 1968) which are used for combinatorial optimization can be of help in this case since the mining algorithms proceed in a search algorithm like manner as well. Hence, we explore two types of algorithms, ones which are mining oriented which we adapt for optimization (BKC, DFRC; explained in Section 3), and ones which are optimization oriented which we adapt for mining (ItBC, MAC; explained in Section 3). We observe that each of these algorithms has its strengths and weaknesses and is suited for a particular type of situation. We categorized such cases and identified the corresponding best algorithms in each case, which can be suitably picked as per the needs of the application.

A conceptually closely related and well-studied problem is that of mining frequent itemsets from a given database with a given minimum support and confidence (Agrawal & Srikant 1994). Since the number of such frequent itemsets are usually very large, mining of *closed* frequent itemsets (Pei et al. 2000), *non-derivable* frequent itemsets (Calders & Goethals 2002), and *maximal* frequent itemsets (Gouda & Zaki 2005) were explored. Additional criteria such as support constraints (Wang et al. 2000), item constraints (Srikant et al. 1997), etc. were used to narrow down the number of itemsets. Mining top-k frequent itemsets was proposed to find the most significant itemsets from a database (Han et al. 2002). Redundancy-aware top-k patterns mining (Xin et al. 2006) was proposed to increase the diversity among the top-k patterns while maintaining significance. Compression of itemsets was done by using condensed representations (Pei et al. 2002). Recent work on frequent itemsets called *summarization* focuses on obtaining a representative set of frequent itemsets which can act as a summary of the whole set of frequent itemsets (Afrati et al. 2004). Also, work has been carried out on mining high utility itemsets (Tseng et al. 2010) where cost of different items, their count in a transaction, and the number of transactions they are involved in are taken into account.

Translation of our objective in the context of frequent itemset mining gives rise to mining K maximal frequent itemsets which cover maximum number of items. Perhaps the closest work in spirit to this is of mining redundancy-aware top-k patterns (Xin et al. 2006) where diverse itemsets are covered via the incorporation of pattern redundancy measure. The proposed coverage problem may be very helpful in coming up with a limited number of recommendations/offers that span a wide variety of items. However, in this paper, we restrict ourselves towards mining the user groups with maximum coverage. Also note that, mining group patterns is different from clustering (Ng & Han 1994) and often has relatively more complex criteria.

This paper makes the following key contributions:

1. We introduce the problem of mining K groups such that maximum number of users are covered and explain the utility and importance of such a set.
2. We propose efficient algorithms for solving this problem which are of low memory footprint and which give good anytime performance.
3. We present extensive empirical analysis using synthetic as well as real spatio-temporal datasets

which are publicly available. We vary the different mining parameters, the simulator parameters in case of synthetic data, K value, and data size. We analyze the anytime performance (solution quality, time taken) as well as memory consumption. Thereby thoroughly testing the efficacy and scalability of the proposed methods.

The rest of the paper is organized as follows: Section 2 presents the required background on the valid group definition, existing algorithms for mining all maximal valid groups, and the best first search algorithms. In Section 3, we present the proposed methods for mining groups with maximum coverage along with their properties. In Section 4, we present the empirical results demonstrating the efficacy of the proposed methods. We conclude in Section 5.

2 Background

In this section, we briefly present the required background on the definition of valid group, algorithms for mining maximal valid groups, and some best first search approaches that are used later.

Valid group (Wang et al. 2003). The following definitions related to a set of users G lead to the definition of a valid group.

Valid Segment: A valid segment is a set of consecutive timepoints $[t_a, t_b]$ where the set of users G are within *max_dis* distance of each other at each timepoint and the length of the segment is at-least *min_dur* (also the segment has to be maximal, meaning, all users of G are not together at $t_a - 1$ and $t_b + 1$).

Group Pattern: A set of users G , thresholds *max_dis*, *min_dur* form a group pattern if G has a valid segment.

Weight: Weight of a group pattern is the sum of the lengths of all valid segments of that group.

Valid Group: A set of users G is called a valid group if they are part of a group pattern whose weight exceeds a threshold *min_wei*.

Maximal Valid Group: A valid group is called a maximal valid group if it is not a subset of any other valid group.

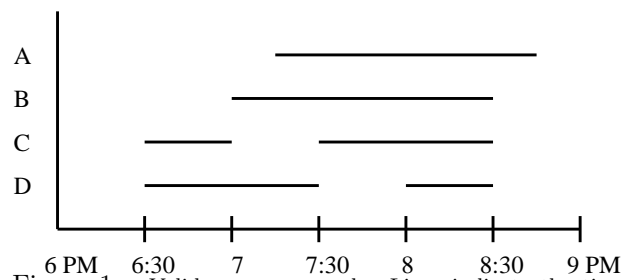


Figure 1: Valid groups example: Lines indicate the time spent by the users A, B, C, and D at a restaurant.

Figure 1 shows the time spent by four users at a restaurant. Assume that the restaurant is identified as a single location (they will be within *max_dis* whenever they are at the restaurant) and the values of *min_dur* and *min_wei* to be 30 Min. and 1 Hour respectively. In such a scenario, the maximal valid groups are: {A,B,C}, {B,C}, {B,D}, and {C,D} (note that, B, C, and D are not all together for *min_wei* time for them to become a valid group).

Mining maximal valid groups (Wang et al. 2008). Two efficient methods, namely, VGMax and VGBK were proposed for mining all the maximal valid groups.

VGMax works as follows: A graph known as VG-graph is formed where the vertices represent the users

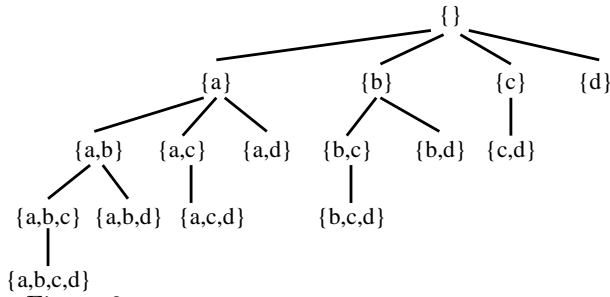


Figure 2: Set enumeration tree consisting of four users.

and each edge represents the set of valid segments between each pair of users. An edge is present between a pair of vertices in VG-graph *iff* the corresponding users form a valid group (i.e., the total length of their valid segments must be at-least *min_wel*). Next, a vertex is extracted from the graph, and a new graph called conditional VG-graph is generated. The conditional VG-graph contains those vertices which form a valid group with the extracted vertex/user. Edges are updated to contain those sets of valid segments which are also shared by the extracted user. This process is repeated in a depth-first manner to generate conditional VG-graphs by extracting vertices one at a time (using the recent conditional VG-graph) until a conditional VG-graph that contains no edges is generated. Whenever a vertex is generated in the conditional VG-graph which does not have any edges incident on it, the extracted set of users and that vertex/user constitute a potential maximal valid group. Since the algorithm traverses the set enumeration tree (Rymon 1992), subgroups of a maximal group will also be recognized as potential maximal valid groups. For example, in Figure 2, if $\{a,b,d\}$ is found to be a maximal valid group, then later on $\{b,d\}$ will also be recognized as a potential maximal valid group. Therefore, *MaximalityChecking* is performed which checks whether any superset already exists in the solution set before adding a potential maximal group to it.

VGBK, inspired by the Bron-Kerbosch algorithm (Bron & Kerbosch 1973), also works in similar manner as VGMax but gets away with *MaximalityChecking* by carefully keeping track of the users (say S) that are ignored by VGMax in its traversal but can form a valid group with the current extracted set. Before adding a valid group g to the set of maximal valid groups, it checks whether S is empty or not. If it is not empty, it means that there exists users which can form a valid group with g , implying g to be non-maximal, and vice-versa.

Best-first search/Heuristic search. Given a search graph/tree, it can be mainly explored in three ways: depth-first, breadth-first, or best-first. Best-first exploration involves an estimation/value corresponding to each node denoting its promise towards leading to a goal node. Based on such heuristic estimates, most promising node is selected and expanded until a goal node is found, at which point either the search is terminated or continued to find better/other solutions.

There are several best-first search based anytime algorithms (Zhou & Hansen 2005, Aine et al. 2007, Thayer & Ruml 2010, van den Berg et al. 2011, Vadlamudi et al. 2012) that are potential candidates which can be used here. However, our situation demands that the algorithms should be able to work within the given memory on large sized problems in which case the following two methods appear to be more promising (which we will adapt to solve our problem later in the paper):

Beam search (Bisiani 1987). It is one of the simplest heuristic search methods. Given a beam-width b , it expands b most promising nodes at each level of the search tree in breadth-first manner until a goal-node is found. It uses $2 * b$ nodes of memory for its execution on a search tree.

MAWA* (Vadlamudi et al. 2011). Memory-bounded Anytime Window A* is a memory bounded anytime heuristic search algorithm based on Anytime Window A* (AWA*) (Aine et al. 2007) and Memory-bounded A* (MA*) (Chakrabarti et al. 1989). Since A* (Hart et al. 1968) proceeds in a purely best-first manner, it takes a long time for finding a solution. AWA* adds a depth-component to A* using a window based restriction while exploring in a best-first manner so that good quality solutions can be found quickly. However, like A*, AWA* also runs out of memory while exploring large search spaces. MAWA* was developed by effectively combining AWA* and MA* to give good anytime performance while working within the given amount of memory. It was shown to be effective on diverse search spaces, especially, when using very low memory.

3 Proposed Methods

In this section, we present the proposed algorithms for mining K maximal valid groups that cover maximum number of users, which guarantee maximal coverage. Firstly, we present a generic routine which is used by all the proposed methods for managing the K -set, then we present the methods based on existing mining techniques, followed by the methods based on depth-first search, followed by the methods based on best-first search.

Maximizing coverage. All algorithms can be viewed to be having two parts in-built, one that mines maximal valid groups, and the other which takes the mined maximal valid groups as they are produced and updates the current best solution (the set of K groups) so as to maximize the coverage. Here, we introduce the latter part which takes in a maximal valid group and updates the current solution set.

The MaximizeCoverage routine (see next page) presents the strategy for maximizing coverage. Initially, the solution set is empty, coverage is zero, and the counts of coverage for each user are all zero. The method keeps on adding a new group (if it is not already present) to the solution set until size K is reached. Then onwards, upon getting invoked with a new maximal group, it efficiently replaces a group from the solution set with the new group such that the overall coverage is maximized. Note that the overall coverage c is only increased (by 1) when an individual user coverage count in A raises from 0 to 1. Similarly, c is only decreased (by 1) when an individual user coverage count in A drops from 1 to 0. c remains unaffected at all other times.

3.1 VGBK based Algorithm

Algorithm BKC. Given the above methodology for obtaining the best possible set of K groups¹, any algorithm mining the maximal valid groups can be linked to it to form a mining algorithm that finds K groups with maximum coverage.

Algorithm 1 presents the routine which combines VGBK and MaximizeCoverage methods to mine K groups with maximum coverage. *MineVGgraph* and

¹In this paper, we use the terms *group*, *maximal group*, *valid group* and *maximal valid group* synonymously unless otherwise being specified in the context.

MaximizeCoverage

```

1: INPUT :: A maximal valid group  $g$ , current so-
   lution set of maximal valid groups  $G$ , number of
   groups  $K$ , number of users covered by  $G$  -  $c$ , and
   an array  $A$  holding the counts of how many times
   each user is covered in  $G$ .
2: if  $|G| < K$  then
3:    $G \leftarrow G \cup g$ ; Update the counts in  $A$  and cover-
   age  $c$  if  $G$  is changed;
4:   Return  $G, c, A$ ;
5: end if
6:  $old\_c \leftarrow c$ ;
7:  $G \leftarrow G \cup g$ ; Update the counts in  $A$  and coverage
    $c$ ;
8: if  $c - old\_c = 0$  then
9:    $G \leftarrow G \setminus g$ ; Update the counts in  $A$ ;
10:  Return  $G, c, A$ ;
11: end if
12:  $max\_gain \leftarrow 0$ ;  $max\_g \leftarrow g$ ;
13: for each  $g' \in G$  other than  $g$  do
14:    $G \leftarrow G \setminus g'$ ; Update the counts in  $A$  and cov-
   erage  $c$ ;
15:   if  $c - old\_c > max\_gain$  then
16:      $max\_gain \leftarrow c - old\_c$ ;  $max\_g \leftarrow g'$ ;
17:   end if
18:    $G \leftarrow G \cup g'$ ; Update the counts in  $A$  and cov-
   erage  $c$ ;
19: end for
20:  $G \leftarrow G \setminus max\_g$ ; Update the counts in  $A$  and
   coverage  $c$ ;
21: Return  $G, c, A$ ;

```

Algorithm 1 VGBK based algorithm for Coverage (BKC)

```

1: INPUT :: A spatio-temporal data set  $\langle u_i, t_i, l_i \rangle$ ,
    $max\_dis$ ,  $min\_dur$ ,  $min\_wei$ , and  $K$ .
2:  $vg \leftarrow \text{MineVGgraph}()$ ;  $G \leftarrow \phi$ ;  $c \leftarrow 0$ ;
    $\forall i A(i) \leftarrow 0$ ;  $old\_c \leftarrow -1$ ;
3: while  $c \neq old\_c$  do
4:    $old\_c \leftarrow c$ ;
5:   Invoke VGBK() with  $vg$  and use Maximize-
     Coverage() to update  $G, c, A$  whenever a new
     maximal valid group  $g$  is mined;
6: end while
7: Return  $G, c$ ;

```

VGBK routines are borrowed from (Wang et al. 2008). VG-graph contains information on all valid groups of size 2. VGBK algorithm involves mining VG-graph as its first step, based on which conditional VG-graphs are generated, used and dissolved in future steps. We separate the VG-graph mining in order to avoid re-generating it in each invocation of VGBK routine. G denotes the set of K maximal valid groups with coverage c . We choose VGBK against VGMax as it does not require *MaximalityChecking* (Wang et al. 2008) (which needs all the currently mined maximal valid groups to be stored in the memory which can be very large) used in VGMax in order to produce maximal valid groups.

Since VGBK involves a depth-first like traversal of the generic set enumeration tree (Rymon 1992), the above algorithm may take a long time in order to produce a good quality set G (especially when the groups being mined are of large size which results in large unuseful subset space). Usually, one may expect to complete at-least one full traversal of the set enumeration tree in order to attain good coverage. Also, VG-graph mining can take quite a lot of

time for data of large number of users, without which the group mining process does not begin, which can hamper the anytime performance (as it causes a large delay at the beginning).

3.2 Depth-First Search based Algorithm**Depth-First search algorithm for mining all Maximal valid groups (DFMax)**

```

1: INPUT :: A spatio-temporal data set  $\langle u_i, t_i, l_i \rangle$ ,
    $max\_dis$ ,  $min\_dur$ ,  $min\_wei$ , solution set  $G$  (ini-
   tially empty), and the current set of users which
   form a valid group  $g$  (initially empty).
2: if  $g = \phi$  then
3:   for  $i = 1$  to  $NUM\_USERS$  do
4:     DFMax(group with single member  $i$ );
5:   end for
6:   Return;
7: end if
8: for  $i = (\text{largest } uid \text{ of } g) + 1$  to  $NUM\_USERS$ 
   do
9:   if  $g \cup i$  is a valid group then
10:    DFMax( $g \cup i$ );
11:   end if
12: end for
13: if no  $i$  could form a valid group with  $g$  then
14:   MaximalityChecking( $g, G$ );
15: end if

```

The DFMax routine presents how all the maximal valid groups could be mined using a depth-first mechanism on the set enumeration tree (Rymon 1992) or a lexicographic tree similar to that of (Agarwal et al. 2000). *MaximalityChecking* routine is borrowed from (Wang et al. 2008). Testing whether the addition of i to g will form a valid group can be done efficiently by storing the valid segments for each group in the data-structure of that group. Then, one can simply perform intersections among the valid segments of the group and the valid segments of 2-groups $\langle i, j \rangle \forall j \in g$ to get the valid segments of $g \cup i$. If the weight of resulting valid segments at any point drops below min_wei while performing intersections, then $g \cup i$ could be right away declared as not valid. Note that, *valid segments* of all groups of size 2 are handled via dynamic programming. This is essentially similar to VG-graph except that it is generated on demand and stored for later use, avoiding big lapse in time early. There are no conditional VG-graphs generated which distinguishes DFMax from VGMax.

The algorithm as such is not as efficient as VGMax or VGBK. However, it requires less amount of memory than VG-graph based algorithms because of not having to generate conditional VG-graphs. It is exactly for this reason it involves a large duplication effort in generating *valid segments* of groups of sizes > 2 , which ultimately amounts to larger time for mining all maximal valid groups. In spite of this or we can say because of this, it could be very apt for use in mining groups with maximum coverage. Since it does not require to generate conditional VG-graphs, it reports the first maximal group very quickly. We use this property effectively to come up with our next algorithm for coverage.

Algorithm DFRC. Algorithm 2 presents the DFMax based method for solving the coverage problem. The idea is to traverse through those set of vertices/users first in each iteration which are not currently covered, which would ensure quick coverage of all the users. Whenever a maximal group is mined

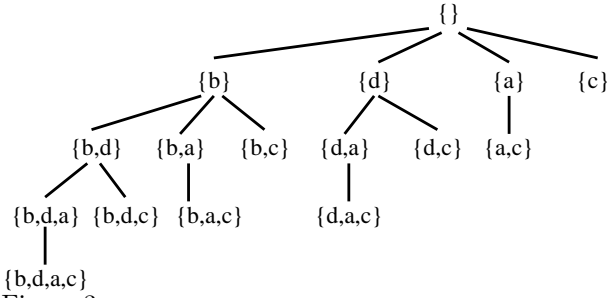


Figure 3: Working of DFRC: Set enumeration tree consisting of four users when they are reordered assuming that a and c are covered.

that increases the overall coverage, the current execution of DFMax is aborted and a new execution is begun which focuses on the new set of uncovered users first. For example, as shown in Figure 3, assuming that a and c are covered currently, set enumeration tree is traversed such that groups involving b and d are mined next. Here, we may need to keep all the maximal valid groups being mined in memory which are necessary for the MaximalityChecking routine. However, since we are only interested in K groups here, we may want to get away with this type of MaximalityChecking operation to save space and time. One way is to follow similar strategy as that of VGBK (which is inspired from the *not* set of Bron-Kerbosch algorithm (Bron & Kerbosch 1973)), but that can become costly to maintain as DFMax does not have conditional VG-graphs for efficiently handling it.

Algorithm 2 DFMax based Re-ordering algorithm for Coverage (DFRC)

- 1: **INPUT** :: A spatio-temporal data set $\langle u_i, t_i, l_i \rangle$, max_dis , min_dur , min_wei , and K .
 - 2: $G \leftarrow \phi$; $c \leftarrow 0$; $\forall i A(i) \leftarrow 0$; $old_c \leftarrow -1$;
 - 3: **while** $c \neq old_c$ **do**
 - 4: $old_c \leftarrow c$;
 - 5: Re-order users in the increasing order of $A(i)$;
 - 6: Invoke **DFMax()** to traverse the set enumeration tree in the above order and use **MaximizeCoverage()** to update G, c, A whenever a new maximal valid group g is mined. Abort DFMax when c is increased;
 - 7: **end while**
 - 8: Return G, c ;
-

Another way is to perform MaximalityChecking in a different manner, namely, by checking whether any user could be added to the potential group and consider all the users in doing so. Previous knowledge of whether a user was already found to be not compatible with the group (or its parent) during DFMax (line 9) can also be used to speed up MaximalityChecking.

Interestingly, here, one may skip maximality checking methods entirely once the first K -set is obtained and directly use MaximizeCoverage to check for improvement in the coverage of K -set since the first group to improve the coverage will always be maximal (note that, supersets are always explored first in the set enumeration tree). Though this involves invoking MaximizeCoverage with a number of non-maximal groups, this strategy has been proved to be the best in our experiments and hence preferred.

