Conferences in Research and Practice in
Information Technology

Volume 170

Data Mining and Analytics 2016 (AusDM 2016)

,

Data Mining and Analytics 2016

Proceedings of the
Fourteenth Australasian Data Mining Conference
(AusDM 2016), Canberra, Australia,
6–8 December 2016

Yanchang Zhao, Md Zahid Islam, Glenn Stone,
Kok-Leong Ong, Dharmendra Sharma and Graham Williams,
Eds.

Volume 170 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 2016. Proceedings of the Fourteenth Australasian Data Mining Conference (AusDM 2016), Canberra, Australia, 6–8 December 2016

Conferences in Research and Practice in Information Technology, Volume 170.

Editors:

Yanchang Zhao
Department of Immigration and Border Protection and RDataMining.com
5 Chan St
Belconnen, ACT 2617, Australia
Email: yanchang@rdatamining.com

Md Zahid Islam
School of Computing and Mathematics
Faculty of Business
Charles Sturt University
Bathurst, NSW 2795, Australia
Email: zislam@csu.edu.au

Glenn Stone
School of Computing, Engineering and Mathematics
Western Sydney University
Locked Bag 1797
Penrith NSW 2751, Australia
Email: g.stone@westernsydney.edu.au

Kok-Leong Ong
La Trobe Business School
College of Arts, Social Sciences and Commerce
La Trobe University
P.O.Box 821, Wodonga Victoria 3689, Australia
Email: kok-leong.ong@latrobe.edu.au

Dharmendra Sharma
Faculty of Information Sciences and Engineering
University of Canberra
Bruce ACT 2617 Australia
Email: Dharmendra.Sharma@canberra.edu.au

Graham Williams
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399, USA
Email: Graham.Williams@togaware.com

Series Editors:
Vladimir Estivill-Castro, Griffith University, Queensland
Simeon J. Simoff, Western Sydney University, NSW
Email:crpit@scem.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 170.
ISSN 1445-1336.
ISBN 978-1-921770-50-0.

Document engineering by CRPIT, December 2016.

Table of Contents

Industry Showcases

Message from the Conference Chairs

On behalf of the Australasian Data Mining 2016 Organising Committee we welcome you to Canberra and to the 14th Australasian Data Mining Conference. Canberra saw the very first AusDM in 2003 and the conference has since been held in many locations and has been affiliated with many other conferences over the years. This year we take the opportunity to hold the conference jointly with the 23rd Australian Statistical Conference. Holding our conferences together allows our communities to interact and to share and learn from one another. It is tremendous