Note that, a BKC like version (traversing entire set enumeration tree) using DFMax would not be effective since it would not have the advantage of conditional VG-graphs for quick traversal. Also, a DFRC

like version (restarting after first improvement) using VGBK/VGMax would not be effective since it would have the unnecessary overhead of conditional VG-graphs which will be thrown away after each iteration as they will not be of any use when the traversal order changes.

3.3 Heuristic Search based Algorithms

In the following, we present the five components using which, one can use any heuristic-search method to mine a valid group covering most number of uncovered users (based on the input):

Start state. The start state is same as the root node of the set enumeration tree which we have also used in DFMax. It consists of empty set of users.

Child generation. Child generation also follows closely with the set enumeration tree and DFMax where exactly one more (and each) lexicographically superior user is added to the set of users in the current state, and the child state is generated if that group is valid.

f-value computation. This is the crucial element which guides the search. f -value is obtained by the addition of two values, namely, g (denoting the cost from start node to current node) and h (denoting the estimated cost from current node to a goal node). We obtain g -value by counting the number of uncovered users in the current set. The h -value is set as the number of uncovered users among the lexicographically eligible set that can form a valid group with the set of users in current node. Note that, this number over-estimates the ideal h -value as all those users may not simultaneously form a valid group with the current set of users (we only know that they form a valid group with the current set individually). Over-estimating heuristics (in case of a maximization problem) are also called admissible heuristics since they do not undermine the significance of a node. The h -value estimation can be quickened by using the information about users which are known to be not forming valid groups with the parent (and ancestors) of current node (such information is already obtained during h -calculation of its ancestors). Finally, $f = g + h$. (f -value of start state is NUM_USERS). Figure 4 shows the set enumeration tree along with g and h values of different nodes when a and c are already covered via maximal valid group $\{a, c\}$ and assuming that any user can form a valid group with any current set of nodes other than $\{a, c\}$. Note that, only valid children are shown in the figure so that their g and h -values make sense.

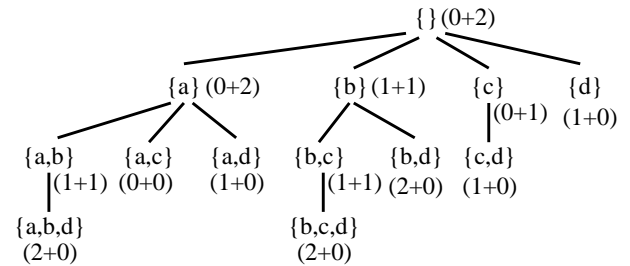


Figure 4: Search tree consisting of four users. f -values corresponding to best-first search are given for each node as $(g + h)$, assuming that a and c are covered via maximal valid group $\{a, c\}$ and any user can form a valid group with any set of users other than $\{a, c\}$.

Goal condition. A node is said to be a goal node if it can not have any valid children (the node itself and its ancestors (except for start node) must all be valid children). Note that, since the traversal is on a set enumeration tree (which has all possible sets as its leaf

nodes), a goal node obtained via anytime best-first search need not translate into a maximal valid group at all times. Therefore either maximality checking like mechanisms (like the ones presented in previous subsection) can be used to determine the maximality of a group or further processing may be done to produce a maximal group containing the current set of users of goal node. The former is good for use in BKC or DFRC as a maximal group is always mined ahead of its subgroups, and therefore we may like to ignore the subgroups. Whereas here, the heuristic-search algorithm in a way promises that the goal node is good in terms of coverage (though may not be maximal), and hence we may like to use it rather than throwing away. The technique for making the set of users of goal node into a maximal valid group is similar to that of the maximality checking used in DFRC except that here we keep on adding all eligible users to the set which form a valid group (instead of reporting whether the current group is maximal or not).

Admissible pruning mechanism. This is not an essential part to use heuristic search algorithms but a very important part in order to speedup the search. Admissible pruning refers to the deletion of certain nodes (and therefore their subtrees) which does not effect the results of the search, in other words, the pruning of nodes that are guaranteed to be of no use. Here, we can prune all the nodes whose f -value is zero, as they will not be able to increase the coverage (we can guarantee this since we have used an admissible heuristic for computing the f -value).

Now, we explain how two efficient heuristic-search algorithms, namely, beam search (Bisiani 1987) and MAWA* (Vadlamudi et al. 2011) can be adapted to mine groups with maximum coverage. The choice of these two particular algorithms has been explained in Section 2.

We have managed the *valid segments* of all groups of size 2 via dynamic programming, similar to how we did it in case of DFRC.

As explained in Section 2, beam search expands *beam-width* number of nodes at each level until a goal node is found. For mining a group quickly that increases coverage, we may like to start with beam-width value 1, and increase the beam-width to explore more search space only if needed. The ItB routine presents this technique which stops as soon as coverage is improved.

Iterative Beam search (ItB)

- 1: **INPUT** :: A spatio-temporal data set $\langle u_i, t_i, l_i \rangle$, max_dis , min_dur , min_wei , K , MAX_BEAM , solution set G , its coverage c , and array A with counts of number of times each user is covered in G .
 - 2: $old_c \leftarrow c$;
 - 3: **for** $i = 1$ to MAX_BEAM **do**
 - 4: Invoke **Beam**(i) with the aforementioned heuristic-search components such that the final maximized goal node is fed to **MaximizeCoverage**() to update G, c, A ;
 - 5: **if** $c > old_c$ **then**
 - 6: Return;
 - 7: **end if**
 - 8: **end for**
-

Algorithm ItBC. Given a set of groups G , the above mining algorithm finds a group which increases its coverage. Now, we iteratively invoke the above algorithm to come up with ItBC (similar to how we iteratively used VGBK). Algorithm 3 presents this technique. Since it may take a while to produce a

starting solution due to the heuristic calculations involved, it is initialized with DFRC which can give the first K-set quickly and also blend in easily as it too uses dynamic programming strategy for managing *valid segments* of groups of size 2. Initialization with DFRC improved the anytime performance of best-first search based algorithms.

Algorithm 3 ItB for Coverage (ItBC)

- 1: **INPUT** :: A spatio-temporal data set $\langle u_i, t_i, l_i \rangle$, max_dis , min_dur , min_wei , K , and MAX_BEAM .
 - 2: $G \leftarrow \phi$; $c \leftarrow 0$; $\forall i A(i) \leftarrow 0$;
 - 3: Run **DFRC**() until K groups are found;
 - 4: $old_c \leftarrow 0$;
 - 5: **while** $c \neq old_c$ **do**
 - 6: $old_c \leftarrow c$;
 - 7: **ItB**(G, c, A, MAX_BEAM);
 - 8: **end while**
 - 9: Return G, c ;
-

Algorithm MAC. We now present the algorithm when MAWA* is adapted for mining K groups with maximum coverage. MAWA* can be used iteratively to obtain the K -set similar to how Iterative Beam Search has been used. This works fine as long as the first solution produced by MAWA* keeps on improving the K -set. However, when the first solution does not improve the K -set and other solutions need to be searched to improve the K -set, MAWA* can sometimes get into an infinite loop driving towards the same solution. This situation does not arise when it finds more than one solution to give anytime performance in case of traditional optimization problems because, there the solution paths already seen will be cutoff based on f -value as better solutions are being looked for, so the previous solutions will not be found again. However, in our case, we do not obtain any particular cutoff upon discovering the first solution or n^{th} solution to prune previously explored goal nodes.

The solution to this is as follows: one needs to inform the parent of a goal node that the child be no longer considered in future once it is processed. That is, for the rest of the search (of that iteration), that child is always completely ignored. However, upon deleting the parent node itself due to memory limit, this information may get lost and the same goal node may again be generated in future, and the algorithm can still get into an infinite loop. Upon careful observation, it can be seen that this situation is similar to the problem faced when the memory given is limited where same path is explored again and again. To handle this, we use the same remedy used by MA*, the *backup* operation (see (Chakrabarti et al. 1989, Vadlamudi et al. 2011) for details). Therefore, one also needs to perform the *backup* operation after processing the goal node and informing the parent. It involves updating the f -value of parent with maximum of f -values of all its children other than the node(s) to be ignored, and recursively *backing up* all its ancestors. This will ensure that the algorithm though may sometimes regenerate a same goal node, will not do so infinitely often before finding another solution. Detailed proof for this can be obtained on the similar lines of the termination proof of MA* (Chakrabarti et al. 1989). We call MAWA* with the above modification as **Modified-MAWA***. Algorithm 4 shows how this can be iteratively used to construct MAC. Like in ItBC, here too we use DFRC for initialization for good anytime performance.

Algorithm 4 MAWA* based algorithm for Coverage (MAC)

```

1: INPUT :: A spatio-temporal data set  $\langle u_i, t_i, l_i \rangle$ ,
    $max\_dis$ ,  $min\_dur$ ,  $min\_wei$ ,  $K$ , and  $MAX$ 
   (memory limit in terms of number of nodes).
2:  $G \leftarrow \phi$ ;  $c \leftarrow 0$ ;  $\forall i A(i) \leftarrow 0$ ;
3: Run DFRC() until  $K$  groups are found;
4:  $old\_c \leftarrow 0$ ;
5: while  $c \neq old\_c$  do
6:    $old\_c \leftarrow c$ ;
7:   Invoke Modified-MAWA*( $G, c, A, MAX$ )
   with the aforementioned heuristic-search
   components such that the final maximized
   goal node is fed to MaximizeCoverage() to
   update  $G, c, A$ . Abort Modified-MAWA* when
    $c$  is increased.
8: end while
9: Return  $G, c$ ;
```

3.4 Properties

No maximal valid group exists which can replace a valid group from the K -sets produced by *BKC*, *DFRC*, and *MAC* at termination.

This is easy to observe as all these three algorithms (being complete in nature) sweep the entire space of maximal valid groups in their last iteration to find any maximal valid group which can replace a group from their current K -sets improving the coverage. However, note that, these K -sets may be different, because replacing one group at a time from the K -set can only mean that the K -set produced is a *local maxima* (hence the algorithms guarantee to produce *maximal* solutions only), which can be many. ItBC on the other hand does not hold this property as it is not complete for any fixed beam-width.

4 Experimental Results

We now present the experimental results obtained on various spatio-temporal datasets. These include the ones obtained from three data generators (also called as mobility simulators or simply simulators): Random Waypoint model (Broch et al. 1998) based simulator, Oporto simulator (Saglio & Moreira 2001), and Network based data generator (Brinkhoff 2002). It may be noted that, IBM city simulator used in (Wang et al. 2008) is no longer available online. Real datasets are scarcely available due to privacy concerns. We present experiments with one real dataset—the popular Geolife dataset collected and provided by Microsoft (MicrosoftResearchAsia 2012) for research purposes. All the algorithms are implemented in C++. All the experiments have been performed on a machine with Intel Core2 Duo CPU at 2.93-GHz and 2.92-GB RAM.

4.1 With Random Waypoint Simulator Data

For generating synthetic data, we developed a simple yet effective simulator inspired from random waypoint model (Broch et al. 1998), whose details are as follows:

The place of activity is square shaped integer grid whose dimensions are calculated as per the population density and number of users. For example, area of the square grid for 1000 users at a density of 25,000 users/sq.km. is 40,000 sq.m. which results in length of each side of the square to be 200 metres (units). Starting location for each user is chosen randomly

on the integer grid. For each timepoint, each user stays at his/her previous location with a probability of 0.5. When the user moves, he/she moves a single step (unit) along x/y-axis in one of the four directions with equal probability (note that, they may slightly get off the grid while moving, no restriction is put).

Factors affecting performance. We observed that the performance of the algorithms is mainly affected by three things: group size distribution of all the maximal valid groups (especially the presence of large sized groups), number of groups K to be mined, and the size of the data (number of users, and number of timepoints). We show representative results on each of the related aspects (results with all possible combinations of parameter values are skipped due to space constraint).

There are several parameters which can be varied: number of users, number of timepoints, population density, max_dis , min_dur , min_wei , and K . Group size is mainly affected by population density, max_dis , min_dur , and min_wei . However, we show results with varying max_dis only (affect of variation in other parameters leading to different group sizes is observed to be similar). We also present results with variation in K and with different sizes of data varying both number of users and timepoints.

Performance measures. The performance of the algorithms can be measured on three fronts: solution quality, time, and memory. The first two are covered by studying the plots showing anytime performance and we provide details on the memory used separately.

Upper bound and Estimate for Coverage.

An upper bound for the coverage obtained by K groups can be the sum of sizes of K largest groups (amongst all maximal valid groups). A rough estimate for the coverage can be $K * \text{Average group size}$. These values can only be known for small datasets where one can mine all groups and can then test the quality of coverage attained. In case of large datasets, the estimate can be $K * \text{Expected average group size}$.

Environment settings. Density is set to 25,000/sq.km. (which is a typical value of a city in today's world²), no. of timepoints: 1000, min_dur : 3 timepoints (30Min.; assuming 10Min. spacing), and min_wei : 1% (of total number of timepoints; equals 10 timepoints) in all the cases. We use a simpler closeness metric than Euclidean distance to quicken the algorithms, namely, two users are declared close if both x-axis and y-axis distances between them are not more than max_dis . The ItBC algorithm is run with 200 beam-width (corresponds to 200×2 nodes of memory as discussed in Section 2), and the MAC algorithm is run with 100 node memory (sufficient as its expected to be greater than maximum depth (group size)) in all our experiments.

Results on a Small dataset. No. of users: 100. For this set, the total number of valid groups mined are (using VGBK): 2828 with an average group size of 6.026. The distribution of the groups is as follows: 2: 13, 3: 179, 4: 462, 5: 628, 6: 549, 7: 343, 8: 290, 9: 209, 10: 74, 11: 49, 12: 18, 13: 11, 14: 3, where $x : y$ denotes that there are y groups of size x . The time taken is just 2 seconds.

Now, we find the top K groups for coverage using the proposed algorithms on the same dataset. Table 1 shows the coverage attained by different algorithms and the time taken for different values of K .

Note that, the time taken to produce the K -sets is less than the total time taken for mining all maximal

²<http://www.citymayors.com/statistics/largest-cities-density-125.html>

Table 1: Coverage attained by the proposed algorithms and the time taken in brackets (Seconds) for different values of K on Random Waypoint simulator data.

Algo.	K				
	20	24	25	26	30
BKC	95 (2)	99 (2)	98 (3)	100 (3)	99 (3)
DFRC	92 (0)	97 (0)	98 (0)	99 (0)	100 (0)
ItBC	93 (5)	99 (5)	98 (0)	99 (0)	100 (0)
MAC	94 (0)	99 (0)	98 (0)	99 (0)	100 (0)

valid groups in most cases. Also, different algorithms terminate with different local maxima solutions as described in Section 3.4. This also results in an interesting scenario where the local maxima corresponding at a lower value of K may be higher than that of the one corresponding to higher value of K, which explains BKC terminating with 99 coverage with K=30 whereas it terminated with 100 coverage with K=26, and similar cases.

Results on Larger datasets. The plots show the coverage values of different algorithms from the instant when the first set of K groups are reported by them. All algorithms are given a fixed time of one hour. Size of each node is around 80KB for 5,000 users (though the size of each node is proportional to the number of users, it will not be problematic as the number of nodes needed is very low as described previously in this subsection).

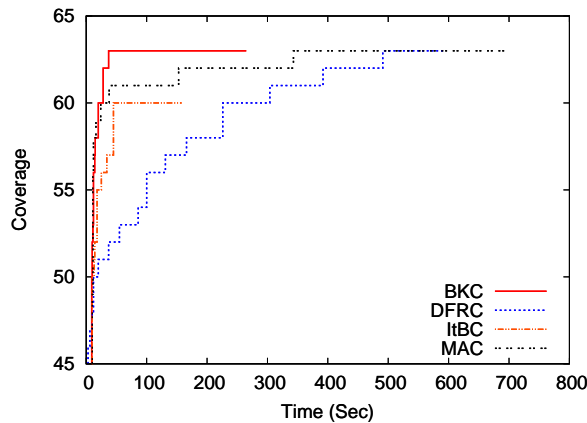


Figure 5: NUM_USERS : 1,000, max_dis : 10m, and K : 5 on Random Waypoint simulator data.

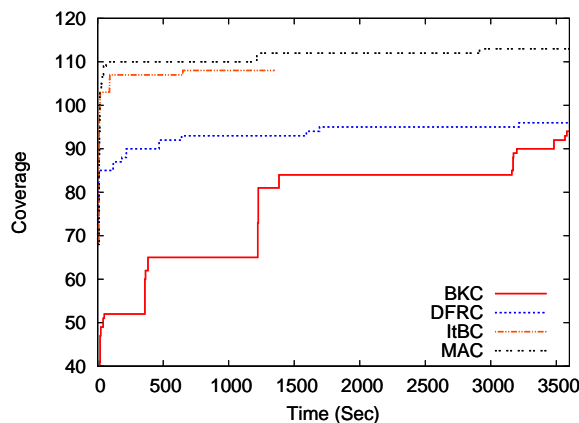


Figure 6: NUM_USERS : 1,000, max_dis : 20m, and K : 5 on Random Waypoint simulator data.

Figures 5 & 6 display the results on a dataset consisting of 1000 users when $K = 5$. max_dis is set to 10 in one set of experiments and 20 in another, resulting in different group size distributions, one with low average group size and the other with high average group size respectively. Note that, when the group

sizes are small (Figure 5) BKC is performing better, and when the group sizes are large (Figure 6) MAC is performing better.

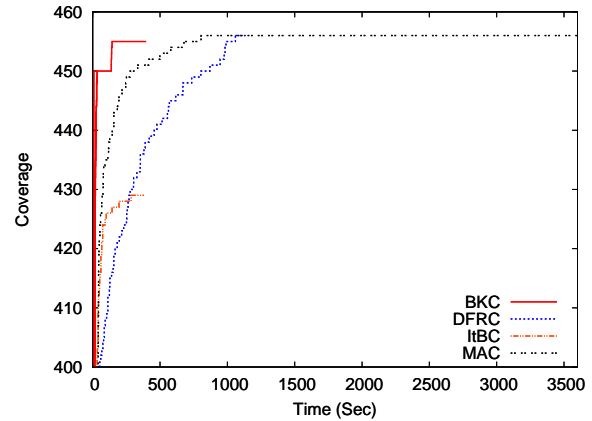


Figure 7: NUM_USERS : 1,000, max_dis : 10m, and K : 50 on Random Waypoint simulator data.

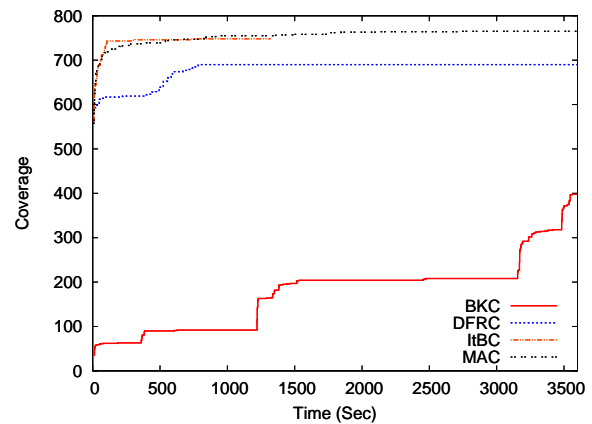


Figure 8: NUM_USERS : 1,000, max_dis : 20m, and K : 50 on Random Waypoint simulator data.

Figures 7 & 8 display the results on the same dataset consisting of 1000 users when $K = 50$. Once again, max_dis is set to 10 in one set of experiments and 20 in another, resulting in different group size distributions, on the lower side and on the higher side respectively. Note that, when the group sizes are small (Figure 7) BKC is performing better, and when the group sizes are large (Figure 8) MAC is performing better, similar to what is noted previously.

We now repeat similar set of experiments with larger number of users, 5000. Figures 9 and 10 display the results obtained for different values of max_dis . Note that, the relative performance of various algorithms is similar to what has been observed with the 1000 user dataset, depending only on the group size distribution.

We would like to mention that, when higher values of K are used such that all the users are covered, the algorithms terminated within very few minutes (details not given here), showing their efficacy. The hardness of the problem effects their performance most when the K value is such that a strict subset of users can only be covered which needs to be maximized, as one may rightly expect.

For even larger number of users, the algorithms slow down a bit but we have not noticed memory problems on our machine up-to 15000 users which is remarkable. However, for best performance, its advisable to devise efficient parallel or distributed methods for $NUM_USERS > 5,000$.

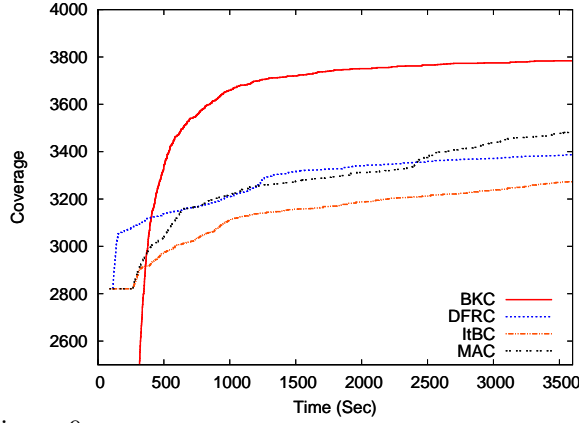


Figure 9: NUM_USERS : 5,000, max_dis : 10m, and K : 500 on Random Waypoint simulator data.

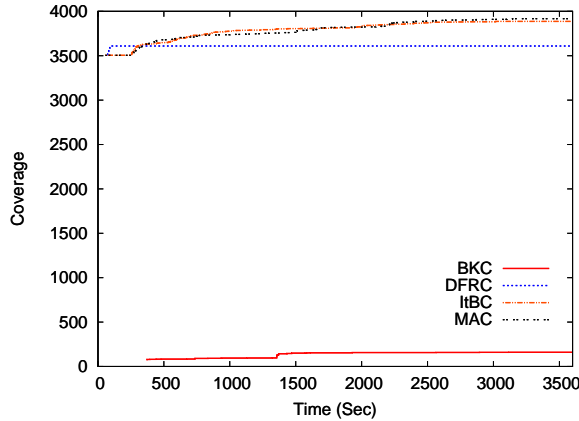


Figure 10: NUM_USERS : 5,000, max_dis : 20m, and K : 300 on Random Waypoint simulator data.

Experience with Clustering based Initialization. We have also experimented with clustering methods for obtaining a good quality K-set quickly which could be used for initialization of proposed algorithms. We have used a variation of K-means algorithm where an user is clustered with a node (set of users) with whom (any member of the set) its total length of *valid segments* is highest. Other nearness criteria, such as, cluster with most number of valid edges (weight $\geq min_wei$), highest sum of weights, highest sum of weights ratio (with cluster size), and number of valid edges ratio, were also tried. After obtaining clusters, we ran DFMax with ordering based on clusters, each time, nodes of a different cluster being put at the front. Highest weight (shared with any user of the cluster) criterion turned out to be the best one among clustering based methods. However, the basic DFRC produced a better starting point in lesser time. This is due to valid group definition being not so close to clustering mechanism which does not help produce a similar group as that of a cluster, making clustering ineffective.

4.2 With Oporto Simulator Data

Now, we present the results obtained using the Oporto simulator (Saglio & Moreira 2001). It tries to model the movement of fishing ships where the ships go in the direction of the most attractive shoals of fish while trying to avoid storm areas. Shoals are themselves attracted by plankton areas. Ships are moving points; plankton or storm areas are regions with fixed center but moving shape; and shoals are moving regions. We used the default parametric settings which come with the simulator provided by them online to

generate datasets consisting of 1000 shoals, and mined for K-sets with different max_dis values.

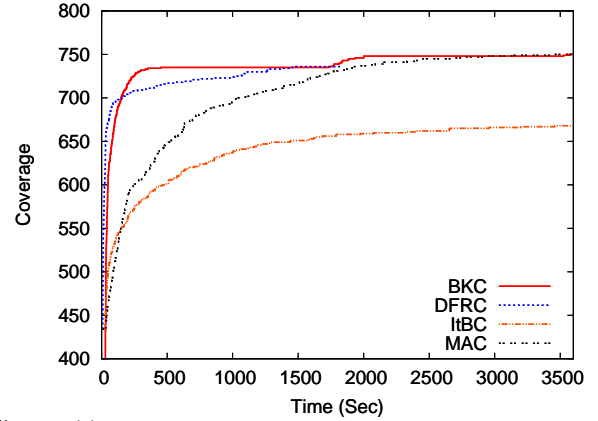


Figure 11: NUM_USERS : 1,000, max_dis : 30000, and K : 100 on Oporto Simulator Data.

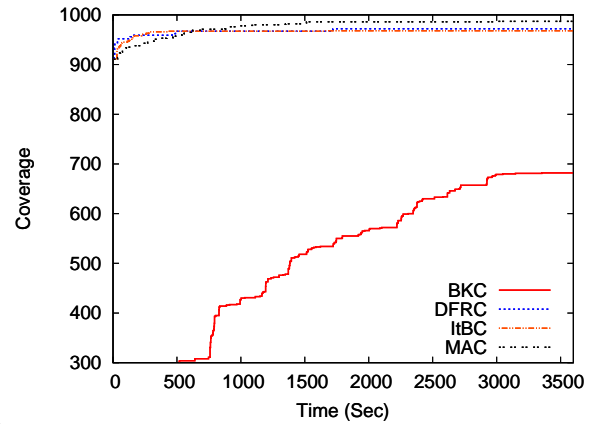


Figure 12: NUM_USERS : 1,000, max_dis : 60000, and K : 100 on Oporto Simulator Data.

Figures 11 and 12 show the results obtained on a 1000 user dataset when max_dis is 30000 and 60000 respectively ($NUM_TIMEPOINTS$: 3,001, min_dur : 3, min_wei : 1%). Note that, these two scenarios correspond to the group size being smaller in the first case and the group size being larger in the second case. And as expected, BKC performs better in the first case and MAC performs better in the second case.

4.3 With Brinkhoff Simulator Data

Here, we present the results obtained on the data generated using the network based mobility simulator of objects proposed by Brinkhoff (Brinkhoff 2002). Salient aspects of the simulator include, the maximum speed and the maximum capacity of connections, the influence of other moving objects on the speed and the route of an object, the influence of external events, time-scheduled traffic, etc. We have run their data generator which is provided online with maximum value for the number of object classes and got the data for 105 points (objects) moving inside a single time interval of 21 timepoints on the Oldenburg city map.

For our experiments, min_dur and min_wei are set to 3 timepoints. With max_dis = 3000, a total of 204 groups are mined by VGBK whose average group size is 4.166667 in 0 seconds. Their size distribution is given by: 1: 10, 2: 14, 3: 38, 4: 60, 5: 52, 6: 17, 7: 8, 8: 5, where $x : y$ denotes that there are y groups of size x . Table 2 shows the results obtained with this

max_dis value for different values of K. On such small datasets, all the algorithms are fairly competitive.

Table 2: Coverage attained by the proposed algorithms and the time taken in brackets (Seconds) for different values of K, when *max_dis* is 3000 on Brinkhoff Simulator Data.

Algo.	K				
	25	30	35	40	45
BKC	89 (0)	95 (0)	97 (0)	101 (1)	105 (0)
DFRC	87 (0)	93 (0)	98 (0)	103 (0)	105 (0)
ItBC	88 (0)	93 (0)	98 (0)	103 (0)	105 (0)
MAC	87 (0)	93 (0)	98 (0)	103 (0)	105 (0)

With *max_dis* = 6000, a total of 469 groups are mined by VGBK whose average group size is 10.650320 in 67 seconds. Their size distribution is given by: 1: 10, 3: 4, 4: 9, 5: 20, 6: 20, 7: 36, 8: 31, 9: 43, 10: 39, 11: 63, 12: 59, 13: 47, 14: 20, 15: 12, 16: 11, 17: 22, 18: 19, 19: 4, where $x : y$ denotes that there are y groups of size x . Table 3 shows the results obtained with this *max_dis* value for different values of K. Note that, the time taken by BKC is significantly larger than the other algorithms owing to the large sized groups present in the search space, which is consistent with our previous observation that the performance of BKC degrades with the presence of large sized groups (which it actually inherits from VGBK (Wang et al. 2008)).

Table 3: Coverage attained by the proposed algorithms and the time taken in brackets (Seconds) for different values of K, when *max_dis* is 6000 on Brinkhoff Simulator Data.

Algo.	K				
	10	15	20	25	30
BKC	86 (78)	94 (122)	98 (73)	103 (82)	105 (74)
DFRC	85 (5)	94 (0)	97 (0)	102 (0)	105 (0)
ItBC	82 (3)	93 (3)	97 (0)	102 (0)	105 (0)
MAC	85 (2)	94 (0)	97 (0)	102 (0)	105 (0)

4.4 With Geolife Trajectories Data

Finally, we present the results on a real dataset, albeit small. The GPS trajectory dataset was collected in (Microsoft Research Asia) Geolife project (Microsoft-ResearchAsia 2012) of 182 users in a period of over five years (from April 2007 to August 2012). For our experiments, we have processed this data to identify a period of 30 days in which most users were active. We found that in the peak month, the number of users active were 41. We processed the data of these 41 users for that month at a sampling rate of 100 timepoints per day (resulting in a total of 3000 timepoints).

Table 4: Coverage attained by the proposed algorithms and the time taken in brackets (Seconds) for different values of K on Geolife Trajectories data.

Algo.	K			
	10	15	20	25
BKC	31 (0)	37 (0)	40 (0)	41 (0)
DFRC	31 (0)	39 (0)	41 (0)	41 (0)
ItBC	32 (1)	39 (0)	41 (0)	41 (0)
MAC	32 (0)	39 (0)	41 (0)	41 (0)

We set the values of *min_dur* and *min_wei* to 2 timepoints (corresponds to approximately 30 Minutes time; using larger values such as *min_wei* = 1% resulted in just a single valid group of size 2) and *max_dis* to 500 (using smaller values reduced the number of groups drastically). VGBK reported a total of 154 maximal valid groups with average group size of 3.097403 in 0 seconds. The size distribution is as follows: 1: 1, 2: 44, 3: 61, 4: 37, 5: 9, 6: 2, where $x : y$ indicates that there are y groups of size x . Table 4 presents the results obtained upon running the proposed algorithms on this dataset with the above

parameters. We note that all the algorithms are fairly competitive on this small real dataset. Also, the coverage estimate given in Section 4.1 can be observed to be working well in this case. For example, for K = 10, K * average group size = 31 and so is the coverage attained.

Overall Observations: We observe the following points from the above sets of experiments (and based on related experiments carried out by us which are not shown due to space constraint):

1. BKC gives the best performance when the expected group size is small (≤ 10). For example, in cases where one is set out to mine groups of close friends, using a small *max_dis* value or large *min_wei* value. However, its performance degrades severely when the expected group size is large.
2. MAC and ItBC give the best performance when the expected group size is large (> 10). For example, in cases where one is set out to mine groups of people living/working in a nearby region, using a large *max_dis* value; or in densely populated regions.

We also measured the time taken for mining all maximal valid groups using VGBK (Wang et al. 2008) which ranged from 2 Minutes to 30 Minutes on a 1000 user dataset depending on the other input parameters, the time taken increasing further with the group size, no. of users (up-to hours and days). We have got 219057 maximal valid groups for the 1000 user Oporto dataset where the algorithms successfully found top K groups for coverage in quick time (showing almost no mining overhead before producing the first solution or in improving it).

5 Conclusion

In this paper, we introduced the problem of finding K groups with maximum coverage in the context of spatio-temporal data mining. We have proposed several efficient methods to solve this hard problem, based on existing mining techniques, depth-first search and heuristic-search techniques. The methods can work within the given amount of memory, produce solutions in anytime manner and guarantee terminating with maximal solutions. This paper also demonstrates a new way of applying best-first search methods to solve newer and more difficult problems. Experimental results show that different methods are effective under different conditions which are clearly categorized for selection as per user application. The methods can also be used when mining groups with other group definitions involving other types of data. The problem also opens up several interesting future directions, including better algorithms, parallel adaptations, and extending to other domains such as frequent itemsets.

Acknowledgment

This work was carried out as part of the Xerox India Innovation university research projects at Indian Institute of Technology Kharagpur. We thank Dr. Nathan Gnanasambandam of Xerox for his valuable comments and suggestions.

References

- Afrati, F., Gionis, A. & Mannila, H. (2004), Approximating a collection of frequent sets, in 'Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery

- and data mining', KDD '04, ACM, New York, NY, USA, pp. 12–19.
- Agarwal, R. C., Aggarwal, C. C. & Prasad, V. V. V. (2000), Depth first generation of long patterns, in 'Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining', KDD '00, ACM, New York, NY, USA, pp. 108–118.
- Agrawal, R. & Srikant, R. (1994), Fast algorithms for mining association rules in large databases, in 'Proceedings of the 20th International Conference on Very Large Data Bases', VLDB '94, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 487–499.
- Aine, S., Chakrabarti, P. P. & Kumar, R. (2007), AWA* - A window constrained anytime heuristic search algorithm, in 'IJCAI', pp. 2250–2255.
- Bisiani, R. (1987), 'Beam search', *Encyclopedia of Artificial Intelligence* pp. 56–58.
- Brinkhoff, T. (2002), 'A framework for generating network-based moving objects', *Geoinformatica* 6(2), 153–180.
- Broch, J., Maltz, D. A., Johnson, D. B., Hu, Y.-C. & Jetcheva, J. (1998), A performance comparison of multi-hop wireless ad hoc network routing protocols, in 'Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking', MobiCom '98, ACM, New York, NY, USA, pp. 85–97.
- Bron, C. & Kerbosch, J. (1973), 'Algorithm 457: Finding all cliques of an undirected graph', *Commun. ACM* 16, 575–577.
- Calders, T. & Goethals, B. (2002), Mining all non-derivable frequent itemsets, in 'Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery', PKDD '02, Springer-Verlag, London, UK, UK, pp. 74–85.
- Chakrabarti, P. P., Ghose, S., Acharya, A. & Sarkar, S. C. D. (1989), 'Heuristic search in restricted memory.', *Artificial Intelligence* 41(2), 197–221.
- Davis, R., Hagey, W. & Horning, M. (2004), 'Monitoring the behavior and multi-dimensional movements of weddell seals using an animal-borne video and data recorder', *Memoirs of the National Institute of Polar Research (Japan) Special Issue* 58, 148–154.
- Dean, T. & Boddy, M. S. (1988), An analysis of time-dependent planning, in 'AAAI', pp. 49–54.
- Forsyth, D. R. (2006), *Group dynamics*, 4 edn, Thomson/Wadsworth, Belmont, CA.
- Gouda, K. & Zaki, M. J. (2005), 'Genmax: An efficient algorithm for mining maximal frequent itemsets', *Data Min. Knowl. Discov.* 11, 223–242.
- Han, J., Wang, J., Lu, Y. & Tzvetkov, P. (2002), Mining top-k frequent closed patterns without minimum support, in 'Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on', pp. 211–218.
- Hart, P. E., Nilsson, N. J. & Raphael, B. (1968), 'A formal basis for the heuristic determination of minimum cost paths', *IEEE Transactions on Systems Science and Cybernetics* 4(2), 100–107.
- Hochbaum, D. S., ed. (1997), *Approximation algorithms for NP-hard problems*, PWS Publishing Co., Boston, MA, USA.
- Kalnis, P., Mamoulis, N. & Bakiras, S. (2005), On discovering moving clusters in spatio-temporal data, in 'In SSTD', Springer, pp. 364–381.
- Lauw, H. W., Lim, E.-P., Pang, H. & Tan, T.-T. (2005), 'Social network discovery by mining spatio-temporal events', *Comput. Math. Organ. Theory* 11, 97–118.
- Lee, A. J. T., Chen, Y.-A. & Ip, W.-C. (2009), 'Mining frequent trajectory patterns in spatial-temporal databases', *Inf. Sci.* 179, 2218–2231.
- MicrosoftResearchAsia (2012), 'Geolife gps trajectories', <http://research.microsoft.com/en-us/projects/geolife/>.
- Ng, R. T. & Han, J. (1994), Efficient and effective clustering methods for spatial data mining, in 'Proceedings of the 20th International Conference on Very Large Data Bases', VLDB '94, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 144–155.
- Pei, J., Dong, G., Zou, W. & Han, J. (2002), On computing condensed frequent pattern bases, in 'Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on', pp. 378–385.
- Pei, J., Han, J. & Mao, R. (2000), Closet: An efficient algorithm for mining frequent closed itemsets, in 'ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery', pp. 21–30.
- Rymon, R. (1992), *Search through Systematic Set Enumeration*, Morgan Kaufmann, pp. 539–550.
- Saglio, J.-M. & Moreira, J. (2001), 'Oporto: A realistic scenario generator for moving objects', *Geoinformatica* 5(1), 71–93.
- Schafer, J. B., Konstan, J. A. & Riedl, J. (2001), 'E-commerce recommendation applications', *Data Min. Knowl. Discov.* 5, 115–153.
- Srikant, R., Vu, Q. & Agrawal, R. (1997), Mining association rules with item constraints, in 'KDD', pp. 67–73.
- Thayer, J. T. & Ruml, W. (2010), Anytime heuristic search: Frameworks and algorithms, in 'SOCS'.
- Tseng, V. S., Wu, C.-W., Shie, B.-E. & Yu, P. S. (2010), Up-growth: an efficient algorithm for high utility itemset mining, in 'Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining', KDD '10, ACM, New York, NY, USA, pp. 253–262.
- Vadlamudi, S. G., Aine, S. & Chakrabarti, P. P. (2011), 'MAWA*—A memory-bounded anytime heuristic-search algorithm', *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 41(3), 725–735.
- Vadlamudi, S. G., Gaurav, P., Aine, S. & Chakrabarti, P. P. (2012), Anytime column search, in 'Australasian Conference on Artificial Intelligence', pp. 254–265.
- van den Berg, J., Shah, R., Huang, A. & Goldberg, K. Y. (2011), Anytime nonparametric A*, in 'AAAI'.
- Verhein, F. & Chawla, S. (2006), Mining spatio-temporal association rules, sources, sinks, stationary regions and thoroughfares in object mobility databases, in 'of Lecture Notes in Computer Science', Springer, pp. 187–201.
- Wang, K., He, Y. & Han, J. (2000), Mining frequent itemsets using support constraints, in 'Proceedings of the 26th International Conference on Very Large Data Bases', VLDB '00, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 43–52.
- Wang, Y., Lim, E.-P. & Hwang, S.-Y. (2008), 'Efficient algorithms for mining maximal valid groups', *The VLDB Journal* 17, 515–535.
- Wang, Y., peng Lim, E. & yih Hwang, S. (2003), On mining group patterns of mobile users, in 'Proceedings of the 14th International Conference on Database and Expert Systems Applications—DEXA 2003', pp. 287–296.
- Xin, D., Cheng, H., Yan, X. & Han, J. (2006), Extracting redundancy-aware top-k patterns, in 'Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining', KDD '06, ACM, New York, NY, USA, pp. 444–453.
- Xu, J. J. & Chen, H. (2005), 'Crimenet explorer: a framework for criminal network knowledge discovery', *ACM Trans. Inf. Syst.* 23, 201–226.
- Zhou, R. & Hansen, E. A. (2005), Beam-stack search: Integrating backtracking with beam search., in 'Proceedings of the 15th International Conference on Automated Planning and Scheduling (ICAPS-05)', Monterey, CA, pp. 90–98.

Mining Cluster-based Patterns for Elder Self-care Behavior

Yu-Shiang Hung¹ Kuei-Ling B. Chen² Chi-Ta Yang³ Guang-Feng Deng⁴

Innovative DigiTech-Enabled Applications & Services Institute

Institute for Information Industry

8F., No.133, Minsheng E. Rd., Taipei City 105, Taiwan

garhung@iii.org.tw¹ klchen@iii.org.tw² gary1122@iii.org.tw³ raymalddeng@iii.org.tw⁴

Abstract

The rapid growth of the elderly population has increased the need to support elders in maintaining independent and healthy lifestyles in their homes rather than through more expensive and isolated care facilities. Self-care can improve the competence of elderly participants in managing their own health conditions without leaving home. This main purpose of this study is to understand the self-care behavior of elderly participants in a developed self-care service system that provides self-care service and to analyze the daily self-care activities and health status of elders who live at home alone.

To understand elder self-care patterns, log data from actual cases of elder self-care service were collected and analysed by Web usage mining. This study analysed 3391 sessions of 157 elders for the month of March, 2012. First, self-care use cycle, time, function numbers, and the depth and extent (range) of services were statistically analysed. Association rules were then used for data mining to find relationship between these functions of self-care behavior. Second, data from interest-based representation schemes were used to construct elder sessions. The *ART2*-enhance *K*-mean algorithm was then used to mine cluster patterns. The analysis results can be used for research in medicine, public health, nursing and psychology and for policy-making in the health care domain.

Keywords: Elder self-care behavior pattern; web usage mining; cluster analysis; sequential profiles; Markov model; association analysis.

1 Introduction

The global population is aging rapidly and is expected to require expanded health care services and facilities. However, since many elders prefer to stay in their private residences for as long as possible, methods are needed to enable them to do so safely and at reasonable costs. Several studies have investigated how information technologies can assist elders in independent living and in daily activity (Barger, Brown et al. 2005; Chung-Chih, Ming-Jang et al. 2006; Honda, Fukui et al. 2007; Leijdekkers, Gay et al. 2007; Seon-Woo, Yong-Joong et al. 2007; Wang, Wang et al. 2007; Dobrescu, Dobrescu et al. 2009)

A major goal of elderly care is facilitating the ability of elders to maintain and promote their own health. Although they may suffer from chronic disease, cognitive impairment and functional limitation, mobilization of their self-care resources can minimize their health problems and enhance their health and well-being (Hoy, Wagner et al. 2007). Even when they have chronic disease and disability, elders often consider themselves active and in good health, and they are usually highly motivated to learn about ageing and health conditions.

Self-care can improve the competence of elders in managing their own health conditions without institutional care (Hoy, Wagner et al. 2007). However, a clear understanding of self-care activities in this group is essential (Arcury, Grzywacz et al. 2012). In 2011, the ComCare elder self-care project in Taiwan established an integrated IT platform that enables elders living at home to use tablet computers for major physical and mental health self-care functions. The ComCare server system has a database server and a web server for collecting useful information about health status and daily activities. The system enables elders to enter the information by using tablet PCs. Web Usage Mining is an area of Web Mining that deals with extracting interesting and useful knowledge from logging information produced by Web servers (Facca and Lanzi 2005; Sajid, Zafar et al. 2010; Wang and Lee 2011). Many researchers have applied Web usage mining for characterizing usage based on navigation patterns (Chen, Bhowmick et al. 2009; Bayir, Toroslu et al. 2012), for behavior prediction (Dimopoulos, Makris et al. 2010), for personalized recommendation (Mobasher, Cooley et al. 2000; Pierrakos, Paliouras et al. 2003; Park, Kim et al. 2012) and for web service improvement (Carmona, Ramírez-Gallego et al. 2012).

The main purpose of this study is to apply data mining techniques, including statistical analysis, clustering, association rules and sequential pattern discovery, for mining Web usage information from ComCare server logs to understand elder self-care behavior patterns. First, a statistical analysis is performed to analyze self-care use cycle, time, function numbers, and association rules to determine the depth and extent (range) of ComCare use. Second, interest-based representation schemes are used to construct elder sessions and then combined with an *ART2*-enhance *K*-mean algorithm to mine cluster patterns. To capture sequence information for self-care behavior patterns for elders using ComCare, sequence-based representation schemes in association with Markov models are combined with an *ART2*-enhance *K*-mean algorithm for mining cluster patterns in elder behavior. The analysis results can be used for research and policy-making by health care experts in medicine, public health, nursing and psychology. In practice, the improved

characterization of elder self-care based on the analytical results in this study can improve personalized service and the design of elder self-care services.

This study is organized as follows Section 2 presents the functions of ComCare. Section 3 describes the Web usage mining technique used to analyse elder self-care in this study, including interest-based representation schemes, and ART2-enhance *K*-mean algorithm. Section 4 presents the experimental results. Section 5 presents conclusions and proposes future works.

2 ComCare Functions Description

The global population is aging rapidly. Taiwan was classified as an aging society in 1993, and will be classified as an aged society by 2018. The major concern is the speed of aging. According to Ministry of Interior data, 10.63% of the total Taiwan population were older than sixty-five years in 2010, and the aging index was 65.05%. The population is aging faster than any country in the world due to the low birth rate, which was only 0.9 in 2010.

ComCare elder self-care service was provided by Institute for Information Industry Innovative DigiTech-Enabled Applications & Services Institute (IDEAS) in Taiwan. The service requires an in-house tablet PC device and a server system. The server system includes a database server and a web server, which collect useful behavior data log from the tablet PC device. ComCare currently offers 14 self-care main-functions to address the everyday needs of seniors (Fig. 1); Of these, five are health management-related functions: ‘Diet Management’, ‘Exercise Management’, ‘Blood pressure Management’, ‘BMI Management’, and ‘Statistical data Management’. The three social community behavior-related main-functions are ‘Friend Management’, ‘Video Interaction’ and ‘Photo sharing’ in interactive care topic. The five life-information-related main-functions are ‘On sale’, ‘Calendar management’, ‘Weather forecast’, ‘Resident information’ and ‘Community management’ in life-information topic. ‘Entertainment management’ is the only entertainment-related main-function in the entertainment category. Additionally, each main function has various sub-functions; ComCare provides 71 such sub-functions for elder self-care service.



Figure 1: Main functions in elder care services

3. Methodology

This section first describes the data log preprocessing procedure and then presents two representation schemes suitable for capturing elder self-care behavior, including interest-based and sequence-based representation in Web

usage mining. Finally, the ART2 neural network and *K*-mean clustering algorithm used in this study are introduced.

3.1 Web-log Preprocessing

Data log preprocessing transforms the original logs so that all web access sessions can be identified. The Web server usually registers the access activities of website users in Web server logs. Different server parameters settings result in many different web log types, but log files typically share the same basic information, including client IP address, request time, requested *URL*, *HTTP* status code, referrer, *etc.* Generally, several preprocessing tasks are required before performing web usage mining algorithms on the Web server logs. The tasks in this work include data cleaning, user differentiation and session identification.

These preprocessing tasks resemble those in any other web usage mining problem and are discussed in detail in Hussain *et al.* (2010). (Hussain, Asghar *et al.* 2010) The original server logs are cleansed, formatted, and then grouped into meaningful sessions before use in web usage mining. A session can be described as the self-care activities performed by an elder between the start and the end of the ComCare session. Therefore, session identification is the process of segmenting the access log of each elder into individual access sessions. The activity stay time-based method developed by Liu and Kešelj (2007) was applied for session identification in this study (Liu and Kešelj 2007). This method limits the time spent on a function of ComCare to a specified threshold. If the time between the request most recently assigned to a session and the next request from the elder exceeds the threshold, a new access session is assumed. A 10 min activity-stay time is considered a conservative threshold for capturing the time for loading and studying page content (Liu and Kešelj 2007). However, this study uses the average use time of the function of all sessions as a threshold for function use time.

3.2 Interest-based Representation

‘Frequency’ and ‘Duration’ information for elder self-care behavior is captured by using an interest-based representation similar to that described in Lin, Liu and Kešelj (2007) (Liu and Kešelj 2007). Two indicators, ‘Frequency’ and ‘Duration’, are used to represent the interest sessions of elders. Let F be a set of sub-functions used by elders in ComCare server logs, $F = \{f_1, f_2, \dots, f_m\}$, each of which is uniquely represented by its associated sub-function ID. Let S be a set of user access sessions. Hence, $S = \{s_1, s_2, \dots, s_n\}$, where each $s_i \in S$ is a subset of F . To facilitate the clustering operation, each session s is represented as an m -dimensional vector over the space of sub-functions, $s = \{(int_1, s), (int_2, s), \dots, (int_m, s)\}$, where (int_i, s) is an interest assigned to the i th sub-function ($1 \leq i \leq m$) used in a session s . All sub-functions are assumed to be equally important to elder self-care pattern profiles. Therefore, regardless of the self-care sequence, the focus is the specific sub-functions used in a session.

The interest (int_i, s) must be determined appropriately to capture the interest of an elder in a sub-function. Two underlying concepts of this measure are ‘Frequency’ and

‘Duration’. ‘Frequency’ is the number of uses of a sub-function. A high frequency for a sub-function is assumed to indicate a strong need or interest of the elders. Equation (1) is the formula for ‘Frequency’, which is normalized by the total number of uses of sub-functions in the session:

$$\text{Frequency}(f_i) = \frac{\text{NumberOfUses}(f_i)}{\sum_{f_j \in \text{UsedFunctions}} \text{NumberOfUses}(f_j)} \quad (1)$$

‘Duration’ is defined as the time spent on a sub-function, *i.e.*, the difference between the requested time of two adjacent entries in a session. We conjecture that, as the time spent on a sub-function increases, the likelihood of the elder becoming interested in the sub-function increases. Typically, an elder quickly jumps to another sub-function if a sub-function is not useful. However, a quick jump might also result from the short operating length of a sub-function. Hence, normalizing ‘Duration’ by the operating length of the sub-function, that is, by the basic operating time of the sub-function, is more appropriate. Equation (2) is used to measure the ‘Duration’ of a sub-function,

$$\text{Duration}(f_i) = \frac{\text{TotalDuration}(f_i) / \text{Length}(f_i)}{\max_{f_j \in \text{UsedFunctions}} \text{TotalDuration}(f_j) / \text{Length}(f_j)} \quad (2)$$

where ‘Duration’ of a sub-function is further normalized by the max ‘Duration’ of sub-functions in the session. For the last access sub-function in each user access session, the duration cannot be estimated by calculating the difference in requested time. The average duration of the relevant session is used as the estimated duration for the last access event.

In this study, ‘Frequency’ and ‘Duration’ are considered two strong indicators of interest of elders. Therefore, ‘Frequency’ and ‘Duration’ are valued equally in the proposed interest measure. The following equation shows how the harmonic means for ‘Frequency’ and ‘Duration’ are used to represent the interest of an elder in a sub-function during a session:

$$\text{Interest}(f_i) = \frac{2 \times \text{Frequency}(f_i) \times \text{Duration}(f_i)}{\text{Frequency}(f_i) + \text{Duration}(f_i)} \quad (3)$$

Equation (3) ensures that ‘Interest’ for a sub-function is high only when both ‘Frequency’ and ‘Duration’ are high. Meanwhile, ‘Interest’ is normalized to a value between 0 and 1 which is convenient not only for understanding, but also for session clustering.

Each elder access session is eventually transformed into an m -dimensional vector of interests of sub-functions, *i.e.*, $s = \{\text{int}_1, \text{int}_2, \dots, \text{int}_m\}$, where m is the number of sub-functions used in all user access sessions. However, if the number of dimensions m exceeds a reasonable size, it not only consumes substantial processing time during clustering sessions, but also limits the real-world applicability of the system. Dimensions are reduced by using a frequency threshold f_{\min} as a constraint to filter out sub-functions that are accessed less than f_{\min} times in all access sessions. Our research showed that 60% of sub-functions appearing in the access sessions were visited at least 50 times. These sub-functions were considered representative functions that drew the attention of elders. Therefore, f_{\min} was set to 50.

3.3 Clustering Analysis

In Web usage mining, clustering finds groups that share common properties and behavior by analyzing the data collected in web servers. Given the transformation of elder self-care access sessions into a multi-dimensional space as interest-based representation vectors or sequence-based representation matrices of functions, a clustering algorithm was applied to the derived elder self-care access sessions. Since access sessions are the images of activities by elders, representative elder self-care patterns can be obtained by clustering. These patterns also facilitate profiling of elder users of the ComCare service. This section describes how session clustering is performed and how cluster number is determined.

3.3.1 Optimizing the Number of Clusters

Since the used clustering algorithm is a supervised clustering method, an ART2 neural network (Kuo and Lin 2010) is needed to determine the number of clusters. The ART2 neural network architecture is designed for processing both analog and binary input patterns. An ART2 neural network consists of F_1 and F_2 layers. The F_1 layer has seven nodes (W, X, U, V, P, Q). The input signal is processed by the F_1 layer and then passed from the bottom to the top value (b_{ij}). The result of the bottom-to-top value is an input signal of the F_2 layer. The nodes of the F_2 layer compete with each other to produce a winning unit, which returns the signal to the F_1 layer. The match value is then calculated with the top-to-bottom value (t_{ji}) in the F_1 layer and compared with the vigilance value. If the match value exceeds the vigilance value, then the weights of b_{ij} and t_{ji} are updated. Otherwise, the reset signal is sent to the F_2 layer, and the winning unit is inhibited. After inhibition, the other winning unit is found in the F_2 layer. If all F_2 layer nodes are inhibited, the F_2 layer produces a new node and generates the initial weights corresponding to the new node.

3.3.2 Session clustering

After the ART2 neural network determines the number of clusters, standard clustering algorithms can partition this space into groups of sessions that are close to each other based on a distance measure. The well-known K -means algorithm is used as the base method for clustering interest-based representation sessions and sequence-based representation sessions. The K -means clustering algorithm groups sessions by attributes/features into a k (positive integer) number of groups by minimizing the sum of squares of distances between data and the corresponding cluster centroid. Additionally, the most popular Euclidean distance is used as the distance measure. The K -means clustering algorithm is performed in the following steps: Step 1: Generate initial random cluster centroids for k clusters and k obtained by ART2 neural network. Step 2: Assign each session to its closest cluster centroid in terms of Euclidean distance. Step 3: Compute new cluster centroids. Step 4: If cluster memberships differ from the last iteration, repeat steps 2–3. Step 5: Stop and store clustering result.

Session clustering obtains a set of clusters, $C = \{c_1, c_2, \dots, c_k\}$ in which each c_i ($1 \leq i \leq k$) is a subset of the set of elder access sessions S where k is the number of clusters. A mean vector m_c of a sequence-based

representation (a mean matrix m_c of an interest-based representation) is computed as a representation for each session cluster $c \in C$. Each mean vector represents the representative user self-care pattern for a cluster in which a particular set of functions are accessed. The mean value for each function in the mean vector is computed as the average weight of the functions across total access sessions in the cluster. Therefore, the mean value is also between 0 and 1. Meanwhile, a weight threshold for the mean vector of each session cluster, w_{min} , is set as a constraint to filter out functions with mean values below the threshold for the cluster. The remaining self-care functions in each cluster are considered of greatest interest to elders and are used as representative self-care patterns for the cluster. Since the least mean value is always far smaller than the second least and the third least mean values, the second least mean value of each mean vector is used as the w_{min} for each session cluster.

In our research elder self-care patterns are described in terms of the common usage characteristics for a group of elders. Since many elders may have common interests up to a point during their self-care navigation, navigation patterns should capture the overlapping interests or the information needs of these users. In addition, self-care patterns should also be capable to distinguish among functions based on their different significance to each pattern. This work defines an elder self-care pattern np as a pattern that captures an aggregate view of the behavior of a group of elders based on their common interests or sequence information. After session clustering, $NP = \{np_1, np_2, \dots, np_k\}$ represents the set of elder self-care patterns, in which each np_i is a subset of F , the set of functions.

4. Elder Self-care Behavioral Analysis

4.1 Dataset and Environment

The ComCare data server was used to obtain a record of each function used by an elder from the start of service until the present. For all of March, 2012, 3391 sessions of 157 elders were identified for analysis. Matlab Language is used to perform the web usage mining algorithm. In addition to collecting the record data, the researchers invited ComCare users to complete a questionnaire survey, which recovered 157 valid questionnaires. The content of the structured questionnaire, which was compiled by a team of experts, includes user impressions of ComCare service (including information quality, service quality, ComCare self-efficacy, and perceived risk. To prevent users from giving noncommittal responses and to ensure easily measurable user responses, a six-item Likert scale with six choices for each dimension was used; the higher the score for each item, the greater the satisfaction of the respondent with that service. Analysis focused on the record data and questionnaire data for the 157 test elders who completed the questionnaire.

The questionnaire results indicated that the age range of the test users was 58 to 86 years, and the average age was 68 years (standard deviation, 8.22). Figure 2 presents basic information concerning ComCare users. Of the 157 test users, 56% were women and 44% were men. Education levels were relatively high; most had at least a university education. At least 75% of the respondents regularly used

the Internet, which was higher than the finding of another survey of internet use among the older generation, which showed that 56.3% of senior respondents had internet experience. Approximately 68% of users were retired or not currently employed, and close to 40% of users had a history of chronic disease such as hypertension or diabetes.

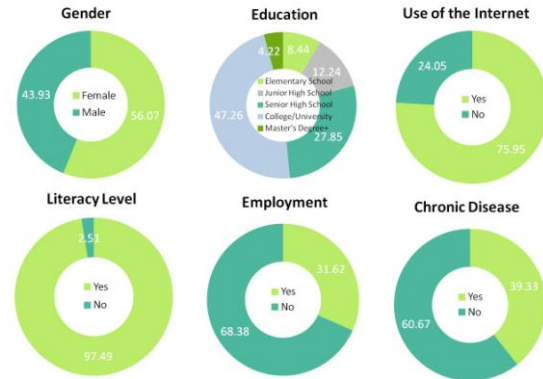


Figure 2: Basic background information of self-care elders

4.2 Elder self-care Profiling Results

Two factors were considered when profiling the self-care behavior of elders: use function length in terms of the number of sub-functions used by elders in ComCare and use duration in terms of the number of hours spent by elders in ComCare. Figure 3 shows the distribution of use function length for elder self-care. The mean number of elder requests per session is almost 4.5, which is lower than the expected number of the ComCare provider.

Comparison with other statistics indicated that the distribution of use functions is strongly right-skewed. For example, the mean (4.5) is higher than the median (3), the mode (1) is the same as the minimum, and the maximum (24) is very large. Figure 3 confirms our deduction that the distribution is right-skewed (to increase granularity, use functions above 20 are omitted from the graph. Inclusion of these records would have made the graph even more right-skewed). The data shows that the great majority of sessions included fewer than five function requests, half included three or fewer, and a disappointingly large number of sessions included only a single function request. This finding should be noted by service developers because it raises the question of why elders are leaving so soon. Notably, however, the third and fourth most common actions included five and eight actions, respectively, which is higher than the expected number.

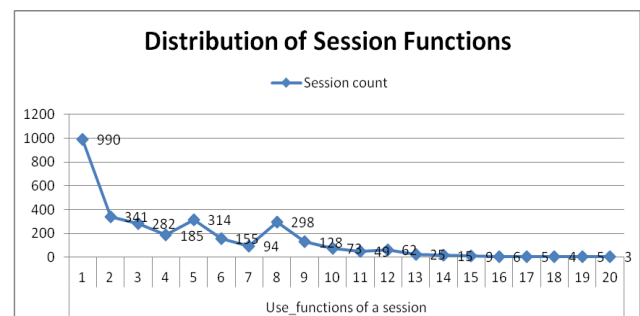


Figure 3: Distribution of session functions

Apart from the number of function requests per session, another important variable is the duration of time per session that the elders spent on the ComCare self-care service. The mean session duration is 554.8 seconds (9.23 minutes). Again, the outcome is lower than the expected number of the ComCare provider. However, the median for right-skewed data (Fig. 4) is a better summary statistic than the mean. The median session duration is 318.5 seconds (about 5.28 minutes), which is a more realistic estimate of the typical duration of a session for those who requested more than one function. Figure 4 shows the distribution of session duration (also known as self-care use time) for multi-function sessions. Again, the upper tail is clipped at 2400 seconds to increase the granularity of the graph. The figure shows that most sessions lasted less than 10 minutes, half lasted 6 minutes or fewer, and a disappointingly large number of sessions lasted only used 3 minutes. Again, service developers should be concerned.

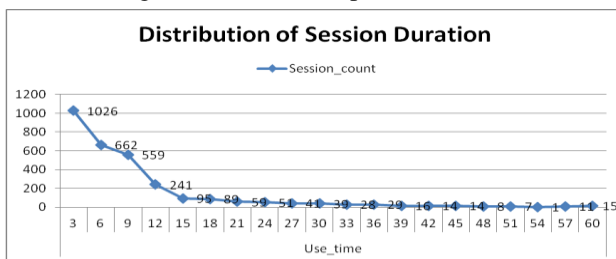


Figure 4: Distribution of session duration

Hierarchical analysis was also performed to understand the depth and extent of use of ComCare. Figure 5 shows the hierarchical architecture of ComCare, which shows that the health management-related sub-functions were the most frequently used activities and that each sub-function of these health topics had a strong association.

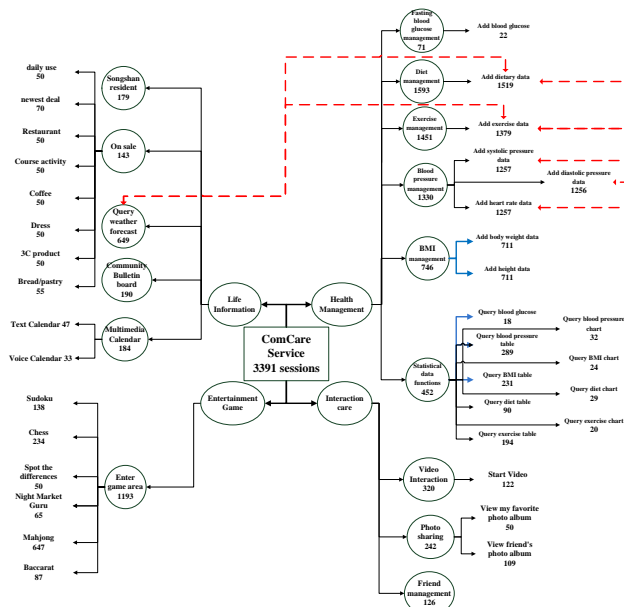


Figure 5: Depth and extent of self-care service use

Among the four topics, the use depth of interaction care and the life information topic revealed low use level. Only about 5% of the total usage session was spent on the sub-function of the Life information topic, excluding Query weather forecast, and only about 30% of usage session was spent on the main-function of the Life information topic keep going to request the

main-function's sub-functions of Life information. The interaction care topic reveals the same condition. The main-function of the interaction care topic was visited in only about 8% of the total usage sessions, and, of these, the sub-function was again used in only about 30%. The results indicate that self-care service providers need further information to improve the low rate of self-care functions.

4.3 Association Analysis of ComCare Service

This paper applied an *a priori* algorithm to the self-care service log data to reveal actionable association rules. The condition support > 5% and confidence > 90% revealed 63 association rules. For brevity, Table 1 lists only three interesting association rules revealed by the *a priori* algorithm. The rules are sorted by "rule support". Consider the first rule in the list. The antecedent is Exercise management function, and the consequent is Diet management function. The form of the rule is therefore Exercise management (1376) \Rightarrow Diet management (1265) conf:(0.92). The support of the rule is 37.46%, meaning that the rule applies to almost 37% of the 3391 total sessions in the data, which is a very high support level. A 92% confidence indicated that, of these 3391 sessions, 1376 met the antecedent condition; that is, 1376 requested Exercise management at some point. Of these 1376 sessions, the Diet management function was also requested in 92% (1265 sessions). According to rule 1, elders who pay attention to the Exercise management also emphasize Diet management in self-care behavior. According to rule 2, elders who pay attention to the BMI management also emphasize Blood management in self-care behavior. Rule 3 is that elders who pay attention to the BMI management & Exercise management also emphasize Diet management & Blood management. Surprisingly, in the support > 5% and confidence > 90% condition, the analysis showed that, of the functions in the four different topic, only the Query weather forecast function in the life-information-related topic was associated with Diet management and Exercise management in the health management-related topic.

Table 1: Association analysis of self-care behavior

Rule ID	Antecedent(account) \Rightarrow Consequent(account)	(Sup,Conf)
1	Exercise management (1376) \Rightarrow Diet management (1265)	(0.37,0.92)
2	BMI management (708) \Rightarrow Blood management (593)	(0.18,0.84)
3	BMI management & Exercise management (659) \Rightarrow Diet management & Blood management (556)	(0.16,0.84)

4.4 Interest-based Cluster Patterns

This section describes the elder self-care pattern obtained in three clusters by applying interest-based representation schemes and ART2-enhance K-mean algorithm. Table 2 characterizes each cluster pattern. Based on the observed means and proportions in Table 2, cluster C1 can be labelled as "Health dominate elders". The BMI, Diet, Exercise, and Blood pressure management functions were

requested at a much higher rate and for a longer duration by Health elders compared to elders in other clusters. Cluster C2 can be labelled as “Entertainment dominate elders”. The Mahjong game functions are requested by these elders at a rate thousands of times higher compared to other typical elders. The same was true for the chess game. Cluster C3 can be labelled “General elders”. The Diet and Exercise management functions were requested at a much higher rate and for a longer duration compared to the other functions. Cluster C3, which is the largest cluster, contains 59 elders; cluster C1 contains 52 elders; cluster C2 contains 46 elders. These data do not enable provisional identification of the “typical” cluster because the three clusters comprise 30%~40% of the sessions.

Health-dominant elders use more self-care functions (8.46) compared to Entertainment dominate elders (2.74) and General dominate elders (4.1) although the session duration of Health dominate elders is the shortest among the three cluster types since the average time per function is a fraction of that of the Entertainment dominate elders. The use cycle of General dominate elders is longest among the three cluster types. The Health dominate elders, however, very rarely access game functions. For all cluster types, Table 2 also shows no interest in functions of Friend management, Photo sharing, Multimedia calendar, Video Interaction and Enter community bulletin board functions. This raises the questions of whether these self-care functions are sufficiently user friendly and how the functions can be changed to induce elders to linger and use more functions. Thus, the interest-based representation scheme reveals interesting differences between the three elder self-care patterns. Perhaps the elder self-care service masters could apply this knowledge to differentiate service for each of the three elder types.

Table 2: Elder self-care characterization of interest-based cluster

Function	C1 Health-dom inate elders	C2 Entertainme nt-dominate elders	C3 General elders
People Account	52	46	59
percentage	33.12%	29.30%	37.58%
Cycle (day)	1.511	1.444	1.845
session duration	382.636	657.045	409.553
Function number	8.467	2.742	4.07
Friend management	0.055	0.048	0.022
BMI management	0.851*	0.026	0.06
Diet management	0.865*	0.164	0.878*
Exercise management	0.878*	0.122	0.758*
Photo sharing	0.034	0.04	0.037
Calendar Management	0.04	0.049	0.051
Video Interaction	0.115	0.103	0.101
Blood management	0.785*	0.291	0.447

Weather forecast	0.319	0.262	0.079
Enter community bulletin board	0.079	0.062	0.037
Game - Chess	0.014	0.111*	0.011
Game - Mahjong	0.074	0.354*	0.106

*: High-interest function

5 Conclusion and Future Work

To improve understanding of the self-care behavior of elders living alone, this study analysed real-world data for elder self-care service by applying Web usage mining methodology, including association analysis, and interest- and sequence-based representation schemes in association with Markov models combined with ART2-enhance K-mean algorithm.

The analysis results show that elder self-care behavior can be classified by an interest-based representation scheme into three distinct cluster types: health-dominate type, entertainment-dominate type and general-dominate type. Each type displays different self-care use cycles, times, function numbers and needs. However, six distinct cluster types can be identified by sequence-based clustering from different use sequence. Each type provides detailed information about self-care use cycle, time, function numbers and characterizations. This research shows that the use of sequence-based clustering in web usage mining effectively finds meaning groups that share common interests and behaviors and effectively extracts knowledge needed to understand the motivation for using elder self-care. The analysis results can be used by experts in medicine, public health, nursing and psychology to further research and to assist in policy-making in the health care domain. Future research will apply the sequence-based clustering results for the ComCare project to improving personalized elder self-care services.

6 Acknowledgments

This study is conducted under the “Smart Living Technology and Service program” of the Institute for Information Industry which is financially supported by the Ministry of Economy Affairs of the Republic of China.

7 References

- Arcury, T. A., J. G. Grzywacz, et al. (2012): Older adults' self-management of daily symptoms: Complementary therapies, self-care, and medical care. *Journal of Aging and Health* **24**(4): 569-597.
- Barger, T. S., D. E. Brown, et al. (2005): Health-status monitoring through analysis of behavioral patterns. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* **35**(1): 22-27.
- Bayir, M. A., I. H. Toroslu, et al. (2012): Discovering better navigation sequences for the session construction problem. *Data & Knowledge Engineering* **73**(0): 58-72.
- Carmona, C. J., S. Ramírez-Gallego, et al. (2012): Web usage mining to improve the design of an e-commerce website: OrOliveSur.com. *Expert Systems with Applications* **39**(12): 11243-11249.

- Chen, L., S. S. Bhowmick, et al. (2009): COWES: Web user clustering based on evolutionary web sessions. *Data & Knowledge Engineering* **68**(10): 867-885.
- Chung-Chih, L., C. Ming-Jang, et al. (2006): Wireless health care service system for elderly with dementia. *Information Technology in Biomedicine, IEEE Transactions on* **10**(4): 696-704.
- Dimopoulos, C., C. Makris, et al. (2010): A web page usage prediction scheme using sequence indexing and clustering techniques. *Data & Knowledge Engineering* **69**(4): 371-382.
- Dobrescu, R., M. Dobrescu, et al. (2009): Embedded wireless homecare monitoring system. *eHealth, Telemedicine, and Social Medicine, 2009. eTELEMED '09. International Conference on*.
- Facca, F. M. and P. L. Lanzi (2005): Mining interesting knowledge from weblogs: a survey. *Data & Knowledge Engineering* **53**(3): 225-241.
- Honda, S., K. I. Fukui, et al. (2007): Extracting human behaviors with infrared sensor network. *Networked Sensing Systems, 2007. INSS '07. Fourth International Conference on*.
- Hoy, B., L. Wagner, et al. (2007): Self-care as a health resource of elders: an integrative review of the concept. *Scandinavian Journal of Caring Sciences* **21**(4): 456-466.
- Hussain, T., S. Asghar, et al. (2010): Web usage mining: A survey on preprocessing of web log file. *Information and Emerging Technologies (ICIET), 2010 International Conference on*.
- Kuo, R. J. and L. M. Lin (2010): Application of a hybrid of genetic algorithm and particle swarm optimization algorithm for order clustering. *Decision Support Systems* **49**(4): 451-462.
- Leijdekkers, P., V. Gay, et al. (2007): Smart homecare system for health tele-monitoring. *Digital Society, 2007. ICDS '07. First International Conference on the*.
- Liu, H. and V. Kešelj (2007): Combined mining of Web server logs and web contents for classifying user navigation patterns and predicting users' future requests. *Data & Knowledge Engineering* **61**(2): 304-330.
- Mobasher, B., R. Cooley, et al. (2000): Automatic personalization based on Web usage mining. *Communications of the ACM* **43**(8): 142-151.
- Park, D. H., H. K. Kim, et al. (2012). A literature review and classification of recommender systems research. *Expert Systems with Applications* **39**(11): 10059-10072.
- Park, S., N. C. Suresh, et al. (2008): Sequence-based clustering for Web usage mining: A new experimental framework and ANN-enhanced K-means algorithm. *Data & Knowledge Engineering* **65**(3): 512-543.
- Pierrakos, D., G. Paliouras, et al. (2003): Web Usage Mining as a Tool for Personalization: A Survey. *User Modeling and User - Adapted Interaction* **13**(4): 311-372.
- Sajid, N. A., S. Zafar, et al. (2010): Sequential pattern finding: A survey. *Information and Emerging Technologies (ICIET), 2010 International Conference on*.
- Seon-Woo, L., K. Yong-Joong, et al. (2007): A remote behavioral monitoring system for elders living alone. *Control, Automation and Systems, 2007. ICCAS '07. International Conference on*.
- Wang, C. C. C., W. Y. Wang, et al. (2007): *Development of a service-oriented healthcare architecture for institution-based care. TENCON 2007 - 2007 IEEE Region 10 Conference*.
- Wang, Y.-T. and A. J. T. Lee (2011): Mining Web navigation patterns with a path traversal graph. *Expert Systems with Applications* **38**(6): 7112-7122.

Data Guided Approach to Generate Multi-dimensional Schema for Targeted Knowledge Discovery

Muhammad Usman, Russel Pears, A.C.M. Fong

School of Computing and Mathematical Science
Auckland University of Technology
Auckland, New Zealand

muhammad.usman@aut.ac.nz, russel.pears@aut.ac.nz, alvis.fong@aut.ac.nz

Abstract

Data mining and data warehousing are two key technologies which have made significant contributions to the field of knowledge discovery in a variety of domains. More recently, the integrated use of traditional data mining techniques such as clustering and pattern recognition with data warehousing technique of Online Analytical Processing (OLAP) have motivated diverse research areas for leveraging knowledge discovery from complex real-world datasets. Recently, a number of such integrated methodologies have been proposed to extract knowledge from datasets but most of these methodologies lack automated and generic methods for schema generation and knowledge extraction. Mostly data analysts need to rely on domain specific knowledge and have to cope with technological constraints in order to discover knowledge from high dimensional datasets. In this paper we present a generic methodology which incorporates semi-automated knowledge extraction methods to provide data-driven assistance towards knowledge discovery. In particular, we provide a method for constructing a binary tree of hierarchical clusters and annotate each node in the tree with significant numeric variables. Additionally, we propose automated methods to rank nominal variables and to generate candidate multidimensional schema with highly significant dimensions. We have performed three case studies on three real-world datasets taken from the UCI machine learning repository in order to validate the generality and applicability of our proposed methodology.

Keywords: Data Mining, Data Warehousing, Schema Generation, Knowledge Discovery.

1 Introduction

Knowledge discovery from data is the result of an exploratory process involving the application of various algorithmic procedures for manipulating data (Bernstein et al., 2005). Data mining and warehousing are two key technologies for discovering knowledge from large datasets. Data mining permits targeted mining of large datasets in order to discover hidden trends, patterns and rules while data warehousing facilitates the interactive

exploration and multidimensional analysis of summarized data. These two technologies are mature in their own right and have essentially the same set of objectives but little research has been carried out to seamlessly integrate the two. This is a challenging task as the techniques employed in each of the technologies are different.

In the past several years, a wide range of data mining techniques have made significant contributions to the field of knowledge discovery in a number of domains. In the banking sector, these techniques are used for loan payment prediction, customer credit policy analysis, classification of customers for targeted marketing, and the detection of money laundering schemes and other financial crimes. Similarly, in the retail industry, such techniques are used in the analysis of product sales and customer retention. In the telecommunication industry these techniques help in identifying and comparing data traffic, system workload, resource usage, profit and fraudulent pattern analysis (Han and Kamber, 2006).

Similarly, data warehousing has contributed extensively as a key technology for complex data analysis, decision support and automatic extraction of knowledge from huge data repositories (Nguyen et al., 2005). It provides analysts with a competitive advantage by providing relevant information to enhance strategic decision making. Moreover, warehousing has reduced costs by tracking trends, patterns, and exceptions over long periods in a consistent and reliable manner. Due to sophisticated analytical powers, these warehouse systems are being used broadly in many sectors such as financial services, consumer goods and retail, manufacturing, education, medical, media, and telecommunication.

The integrated use of data mining and warehousing techniques has been observed recently. A number of proposals (Fabris and Freitas, 2001, Goil and Choudhary, 2001, Messaoud et al., 2006, Missaoui et al., 2007, Pardillo et al., 2008, Usman and Asghar, 2011) emphasized the benefits of integrating these technologies and motivated diverse research directions in the area of knowledge discovery. However, it is a daunting task for data warehouse developers to integrate the outcomes of data mining techniques with data warehouse to perform analytical operations.

Data mining results need to be modelled in the form of a multidimensional schema to support interactive queries for the exploration of data. Multi-dimensional modelling is complex and requires extensive domain knowledge along with a familiarity with the data warehouse technologies.

Additionally, its techniques require multiple manual actions to discover measures and relevant dimensions from the dataset, creating a bottleneck in the knowledge discovery process. Even if the human data warehouse designer tries to resolve these problems, an incorrect design can still be generated if he/she doesn't understand the underlying relationships among the data items. In data warehouses, the choice of the dimensions and measures heavily influences the data warehouse effectiveness (Pighin and Ieronutti, 2008).

In prior research (Usman and Pears, 2010, Usman and Pears, 2011, Usman et al., 2011) methodologies have been developed in which the clustering and visualization techniques of data mining are integrated with the well-known multi-dimensional designing technique of data warehouse. This methodology constructs a renowned *STAR* schema through the use of hierarchical clustering and enhances the schema design by using the famous visualization technique known as multi-dimensional scaling. However, the design and implementation of generic methods is required for the process of knowledge discovery from this generated multidimensional schema. Moreover, the previously developed methodologies have two major limitations. Firstly, the methodologies need key extension for extracting more complex knowledge from the multidimensional schema. Secondly, it lacks automated support for data warehouse designers to identify more informative dimensions from high dimensional data to generate compact and informative schema for targeted knowledge discovery. This motivated us to formulate a generic methodology to provide data-driven assistance to a wide spectrum of users especially in those domains where very limited domain knowledge exists.

In this paper we make the following contributions in the field of integrating data mining with data warehousing. Firstly, we present a generic methodology for the generation of multidimensional schema by combining the benefits of hierarchical clustering and multi-dimensional scaling techniques of data mining discipline. Secondly, we provide an algorithm for constructing a binary tree from hierarchical clustering results (dendrogram). Thirdly, we identify the significant numeric variables that define the split of a cluster in the binary tree and rank them in order of significance. Fourthly, we rank nominal variables in each cluster by performing comparative analysis of similar variables distributed in neighbouring clusters (parent & sibling). Fifthly, we provide an algorithm to construct candidate schema with highly ranked dimensions (nominal variables) and measures (numeric variables). Finally, we highlight the significant interrelationships between the highly significant dimensions and measures present in the generated schema to be explored in an OLAP manner.

The rest of the paper is organized as follows. In the next section, we look at the previous work in the area of integrating data mining with warehousing. In section 3, we give an overview of the proposed methodology while section 4 presents the implementation details. The methodological steps are explained through an example in section 5. Our case study results are discussed in section 6. Section 7 gives a general comparison of

proposed work with existing manual data analysis. Finally, we conclude the paper in section 8 with a summary of achievements and discussion of possible future research directions.

2 Related Work

In the last decade, many researchers have recognized the need for the integration of data mining with data warehousing (Kamber et al., 1997, Missaoui et al., 2007, Zubcoff et al., 2007, Goil and Choudhary, 2001). For instance, (Goil and Choudhary, 1997) identified that the decision making process requires complex operations on underlying data which can be very expensive in terms of computational time. They proposed an algorithm for the construction of data cubes on distributed memory parallel computers. In their approach they integrated OLAP and the Attribute Focusing (AF) technique, which relies on exploration and interpretation of attributes, of data mining. AF calculates associations between attributes using the notion of percentages and sub-populations. This integration allows interesting pattern mining on distributed data cubes.

A similar framework was proposed by the same authors that used the summary information present in the data cubes for association rule mining and decision tree classification (Goil and Choudhary, 1999). Data mining uses some pre-aggregating calculations to compute the probabilities needed for calculating support and confidence measures for association rules and the split point evaluation while building a classification tree (Goil and Choudhary, 2001). A limitation of these approaches is that they focused only on improving the OLAP query processing time by constructing the distributed data cubes. However, these distributed data cubes are unable to provide any automated assistance on which dimension and measures of the cubes should be explored in a high dimensional space for focused knowledge discovery.

Similarly, another system that implements a wide spectrum of data mining functions (association rule mining, clustering, etc.) called *DBMiner* has been proposed to support multiple mining functions (Han, 1998). The work focused on the efficiency and scalability of the data mining algorithms with the help of exploratory analysis provided by OLAP systems. The author emphasized that data mining should be performed interactively like an OLAP analysis and at different levels of data abstraction. This work has an obvious advantage of supporting multiple data mining functions. However, the author proposed mining to be applied on the pre-aggregated data cube, which fails to provide any assistance to the human data warehouse developers to build an appropriate schema in the first instance on which mining techniques can be applied later. Furthermore, the proposed system requires the analyst to possess knowledge of both data characteristics and the roles of data mining functions to select an appropriate mining algorithm. Analysts in various domains lack complete knowledge about the data and mining algorithms. It makes the application of such a proposal very difficult and identifies the need for an approach that can assist such analysts in the meaningful knowledge discovery. Similar to (Han, 1998), (Liu and Guo, 2001) put forward

a new architecture for integrating data mining with OLAP. Yet again, the proposed architecture supported mining on the pre-aggregated data cubes built on top of manually constructed schema. However, the authors supported the idea of mining at different stages of the knowledge discovery process and at multiple levels of data abstraction. This approach is similar to our approach in terms of mining at multiple levels of data abstraction as we also produce hierarchical clusters at multiple levels of data abstraction for analysing information at multiple levels. However, the difference is that we not only apply mining at multiple levels of data abstraction but also provide a set of significant numeric and nominal variables and the strong relationships that exist between the two.

More recently, (Usman et al., 2010) proposed an enhanced architecture for combining mining and warehousing to provide OLAP performance enhancement and visualization improvement. However, the analyst relies on a few levels of data abstraction and manual discovery of useful data chunks provided by the neural network technique with no way of interactively exploring the original data variables of a particular cluster in order to extract useful knowledge. Each cluster and the cluster hierarchy were assumed to be the navigational space for cube data exploration. That limited search space and limited navigation was not adequate for meaningful knowledge discovery from large data cubes. These limitations led to the work done by (Usman and Pears, 2010) to enhance the previous methodology by providing support for complex data containing a mix of numeric and nominal variables. Moreover, authors utilized visualization technique to identify the hidden relations present in the high cardinality nominal variables. Yet, the work was limited by its generic methods for constructing the binary tree from the clustering results and the method for automatic generation of schema for a given dataset. Additionally, the authors assumed that all the nominal variables in the dataset were candidates for dimensional variables. This proves to be very unrealistic as real-world datasets consist of a large number of nominal variables and all of these nominal variables cannot be taken as dimensions. There is a need to identify the highly informative dimensions and eliminating the least informative ones to form compact data cubes at multiple levels of data abstraction. Also, a large number of dimensions increase the cube construction time as many views need to be materialized. In this paper, we overcome these limitations and provide methods to identify and rank dimensions in order of significance to prune the search space and assist users in targeted discovery.

It is apparent that most of related work in this area is towards the application of data mining techniques on the top of a manually constructed schema or data cube. Little research has been carried out in exploiting data mining techniques to generate multidimensional schema. Multidimensional modelling is a complex task and we believe that mining techniques can not only assist in schema design process, but can also improve the knowledge discovery process. Moreover, the related work has a number of limitations and there is a strong need for tight coupling of the two disciplines by providing some

generic methods to automate the knowledge discovery process, making it more data driven.

3 Overview of Proposed Methodology

In this section, we give an overview of our proposed methodology for the seamless integration of data mining and data warehousing. A critical question in the design of such an integrated methodology is how to integrate or couple the mining techniques within a data warehousing environment? According to (Han and Kamber, 2006) the possible integration schemes include no coupling, loose coupling, semi tight coupling and tight coupling. Tight coupling means that data mining system is smoothly (seamlessly) integrated into the data warehouse system. Such smooth integration is highly desirable because it facilitates efficient implementation of data mining algorithms, high system performance and an integrated knowledge discovery environment.

However, implementation of such a system is a hard task as the techniques employed by each discipline are substantially different from each other. Multidimensional modelling is a challenging task requiring domain knowledge, solid warehouse modelling expertise and deep understanding of data structure and variables (Han and Kamber, 2006). In a real world scenario, data warehouse designers possess the modelling expertise but lack the domain knowledge and thorough understanding of semantic relationships among data variables, which can lead to a poor warehouse design that can dramatically affect the knowledge discovery process. Data mining techniques such as clustering and pattern visualization can assist in understanding and visualizing the complex data structures. A methodology for such data driven multidimensional modelling is required to support both the human warehouse designers and the decision makers to extract useful knowledge from the data.

The aim of our proposed methodology is tight coupling of the two technologies. We employ a hierarchical clustering technique along with the well-known parallel coordinate technique (Rosario et al., 2004) to assist the analysts and designers in finding natural groupings in the data. In the absence of explicit domain expert input we rely on the strengths of hidden relationships and groupings among the variables in a given dataset. Figure 1 depicts the main steps of the proposed methodology. We provide an overview of each step in this section and the details of the tools and methods involved in each step are explained in the section 4.

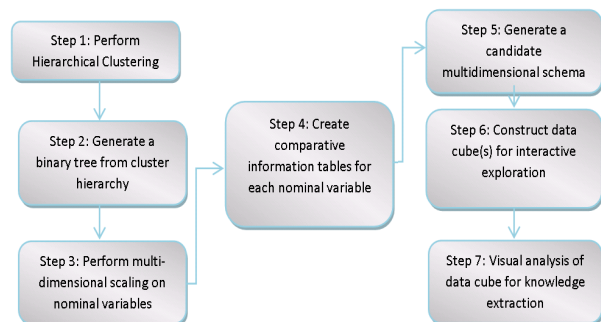


Figure 1: Proposed methodological steps for knowledge extraction

The first step is to apply agglomerative hierarchical clustering algorithm on the data to generate hierarchical clusters based on a similarity measure. For the purpose of generating clusters, we use only the numeric variables in order to get optimal clustering results, as clustering algorithms perform well on numeric data. In the second step, we generate the binary tree from the hierarchical dendrogram (tree) and efficiently determine the cut-off point in the tree. By efficiently determining the cut-off point we can limit the number of clusters produced by the mining algorithm for analysis and highlight the significant variables in each cluster which define the split of a particular cluster and name each cluster according to its data abstraction level in the binary tree.

For instance, we name the cluster at the first level as C1 and C2. Each cluster represents a parent child relationship in the decision tree, so C1 becomes the parent of C11 and C12 at second level in the hierarchy. In our methodology, we intend to compare the similarities and differences among parent, child and sibling (e.g. C1, C11 and C12) relationships. We take out nominal data from each cluster that did not play any part in the clustering process. In the third step, the effective visualization technique known as Distance-Quantification-Classing (DQC) proposed by (Rosario et al., 2004) for nominal variables is applied on extracted nominal data. This technique maps nominal values to numbers in a manner that conveys semantic relationships by assigning order and spacing among the values, helping in the visualization of the natural grouping among the data values in each nominal variable.

In the fourth step, we construct an information table for each nominal variable present in a cluster by comparing it with its values in neighbouring clusters (parent and sibling). For instance, C11 is compared with its parent (C1) and sibling (C12). We have developed a generic method that calculates the amount of change occurred in each nominal variable at various level of data abstraction and rank the variables based on the calculated value of change from higher to lower. Higher calculated change suggests more degree of variation in a given variable and thus it becomes the potential dimension for multidimensional schema. This highest degree of variation highlights the fact that a given nominal variable has highly significant characteristics as compared to its neighbouring clusters. Therefore, it should be taken as the most informative dimension to explore the numeric variables to discover hidden relationships among numeric and nominal variables.

Such variables with highly significant differences explain the hidden relations between the numeric and nominal variables. In the fifth step, we use the outcome of the visualization technique and comparative information tables to construct the multidimensional schema. The groupings present in ranked dimension in the previous helps in formulating the dimensional hierarchy. Finally, a set of generic *SQL* queries create dimension and fact tables and populate the tables of the multidimensional schema.

In the sixth step, we construct the data cube using the generated schema using automated queries. The final step of the methodology utilizes the data cube by employing

the OLAP visualization technique. Users can visually explore the targeted cube structure to extract useful information and knowledge from the underlying data cube built upon multidimensional schema.

4 Integration Tools and Methods

In this section we present the details of the tools and methods involved in the methodology for the seamless integration.

4.1 Hierarchical Clustering of numeric variables

For the purpose of the hierarchical clustering of numeric data we have employed the Hierarchical Clustering Explorer (HCE) tool (Seo and Shneiderman, 2006) as it allows interactive analysis of multidimensional data and helps in identifying the natural number of clusters with interactive visual feedback and dynamic mining query controls. It supports various clustering parameters such as linkage methods (single, average or complete), clustering directions and similarity/distance measure. We use the most commonly used distance measure known as *Euclidean* distance and complete linkage method in our cluster analysis step.

4.2 Hierarchical Clustering of Numeric Variables

In this step, we generate a binary tree to a suitable data abstraction level. This is an important step because the clustering algorithm starts from the leaf nodes, which are the data points, and merges the nodes based on distance measure, creating a single cluster. In large data sets this approach can produce a huge number of clusters and it is not feasible to analyse the clusters which have very few records or no significant information available for extraction. It is a crucial stage of the methodology to determine the data abstraction level in the cluster hierarchy, which can give the analyst a meaningful picture of the overall data distribution. This leads to a sub step 2.1, determining the cut-off point.

4.2.1 Determining cut-off point in cluster hierarchy

Various ways to determine a suitable cut-off point of the hierarchical tree (dendrogram) have been proposed in the literature. (Basak and Krishnapuram, 2005) have used a lower bound on the minimum number of data points in a cluster as the stopping criterion. However, they suggested the depth of the decision can also be used in combination with the minimum number of data points. In the Biofuel industry (Rivière and Marlair, 2010) relied on cutting the tree at various levels and finally decided (based on domain expertise) the cut-off point where the clustering was more understandable. A more appropriate and flexible approach is adopted by (Seo and Gordish-Dressman, 2007) using the HCE tool built in controls of the minimum similarity bar and detailed cut-off bar. The minimum similarity bar helps users find the right number of clusters and the detail cut-off bar helps users control the level of detail by rendering the sub clusters below the bar with their averages.

In the proposed methodology we retain the flexibility of choosing the specific number of clusters supported by

the HCE tool; however, users often do not have a clear idea how many clusters will be adequate for the analysis. To tackle this problem we use the linkage inconsistency coefficient threshold, which is defined as the length of a link in a cluster hierarchy minus the average length of all links divided by the standard deviation of all links (Cordes et al., 2002), to find the natural cluster division in the dataset. If the length of the link in dendrogram has approximately the same length as neighbouring links then the objects have similar features (high consistency) and vice versa. We calculate this threshold by using the *MATLAB* inconsistent function that returns data about the links and returns an inconsistency coefficient value. This calculated threshold value helps in determining the suitable cut-off point for a given dataset.

4.2.2 Cluster Naming, Significant Splitting Variable Identification and Data Extraction

After determining the suitable threshold value for the cut-off point the next task is to name the cluster in a logical and meaningful way. Cluster naming helps identify the data group that needs further exploration. As all the numeric variables are used in the clustering process, it is vital to identify the significant variables in a given cluster say C1. These significant numeric variables define the splitting of a cluster into two sub clusters, for instance, C11 and C12, and assist in picking the suitable measures or facts for the multidimensional model to be built. For this purpose, we perform a statistical function called the analysis of variance (ANOVA) and sort the variables based on their variance value. We construct the binary tree showing the cluster names and significant variables in each cluster. However, the high dimensionality of the data sets still hinders users from finding interesting patterns and outliers. One reason for this is the presence of nominal variables in each cluster which are not involved in the clustering process. In order to find hidden relationships among nominal and numeric data, we extract the nominal data from each cluster and store it in temporary files to be used later. Algorithm 1 elaborates the steps involved in generating a binary tree.

Algorithm 1. Binary tree generation

Input: HD, hierarchical dendrogram result
TH, a calculated threshold value for cut-off
K, total number of data records

Output: A binary tree highlighting significant splitting variables and number of instances in each cluster.

Method:

1. Set SV ← 0 /*SV is the initial similarity value where all records are in single cluster
2. Initialize Cn = 1 ; NL = 0 /*Cn is the total number of clusters and NL is the data abstraction level
3. Assign Main_CL ← K DT = O /*Main_CL represents the root cluster & DT is the data structure to store the decision tree with links L(1 to n).
4. while (SV <= TH)
5. repeat
6. check increment of 1 in Cn value /*to identify a new cluster while navigating in the hierarchy
7. if Cn is incremented
8. get value of SV at that point in NV /*NV is for storing the similarity value where the first split occurs
9. Assign SV ← NV
10. Increment NL by 1
11. if NL == 1
12. Extract numeric and nominal data from each cluster at this similarity in separate arrays ED(num) = (Vi to n) and ED(nom) = (Vj to n) /*ED(num) is for storing the numeric variables and ED(nom) is for nominal variables
13. Give name to each cluster using abbreviation "C" and concatenate it with numeric subscript of 1 and 2
14. Apply ANOVA function on ED(num) array data.
15. Sort the numeric variables present in ED(num) on the basis of variance
16. Draw parent-child links and highlight the significant numeric attributes in descending order and number of records present in each cluster
17. Store the decision tree in DT with all the links Li to Ln
18. else
19. repeat step 12 to 17 for the cluster that is portioned into two sub cluster at lower level of data abstraction.
20. store array ED(nom) with variables and instances /*ED(nom) array is stored for the use of applying visualization technique for later steps of the proposed methodology
21. end if
22. end while
23. return DT

Figure 2: Algorithmic steps of binary tree generation

4.3 Perform Multidimensional Scaling on Nominal Variables

After retrieving the nominal data, we apply the DQC technique to efficiently map the nominal values into numbers. This technique is implemented in a java based program developed by (Rosario et al., 2004). The program maps the nominal values efficiently to numbers so that the semantic relationships among the values can be easily visualized. Two output files are created by the program, one holding the mapped values and the other an *xml* file for meta-data. The purpose of this implementation is the visualization of nominal values using a well-known visualization technique called parallel coordinate. The major drawback of this parallel coordinate technique is that it is only suitable when there is small number of nominal variables. Visualization of more than 10 nominal variables becomes very unclear using parallel coordinate display.

Moreover, this technique fails when there are a large number of instances in a single nominal variable. For instance, *Country* is a nominal variable which usually have more than 40 distinct values (country names). Such variables are difficult to visualize using parallel coordinates as most of the values overlap each other. This motivated us to find a solution for extracting the groupings produced by this visualization technique. In our investigation, we identified that the output *xml* file produced by the Java based tool holds the structure of all the nominal variables and their instances. This led to the development of our own prototype that reads this output *xml* file and groups the values based on automatically calculated thresholds for each nominal variable. In addition to this, the developed prototype also allows comparative analysis of a given cluster with neighbouring clusters. However, our prototype is flexible enough to allow comparison of a given cluster with any other cluster present in the hierarchical tree. Algorithm 2 illustrates the steps involved in our developed prototype for finding natural groupings in nominal variables.

Algorithm 2. Grouping nominal values and cluster comparison

Input: XML metadata file of cluster(s) containing nominal data information

Output: A dimension tree showing the nominal variables
Natural groupings of values in each nominal variable
Group and value comparison of selected clusters

Method:

1. read nominal variables from temp[N] and populate the tree view control /*the tree view control allows displaying the nominal variables present in a cluster in a hierarchical tree
2. Attach XML data node as a data input tag with relevant tree nodes /*this step attaches the instances (values) of each nominal variable and each variable is displayed as a dimension
3. Invoke processing of calculation with OnSelect function /*OnSelect function allows the selection of a dimension so that its grouping and values can be displayed
4. Fetch scale and category tag values for each dimension /*in xml file the mapped numeric values are stored in the scale tag and its actual nominal value is store in the category tag.
5. Find min(scale) and max(scale) for each dimension /*minimum and maximum values are extracted
6. Calculate Th = max (scale) - min (scale) / T (category) /*Th is the threshold that is calculated for each dimension. T(category) denotes the total number of distinct values present in a dimension
7. For each dimension D (1 to n) /*n is the total number of dimensions
8. Create group (Group1) and assign initial scale value (Vi)
9. Take next scale value Vn and subtract it with the previous one
10. Store the subtraction result in R and compare it with Th /*threshold comparison
11. if R < Th
12. Then add the value in Group1
13. Else
14. Create next group (Group2) and assign scale value (Vi) to it
15. Repeat step 15 to 21 for all the values present in the scale
16. Eliminate groups having single value and gather the values in a new group called Group_others /*Outlier values present in the dimension which are not close to any other values
17. End if
18. End for
19. Display the values of all created groups in tabular form

Figure 3: Algorithmic steps of grouping similar nominal values

Algorithm 2.1 describes the steps for comparative analysis (Parent-Sibling) of selected clusters. Moreover, Algorithm 2.1 highlights the similarities and differences among the dimensions in different clusters. The outcomes of cluster comparison steps are fed into our next step for the generation of rich information tables for each cluster in the tree.

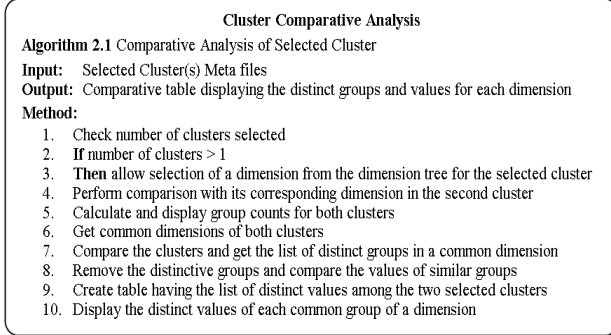


Figure 4: Algorithmic steps of comparative analysis of clusters

4.4 Create Comparative Information Tables for Each Nominal Variable

This step consists of two main tasks. First, providing a method for calculating the variable change at multiple levels of data abstraction and second, ranking the variables to highlight the potential dimensions for the multidimensional schema. Table 1 shows the generic structure of the information that is provided for each dimension in a given cluster, for instance cluster C1.

Name	Groups	Values	Parent comparison	Sibling comparison	Calculated change
Dim1	Dim1 G1	G1 {V1....Vn}	Par(Dim1G1) {V1....Vn}	Sib(Dim1G1) {V1....Vn}	Count (Pdv + Pdg)
	to				+
	Dim1 Gn	Gn {V1....Vn}	Dim1 G1 {V1....Vn} = Pdv and distinct groups (Pdg)	Dim1 G1 {V1....Vn} = Sdv And distinct groups (Sdg)	Count (Sdv + Sdg) = Dim1AC

Table 1: Structure of information table for a single dimension in a cluster

We explain the notations used in each column of Table 1. In the first column the name of a dimension is displayed. Each cluster consists of a set of dimension say *Dim* {1,2,...,n} where n is the total number of dimensions.

The second column shows the natural groups present in the first dimension (*Dim1*). The groups in a dimension can vary from 1 to n number of groups. We name the groups in ascending order starting from Group1 (*G1*), Group2 (*G2*) and so on. The third column shows the values which are grouped together using the procedure shown in Algorithm 2.

Similarly, every group has *n* number of values {V1,V2,...,Vn} in it. The forth column is the calculation column that calculates the change of a given dimension in a cluster with its corresponding dimension in the parent, with two sets of comparisons performed in this column. First we perform a value comparison. Each group of selected cluster dimension is compared with its corresponding group in the parent cluster. For instance,

Group1 of C11 is compared with Group1 of C1 for a particular dimension Dim1. The resultant is a set of distinct values present in the parent dimension but that are not present in *Dim1*. This is achieved by applying the minus operator and the result is stored in the *Pdv* variable. Secondly, we compare the groups of the Dim1 with the groups of its parent (*Par(Dim1)*). We calculate the group change in another variable called *Pdg*. In the fifth column the same set of comparisons are performed with the sibling dimension, (*Sib(Dim1)*). In the last column, we count the parent change (*Pdv + Pdg*) and sibling change (*Sdv + Sdg*) and store the result of accumulated change in the *Dim1AC* variable. This accumulated change is calculated for each dimension *Dim1* to *Dim(n)* and the dimensions are ranked on the basis of this calculated value. The dimension that possesses higher accumulated change is ranked higher and vice versa. The generic method of calculation is applied to all the dimensions present in a cluster to get the accumulated dimensional change represented by equation 1.

$$ADC = \sum_{i=1}^n \{Dim(i) AC\} \quad (1)$$

This helps in presenting the overall change present in a cluster as compared to its neighbouring clusters. The formula for calculating overall cluster change is represented in equation 2.

$$OCC = \sum_{i=1}^n \{ADC + SA\} \quad (2)$$

here *SA* represents the count of significant variables present in a cluster as a result of performing the ANOVA function described in step 2.2 of the proposed methodology. Figure 5 depicts an example of information provided to the user after the comparison with its neighbouring clusters.

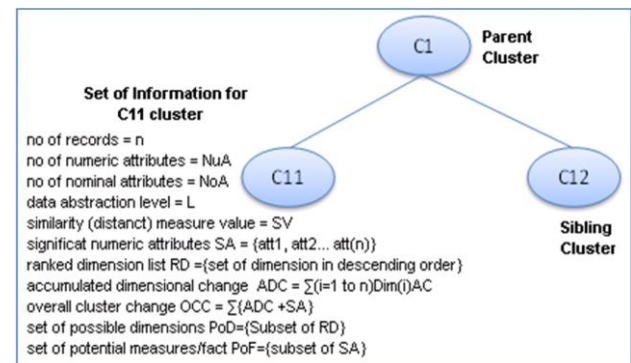


Figure 5: Set of information for cluster C11 after parent (C1) and sibling (C12) comparison

After performing comparisons a set of highly ranked dimensions and measures are fed into the automatic schema generator that constructs the well-known *STAR* schema for multidimensional analysis.

4.5 Generate a Candidate Multidimensional Schema

For the generation of schema, we have developed a generic method that constructs the *STAR* schema in

Microsoft SQL Server 2005 database. The steps are presented in Algorithm 3. Figure 7 depicts the script that is used by the schema generator to create a dimension table in the database server. Each dimension consists of a group level and an individual value level and defines the concept hierarchy of the generated dimensions for the dimensional data navigation. Figure 8 represents the structure of the fact table creation script. First, it creates the fact table ID followed by selected dimension IDs. In the end, it applies primary key and foreign key constraints on the dimensions.

Algorithm 3. Automatic schema generation

Input: Ck, selected cluster with records; PoD, set of ranked dimensions; PoF, set of potential measures
 DG, set of group each dimension; DGV, set of values in each group
Output: STAR schema having the multidimensional data in a database server
Method:

1. Fetch PoD and PoF of a given cluster Ck
2. For each dimension present in PoD
3. Get DG [g1 to gn] and DGV [v1 to vn] in tabular form /* to get the groups and their values
4. End for
5. Establish connection string /* connection is established to communicate with the database server
6. Create new database DB (Ck) /* a new database is created for Ck schema
7. For each dimension in PoD
8. Create dimension table using create_dimension SQL script /* refer Figure 7
9. Insert values of DG and DGV in respective columns of the newly created dimension table
10. End for
11. Create fact table for Ck using create_facttable SQL script /* refer Figure 8
12. Populate fact table of Ck using populate_facttable SQL script /* refer Figure 9

Figure 6: Algorithmic steps of schema generation

After creating the fact table, the script in Figure 9 is used to insert (populate) data in it. It selects the dimension IDs from the dimension table and the measures or facts in the given cluster Ck where the names in source data cluster Ck are equivalent to the distinct names in the dimensional table. These scripts create a complete multidimensional model in the database server for the use of cube construction.

```
CREATE TABLE [Ck_Dim1] (
    [Dim1_id] [int] IDENTITY (1, 1) NOT NULL,
    [Dim1_grouping] [varchar] (50),
    [Dim1_name] [varchar] (50),
    CONSTRAINT [PK_Ck_Dim1] PRIMARY KEY
    ([Dim1_id]) ON [PRIMARY]
) ON [PRIMARY]
GO
```

Figure 7: Structure of SQL script for creating dimension table

```
CREATE TABLE [Ck_Fact_Table] (
    [Fact_Table_Id] [int] IDENTITY (1, 1) NOT NULL,
    [Dim1_Id] [int] NULL, [Dim2_Id] [int] NULL, ..... to ..... [DimN_Id] [int] NULL,
    CONSTRAINT [PK_C1_Fact_Table] PRIMARY KEY
    ([Fact_Table_Id]) ON [PRIMARY],
    CONSTRAINT [FK_Ck_Fact_Table_Ck_Dim1] FOREIGN KEY
    ([Dim1_Id]) REFERENCES [Ck_Dim1] ([Dim1_Id]),
    to
    CONSTRAINT [FK_Ck_Fact_Table_Ck_DimN] FOREIGN KEY
    ([DimN_Id]) REFERENCES [Ck_DimN] ([DimN_Id]),
) ON [PRIMARY] GO
```

Figure 8: Structure of SQL script for creating fact table and setting dimension and fact tables' relationships

```
INSERT INTO Ck_Fact_Table
(Fact_Table_Id, Dim1_Id, ..... DimN_Id)
SELECT Ck.IDs, and D1.Dim1_Id, ..... Dn.DimN_Id
and Ck.measure1, ..... to ..... Ck.measureN
FROM Ck_Dim1 D1, ..... to ..... Ck_DimN Dn and Ck
WHERE
Ck.D1 = Ck.Dim1_name ,
Ck.D2 = Ck.Dim2_name ..... to ..... Ck.Dn = Ck.DimN_name
```

Figure 9: Structure of SQL script for populating data in fact table

4.6 Construction of Data Cubes for Visual Exploration

In this step, we use Microsoft Analysis services 2005 software to construct the cube from the automatically generated multidimensional schema. We use MOLAP because this storage type maps a multidimensional view directly to the data cube array structure. This gives the advantage of fast indexing of pre-computed summarized data. The proposed methodology allows construction of data cubes containing highly informative dimensions and measures present in each cluster and eliminate the lowly ranked (less informative) dimensions. It efficiently narrows down the cube search space and improves cube construction time as fewer views remain for cube materialization.

4.7 Visual Analysis of Cube for Knowledge Extraction

In the final step of the methodology, we have developed an application for the front-end OLAP analysis by using the OLAP Services Control developed by Dundas Data Visualization, Incorporation. The application connects with the Cube server and shows the data cubes for performing OLAP operations such as drill-down, roll-up, slice and dice. Analysts can interactively and visually explore the data cube which has the most significant dimensions and measures of a given cluster. These most significant dimensions assist the users to quickly identify the hidden relationships between the highly informative dimensions and thus lead the analyst to extract knowledge from a constrained yet informative search space.

5 An Example of application – Automobile dataset

In this section, we discuss an example to illustrate the steps of our proposed methodology. We use an *Automobile* (Schlimmer, 1985) dataset which has 205 records and 26 variables. This data set consists of three types of entities: (a) the specification of an auto in terms of various characteristics, (b) its assigned insurance risk rating, (c) its *normalized losses* in use as compared to other cars. The second rating corresponds to the degree to which the auto is more risky than its price indicates. Cars are initially assigned a risk factor symbol associated with its price. Then, if it is more risky (or less), this symbol is adjusted by moving it up (or down) the scale. This process is called "symboling". A value of +3 indicates that the auto is risky, -3 that it is probably pretty safe. A full description of all the variables of this dataset can be found at the UCI machine learning website (Asuncion and Newman, 2010). For the first step, we use the 16

numeric variables present in the dataset to perform hierarchical clustering.

From this dendrogram we determine the suitable cut-off point and generate the binary tree using Algorithm 1, which helps the analyst to pick the cluster of his choice for further analysis. At this stage, we provide the significant numeric variables present in each cluster of the tree. Figure 10 depicts the binary tree generated for the automobile dataset. Each cluster is given a unique name and the significant variables are shown in the surrounding of the cluster. The next step of the methodology is the application of Distance-Quantification-Classing (DQC) technique in order to identify semantic relationships among high cardinal nominal variables.

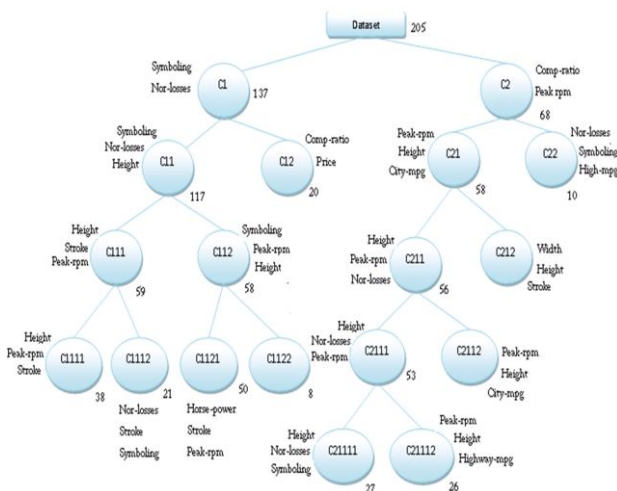


Figure 10: Binary tree for generated from the dendrogram showing significant numeric variables

Our developed prototype takes clusters nominal data as input to show the groupings for each nominal variable. Figure 11 shows the groupings of the *Make* variable in Cluster C12 using our developed prototype.

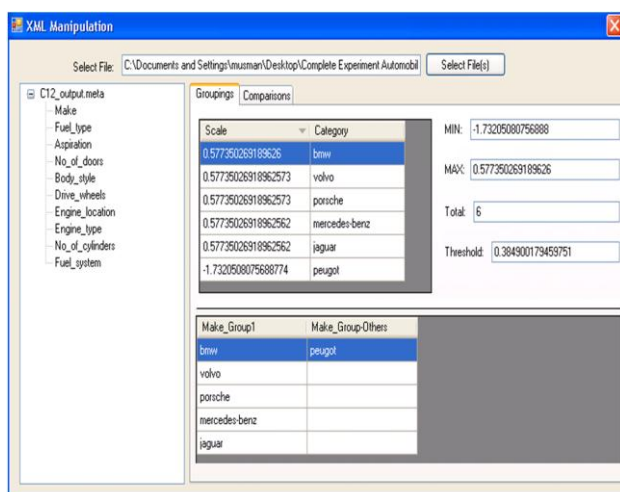


Figure 11: View of the groupings achieved for Cluster C12 through the developed prototype

It can be seen that Cluster C12 has two groups and the automobiles that have a strong association in the underlying cluster records are grouped together. For instance, *BMW*, *Volvo*, *Porsche*, *Mercedes-Benz* and

Jaguar are closer to each other. On the other hand, *Peugeot* is a local outlier because no other automobile has similar variables. At this stage, the analyst can identify the strong relationships as well as the anomalies present in each nominal variable in a given cluster. Using Algorithm 3, we generate the multidimensional schema and construct data cubes, allowing various analytical operations and giving the analyst a targeted multidimensional space for meaningful exploration for discovering knowledge using OLAP operations.

6 Real-world Case Studies (Results and Discussion)

In this section, we discuss the results of the case studies which have been performed on real-world datasets namely *Automobile* (Schlimmer, 1985), *Statlog German Credit data* (Hofmann, 1994) and *Adult* (Kohavi and Becker, 1996) datasets taken from the UCI machine learning repository. In Table 2, we give the summary of the variables present in these three datasets. Further detailed descriptions of each dataset can be obtained from UCI's machine learning website.

Data Set	Variable types	No of attributes	Nominal attributes	Numeric attributes	No of records
Automobile	numeric, nominal	26	10	16	205
German Credit data	numeric, nominal	20	13	7	1000
Adult	numeric, nominal	14	8	6	48842

Table 2: Summary of datasets used for case studies

For our three case studies we first performed hierarchical clustering on numeric variables to get a hierarchy of clusters in the form of a dendrogram. For the adult dataset we removed the records having missing values and used 30162 (61%) records for hierarchical clustering. As clustering results tend to be better with numeric variables, all numeric variables were involved in the clustering process. Other clustering parameters used in this step are explained above in section 4. However, agglomerative hierarchical clustering algorithm only works well for small datasets. It took 329 seconds for the HCE tool to produce a dendrogram for the Adult dataset on a machine with 2 GHz processor and 4 GB of RAM.

A suitable cut-off point in the dendrogram was calculated for each of the datasets using the inconsistency function. This calculation helped in setting up the detailed cut-off bar and suggested the number of levels of data abstraction that should be adequate for extracting knowledge from a given dataset. In Table 3, we highlight the calculated similarity value (cut-off point), the number of levels and number of clusters at the bottom level.

Data Set	Calculated Similarity value for cut-off point (0 to 1)	Total levels of data abstraction for analysis	No of clusters at lowest data abstraction level	Total Clusters sufficient for overall analysis
Automobile	0.676	5	10	18
German Credit data	0.557	7	14	26
Adult	0.623	9	18	34

Table 3: Calculated cut-off points and cluster information at lowest data abstraction

It can be seen from Table 3 that regardless of the size of dataset the hierarchical clustering tree can be efficiently cut at approximately 0.60 similarity measures. However, the levels of data abstraction and the number of clusters are directly proportional to the size of the datasets. After calculating the cut-off point and determining the number of clusters, we extract data from the clusters and apply ANOVA function and find the significant variables in each cluster that define a split. These significant numeric variables are ranked at this stage of the methodology for each cluster and become the possible measures in a ranked order for a given cluster in the hierarchy. The significant variables found for the datasets are shown in Table 4. Due to limited space, we only show the top 3 significant variables of 10 distinct clusters from each dataset. It can be seen from the results in Table 4 that the Automobile dataset each cluster has its own significant variables. However, in a particular section of the tree (or subset of the tree) $\{C1, C11, C12, C111 \text{ \& } C112\}$ the important measures are symboling, normalized losses (nor-losses) and height.

Cluster Names	Automobile	German credit data	Adult
C1	Symboling, nor-losses, height	Residence, credits, Maint-ppl	Final-wgt, Cap-gain, Cap-loss
C11	Symboling, nor-losses, height	Residence, installment, age	Final-wgt, Cap-gain, Cap-loss
C12	Comp-ratio, price	Maint-ppl, credits, installment	Final-wgt, hr-per-week, Edu-no
C111	Height, stroke, peak-rpm	Installment, residence, age	Final-wgt, Cap-gain, Cap-loss
C112	Symboling, peak-rpm, height	Maint-ppl, residence, amount	Final-wgt, Cap-gain, Cap-loss
C2	Comp-ratio, peak-rpm	Month, amount, age	Final-wgt, Cap-gain, Cap-loss
C21	Height, peak-rpm, city-mpg	Amount, age, residence	Final-wgt, Cap-gain, Cap-loss
C22	Nor-losses, symboling, high-mpg	Amount, age, Maint-ppl	Final-wgt, Cap-loss, Age
C211	Height, peak-rpm, nor-losses	Residence, amount, months	Final-wgt, Cap-gain, Cap-loss
C212	Width, height, stroke	Amount, age, residence	Final-wgt, Cap-gain, Cap-loss

Table 4: Significant variables present in each cluster of the three datasets

The only exception is in the C12 cluster in which the compression ratio (comp-ratio) and price variables are significant. This information can play a vital role in the knowledge extraction process. For instance, an analyst interested in looking at the variation of price of the automobiles will focus mainly on cluster C12. On the other hand, symboling value, which represents the safety measure of the car, can be looked at a number of clusters to see its variation. Similarly, in German credit data, the analyst can easily extract the information from a subset of the tree $\{C2, C21, C22, C211 \text{ \& } C212\}$ that *Amount* and *Age* are the two dominant variables. This means that by concentrating on this sub-tree the analyst can find interesting correlation between the two variables namely *Amount* and *Age*.

The third dataset yields the most interesting set of variables from all the clusters. Final Weight (*Final-wgt*) is the weight assigned to people taking into account the (*age, sex and race*) variables. This assigned weight is leading in all clusters. However, there are a few obvious distinctions in cluster C22 and C12. These clusters can be of focus if (*hours per week* or *Age*) variable related data needs to be explored. Up to this point, our methodology targeted only numeric variables but in real world datasets there are a number of highly cardinal nominal variables exist. Mix of numeric and nominal variables require

efficient analysis in order to discover knowledge from these complex real-world datasets with mixed variables.

The prevalent mining technique used for the efficient analysis of mixed data is Clustering. A number of authors (Li and Biswas, 2002, Ahmad and Dey, 2007) have adopted this technique and a variety of algorithms are proposed to tackle the mixed data analysis problem. Hierarchical Clustering, in particular, has shown good results in this area. We also have adopted hierarchical clustering technique for efficient analysis of numeric data. However, we use the visualization technique along with our own information processing method (details given in step 3; section 4). We extract the nominal data from each of the clusters in the tree and apply the visualization technique (DQC) on the nominal variables. After, identifying the groupings in each nominal variable we proceed to construct the rich information tables for the dimension in a cluster. Table 5, shows the information table data produced for the top dimension of *Automobile* dataset. Similarly, the dimensional change value is calculated for the given cluster with respect to its neighbouring clusters to provide detailed information about the nominal variables and the distribution among a specific region in the hierarchical tree. Up to this step, the proposed methodology has identified and ranked the possible measures (significant numeric variables) and the potential dimension (ranked nominal variables). This set of information is given to the binary tree to depict more detailed information about the underlying data in a cluster as shown in Figure 5. It leads to the next step which concentrates on the automatic generation of multidimensional schema.

Name	Groups	Values	Parent Comparison (C111)	Sibling Comparison (C112)	Calculated change
Make	Dim1 G1	G1 [V1...Vn]	Par(Dim1G1) [V1...Vn]	Sib(Dim1G1) [V1...Vn]	Count (Pdv + Pdg)
	to	to	Dim1 G1 [V1...Vn] = Pdv	Dim1 G1 [V1...Vn] = Sdv	+
	Dim1 Gn	Gn [V1...Vn]	And distinct groups (Pdg)	And distinct groups (Sdg)	Count (Sdv + Sdg)
					= Dim1AC
	Group 1	Mazda, Mitsubishi Toyota Honda Nissan Isuzu Dodge, Plymouth	Volvo (1)	Volvo, Peugeot (2)	(1+1+0) + (2+1+0)
	Group others	Subaru	Peugeot (1)	Nissan (1)	= 5
			No distinct group (0)	No distinct group (0)	

Table 5: Information summary of values present in top dimension (Make) of Cluster C111

After the application of Algorithm 3, we generated the schema for each of three clusters in each dataset used in our case studies. Table 6 shows the potential dimensions and significant measures used for generating the STAR schema for three clusters from each dataset representing the parent-child relationship.

Data Sets	Clusters	Significant dimensions	Significant measures
Automobile	C1	Make, Fuel-System, Body-Style, Engine-Type, Cylinders	Symboling, nor-losses, height
	C11	Make, Body-Style, Fuel-System, Engine-Type	Symboling, nor-losses, height
	C12	Make, Engine-Type	Comp-ratio, price
German Credit data	C1	Purpose-of-credit, credit-history, Employment-duration	Residence, credits, maint-ppl
	C11	Purpose-of-credit, Personal-status, credit-history	Residence, installment, age
	C12	Purpose-of-credit, credit-history, Account-status	Maint-ppl, credits, installment
Adult	C1	Education, Country, Occupation, Work-class	Final-wgt, Cap-gain, Cap-loss
	C11	Country, Education, Occupation	Final-wgt, Cap-gain, Cap-loss
	C12	Country, Education, Occupation	Final-wgt, hr-per-week, Edu-no

Table 6: Information of dimensions and measures for cluster C1, C11 and C12 for each dataset

The automatic schema generation method gives two advantages for targeted knowledge discovery by reducing the number of dimension present in a data cluster and providing the significant measures present in a cluster. Moreover, it gives an abstract level relationship between the most influential numeric and nominal variables. For instance, the C12 cluster of the *Automobile* dataset highlights a relationship between *Make (model name)* and *Engine Type* variables with the *compression ratio (compression ratio)* and *price* of the automobile. Similarly, the *Purpose of credit* and *credit history* variables can be used to explore the credit amount and residential years from the credit cards data.

The *Adult* dataset suggests the prominent dimensions of *country*, *education* and *occupation*, with cluster C12 predicting a strong association between the *hours per week (hrs-per-week)* variable with prominent dimensions in this cluster. Data warehouse designers and knowledge workers can use this information as a starting point to explore further and extract hidden patterns from the data using OLAP functions. To perform the OLAP functions, we constructed data cubes for each cluster based on the identified dimensions and measures. With minimum number of dimensions and measures, our methodology limits the large number of views to be materialized and also provides narrow search space for the rapid discovery of knowledge from the underlying data. In each step of the proposed methodology, we provide step-by-step guidance to the data warehouse designers and knowledge workers to find useful knowledge. The set of tools, techniques and integration methods assist in recognizing complex and large data sets. In this work, we intend to offset the individual weaknesses of existing knowledge discovery methods and techniques and integrate the strengths tightly in a logical way. The proposed work can be applied in many areas where knowledge workers deal with complex datasets and have very limited knowledge of the domain. Data warehouse designers rely heavily on the user requirements and domain experts for modelling the warehouse schema. Because user requirements are unpredictable and constantly change with time, a design based solely on such requirements is unstable and poor choice of dimension and measures can intensively affect the knowledge discovery process and quality of decisions. Furthermore, even domain experts ignore the semantic relationships among data variables as they cannot be identified without the assistance of automated techniques.

It is apparent from the case studies that the proposed methodology helps in recognizing the predominant variables and finding their hidden relationships at every step. The identified patterns and relationship information has the potential to lead to the discovery of knowledge.

7 Comparison of Proposed Automated Analysis with Manual Analysis

In this section we give a general discussion on the proposed system and compare it with the traditional method of manual data analysis. It is important to clarify that the proposed integrated system doesn't solve a classical data mining or data analysis problem so there is no point to show that the system is able to complete a mining or analysis task on some standard dataset.

Moreover, we emphasize that our proposed methodology is suitable in cases where very limited domain knowledge exist and analysts depend on the systems to guide him/her in discovering knowledge. In our proposed work, we start the automated analysis by performing hierarchical clustering. It is assumed that data miners have a good idea of the important variables on basis of which clustering should be performed and later they interpret the clustering results. Also, based on their domain knowledge they extract the number of clusters which seem to be adequate for analysis. In high dimensional data, it is difficult for the human analyst to come up with important dimensions on the basis of which clustering is performed. Moreover, it is not feasible to rely heavily on limited domain knowledge to decide the number of clusters adequate for discovering knowledge.

In our work we automate this process by taking all the numeric variables into account for clustering and later perform statistical analysis to highlight the important variables in each cluster. Furthermore, we help in identifying the number of clusters to be taken for analysis by automatically calculating the cut-off point in the hierarchy. Likewise, we automate the manual way of identifying and by visualizing nominal variables in each cluster via multi-dimensional scaling technique. Additionally, we provide a generic way to assign nominal values in various groups which are near to impossible for the human analyst to visualize and create manually. These data guided numeric and nominal variables and their groupings information assist the human data warehouse designers to use this information to design a multi-dimensional schema for analytical analysis.

Without the presence of this automated system, data warehouse designers have to rely heavily on domain knowledge to manually model dimensions and dimensional hierarchies. Moreover, the manually modelled schema is unable to highlight the natural grouping of values which are interesting and worth exploring using OLAP tool. For instance, *Country* could be taken as a dimension by human data warehouse designer and one meaningful way of grouping countries is to assign countries based on their regions such as *Asia*, *Europe*, and *Africa*. However, the automated way proposed in this paper, *Country* dimension can form natural groups and these groups show the semantic relationships. For example, in an automated way of grouping *Australia*, *Mexico* and *Spain* could be together to highlight strong relations (i.e. almost same GDP) in underlying data. This interesting information is not obvious in manually modelled schema and will take unnecessary time to discover it manual inspection of OLAP cubes.

We believe that the proposed system facilitates a broad range of users (data warehouse designers, data miners, analysts) as different users have diverse analytical needs. For instance, a data miner may be interested in finding natural grouping (clusters) of data whereas the warehouse designer is more interested in finding important dimensions and measures in order to design a multi-dimensional scheme and may not be interested in knowing the natural clusters that exist in the data. It shows that certain information which appears to be

knowledge for one type of user may not appear the same for the other. It is near to impossible to discover knowledge without the automated aid provided by integrating the strengths of data mining and warehousing technologies. However, we do not challenge the existing manual usage of integrated techniques and emphasize that our proposal should be taken as a complementary method to assist knowledge discovery.

8 Conclusion and Future Work

In this paper, we have presented a generic methodology for the seamless integration of data mining and data warehousing with flexibility and efficiency objectives in consideration. In particular, we employed hierarchical clustering and multidimensional scaling technique for better understanding and efficient analysis of datasets. Moreover, the generic methods are provided to use the clustering results for the automatic generation of binary tree, which provides rich information of data variables at different levels of data abstraction. This rich information not only helps users to identify clusters of interest, but also highlights the semantic relationships and associations between the numeric and nominal variables within a cluster. These clusters and their data association information guide the human data warehouse developers to select the best possible set of dimensions and measures for the multidimensional model construction. We also propose an automated method for the generation of multidimensional schema to construct compact and informative data cubes. The case studies performed on real world datasets validated our proposal and elaborated its significance at various stages of knowledge discovery process. Results show that a hybrid solution complemented by automated methods seamlessly enhances the knowledge discovery process. In the end, we discussed the applicability of the proposed methodology in those domains where the data warehouse designers need automated assistance to design the schema and knowledge workers need data drive assistance and flexibility to explore complex dataset to find knowledge.

The future work is mainly focused on overcoming the existing limitations of the methodology. For instance, we intend to use more statistical functions such as entropy and information gain to rank dimensions and measures in the data cube. In addition to this, we are in a process of applying association rule mining on our generated schema to produce association rules and compare them with rules produced without schema (flat-file).

9 References

- AHMAD, A. & DEY, L. 2007. A k-mean clustering algorithm for mixed numeric and categorical data. *Data & Knowledge Engineering*, 63, 503-527.
- ASUNCION, A. & NEWMAN, D. J. 2010. UCI machine learning repository <http://archive.ics.uci.edu/ml> Irvine, CA: University of California, School of Information and Computer Science.
- BASAK, J. & KRISHNAPURAM, R. 2005. Interpretable hierarchical clustering by constructing an unsupervised decision tree. *IEEE Transactions on Knowledge and Data Engineering*, 121-132.
- BERNSTEIN, A., PROVOST, F. & HILL, S. 2005. Toward intelligent assistance for a data mining process: An ontology-based approach for cost-sensitive classification. *Knowledge and Data Engineering, IEEE Transactions on*, 17, 503-518.
- CORDES, D., HAUGHTON, V., CAREW, J. D., ARFANAKIS, K. & MARAVILLA, K. 2002. Hierarchical clustering to measure connectivity in fMRI resting-state data. *Magnetic resonance imaging*, 20, 305-317.
- FABRIS, C. C. & FREITAS, A. A. Incorporating deviation-detection functionality into the OLAP paradigm. the 16th Brazilian Symposium on Databases (SBBD 2001), 2001 Rio de Janeiro, Brazil. 274-285.
- GOIL, S. & CHOUDHARY, A. 1997. High performance OLAP and data mining on parallel computers. *Data Mining and Knowledge Discovery*, 1, 391-417.
- GOIL, S. & CHOUDHARY, A. A parallel scalable infrastructure for OLAP and data mining. 1999. Published by the IEEE Computer Society, 178.
- GOIL, S. & CHOUDHARY, A. 2001. PARSIMONY: An infrastructure for parallel multidimensional analysis and data mining. *Journal of parallel and distributed computing*, 61, 285-321.
- HAN, J. 1998. Towards on-line analytical mining in large databases. *ACM Sigmod Record*, 27, 97-107.
- HAN, J. & KAMBER, M. 2006. *Data mining: concepts and techniques*, Morgan Kaufmann.
- HOFMANN, H. 1994. Statlog (GermanCreditData) [Online]<http://archive.ics.uci.edu/ml/datasets/Statlog>
- KAMBER, M., HAN, J. & CHIANG, J. Y. Metarule-guided mining of multi-dimensional association rules using data cubes. *KDD'97*, 1997. 207-210.
- KOHAVI, R. & BECKER, B. 1996. *Adult dataset* [Online] Available: <http://archive.ics.uci.edu/ml/datasets/Adult>.
- LI, C. & BISWAS, G. 2002. Unsupervised learning with mixed numeric and nominal data. *IEEE Transactions on Knowledge and Data Engineering*, 673-690.
- LIU, Z. & GUO, M. 2001. A proposal of integrating data mining and on-line analytical processing in data warehouse. *Info-tech and Info-net*, 2001. *Proceedings. ICHI 2001 - Beijing*, 2001 International Conferences on Beijing, China.
- MESSAOUD, R. B., RABASÉDA, S. L., BOUSSAID, O. & MISSAOUI, R. Enhanced mining of association rules from data cubes. *DOLAP '06 Proceedings of the 9th ACM international workshop on Data warehousing and OLAP 2006* New York. ACM, 11-18.
- MISSAOUI, R., JATTEAU, G., BOUJENOUI, A. & NAOUALI, S. 2007. Toward Integrating Data Warehousing with Data Mining Techniques. *Data warehouses and OLAP: concepts, architectures, and solutions*, 253.

- NGUYEN, T. M., TJOA, A. M. & TRUJILLO, J. 2005. Data warehousing and knowledge discovery: A chronological view of research challenges. *Data warehousing and knowledge discovery*, 530-535.
- PARDILLO, J., ZUBCOFF, J., MAZÓN, J. N. & TRUJILLO, J. 2008. Applying MDA to integrate mining techniques into data warehouses: a time series case study. *Mining Multiple Information Sources MMIS 08*. Las Vegas.
- PIGHIN, M. & IERONUTTI, L. 2008. A Methodology Supporting the Design and Evaluating the Final Quality of Data Warehouses. *International Journal of Data Warehousing and Mining (IJDWM)*, 4, 15-34.
- RIVIÉRE, C. & MARLAIR, G. 2010. The use of multiple correspondence analysis and hierarchical clustering to identify incident typologies pertaining to the biofuel industry. *Biofuels, Bioproducts and Biorefining*, 4, 53-65.
- ROSARIO, G. E., RUNDENSTEINER, E. A., BROWN, D. C., WARD, M. O. & HUANG, S. 2004. Mapping nominal values to numbers for effective visualization. *Information Visualization*, 3, 80-95.
- SCHLIMMER, J. C. 1985. *Automobile dataset* [Online]. Available: <http://archive.ics.uci.edu/ml/datasets/Automobile>
- SEO, J. & GORDISH-DRESSMAN, H. 2007. Exploratory data analysis with categorical variables: An improved rank-by-feature framework and a case study. *International Journal of Human-Computer Interaction*, 23, 287-314.
- SEO, J. & SHNEIDERMAN, B. 2006. Knowledge discovery in high-dimensional data: Case studies and a user survey for the rank-by-feature framework. *IEEE Transactions on Visualization and Computer Graphics*, 311-322.
- USMAN, M. & ASGHAR, S. 2011. An Architecture for Integrated Online Analytical Mining. *Journal of Emerging Technologies in Web Intelligence*, 3, 74-99.
- USMAN, M., ASGHAR, S. & FONG, S. 2010. Integrated Performance and Visualization Enhancement of OLAP Using Growing Self Organizing Neural Networks. *Journal of Advances in Information Technology*, 1, 26-37.
- USMAN, M. & PEARS, R. 2010. Integration of Data Mining and Data Warehousing: A Practical Methodology. *International Journal of Advancements in Computing Technology*, 2, 31 - 46.
- USMAN, M. & PEARS, R. 2011. Multi Level Mining of Warehouse Schema. Networked Digital Technologies, *Communications in Computer and Information Science*. Springer Berlin Heidelberg. 136:395-408
- ZUBCOFF, J., PARDILLO, J. & TRUJILLO, J. 2007. Integrating clustering data mining into the multidimensional modeling of data warehouses with UML profiles. *Data Warehousing and Knowledge Discovery*, 199-208.

Author Index

- Abawajy, Jemal, 93
 Adly, Amr, 119
- Bahgat, Roba, 119
 Brankovic, Ljiljana, 139
 Bui, Dang Bach, 109
- Calvo, Rafael, 103
 Cao, Longbing, 5
 Catchpoole, Daniel, 53
 Chakrabarti, Partha Pratim, 209
 Chawla, Sanjay, 3
 Chen, Kuei-Ling B, 221
 Chetty, Girija, 157
 Christen, Peter, iii, 127
- Deng, Guang-Feng, 221
 Djuana, Endang, 183
 Dobbie, Gillian, 167
- Feng, David, 21
 Fong, A C M, 229
- Ghous, Hamid, 53
 Giggins, Helen, 139
 Guan, Genliang, 21
- Hadzic, Fedja, 109
 Hafeez, Mohsin, 199
 Hecker, Michael, 109
 Ho, Nicholas, 53
 Hung, Yu-Shiang, 221
 Hussain, Md. Sazzad, 103
 Hutter, Marcus, 79
- Islam, Md Zahidul, 27
 Islam, Md. Zahidul, 199
- Jelinek, Herbert, 93
- Kelarev, Andrei, 93
 Kennedy, Paul J, iii, 53, 175
 Khan, Mahmood, 199
 Koh, Yun Sing, 167
- Li, Jialing, 191
 Li, Jiuyong, iii
 Li, Li, 191
 Li, Ming, 9
 Li, Yuefeng, 183
- Li, Zhe, 21
 Li, Zhijie, 9
 Liao, Jianwei, 191
 Liu, Derek, 9
- Mayo, Michael, 149
 Meng, Qinxue, 175
 Monkaresi, Hamed, 103
 Muhammad Fuad, Muhammad Marwan, 85
- Naiwala, Chandrasiri, 61
- Ong, Kok-Leong, 9
- Pears, Russel, 229
- Rahman, Md Anisur, 27
 Ritz, Christian, 71
- Sammut, Claude, 71
 Sarkar, Sudeshna, 209
 Shao, Wen, 79
 Singh, Lavneet, 157
 Soliman, Omar S, 119
 Stirling, David, 71
 Stranieri, Andrew, 93
 Sun, Chao, 71
 Sunehag, Peter, 79
- Taha, Kamal, 43
 Ting, Kai Ming, 61
- Usman, Muhammad, 229
- Vadlamudi, Satya Gautam, 209
 Vatsalan, Dinusha, 127
- Wang, Zhiyong, 21
 Wells, Jonathan, 61
 Wen, Xiao, 191
- Xu, Yue, 183
- Yang, Chi-Ta, 221
 Yearwood, John, 93
 Yu, Kaimin, 21
- Zhang, Jing, 9
 Zhao, Yanchang, iii

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 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, 17-20 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 Kerry Butler-Henderson, Curtin University, Australia and Tony Sahama, Queensland 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 121 - Data Mining and Analytics 2011**
Edited by Peter Vamplew, University of Ballarat, Australia, Andrew Stranieri, University of Ballarat, Australia, Kok-Leong Ong, Deakin University, Australia, Peter Christen, Australian National University, Australia and Paul J. Kennedy, University of Technology, Sydney, Australia. December 2011. 978-1-921770-02-9.
Contains the proceedings of the Ninth Australasian Data Mining Conference (AusDM'11), Ballarat, Australia, 1-2 December 2011.
- Volume 122 - Computer Science 2012**
Edited by Mark Reynolds, The University of Western Australia, Australia and Bruce Thomas, University of South Australia. January 2012. 978-1-921770-03-6.
Contains the proceedings of the Thirty-Fifth Australasian Computer Science Conference (ACSC 2012), Melbourne, Australia, 30 January – 3 February 2012.
- Volume 123 - Computing Education 2012**
Edited by Michael de Raadt, Moodle Pty Ltd and Angela Carbone, Monash University, Australia. January 2012. 978-1-921770-04-3.
Contains the proceedings of the Fourteenth Australasian Computing Education Conference (ACE 2012), Melbourne, Australia, 30 January – 3 February 2012.
- Volume 124 - Database Technologies 2012**
Edited by Rui Zhang, The University of Melbourne, Australia and Yanchun Zhang, Victoria University, Australia. January 2012. 978-1-920682-95-8.
Contains the proceedings of the Twenty-Third Australasian Database Conference (ADC 2012), Melbourne, Australia, 30 January – 3 February 2012.
- Volume 125 - Information Security 2012**
Edited by Josef Pieprzyk, Macquarie University, Australia and Clark Thomborson, The University of Auckland, New Zealand. January 2012. 978-1-921770-06-7.
Contains the proceedings of the Tenth Australasian Information Security Conference (AISC 2012), Melbourne, Australia, 30 January – 3 February 2012.
- Volume 126 - User Interfaces 2012**
Edited by Haifeng Shen, Flinders University, Australia and Ross T. Smith, University of South Australia, Australia. January 2012. 978-1-921770-07-4.
Contains the proceedings of the Thirteenth Australasian User Interface Conference (AUIC2012), Melbourne, Australia, 30 January – 3 February 2012.
- Volume 127 - Parallel and Distributed Computing 2012**
Edited by Jinjun Chen, University of Technology, Sydney, Australia and Rajiv Ranjan, CSIRO ICT Centre, Australia. January 2012. 978-1-921770-08-1.
Contains the proceedings of the Tenth Australasian Symposium on Parallel and Distributed Computing (AusPDC 2012), Melbourne, Australia, 30 January – 3 February 2012.
- Volume 128 - Theory of Computing 2012**
Edited by Julián Mestre, University of Sydney, Australia. January 2012. 978-1-921770-09-8.
Contains the proceedings of the Eighteenth Computing: The Australasian Theory Symposium (CATS 2012), Melbourne, Australia, 30 January – 3 February 2012.
- Volume 129 - Health Informatics and Knowledge Management 2012**
Edited by Kerry Butler-Henderson, Curtin University, Australia and Kathleen Gray, University of Melbourne, Australia. January 2012. 978-1-921770-10-4.
Contains the proceedings of the Fifth Australasian Workshop on Health Informatics and Knowledge Management (HIKM 2012), Melbourne, Australia, 30 January – 3 February 2012.
- Volume 130 - Conceptual Modelling 2012**
Edited by Aditya Ghose, University of Wollongong, Australia and Flavio Ferrarotti, Victoria University of Wellington, New Zealand. January 2012. 978-1-921770-11-1.
Contains the proceedings of the Eighth Asia-Pacific Conference on Conceptual Modelling (APCCM 2012), Melbourne, Australia, 31 January – 3 February 2012.
- Volume 131 - 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.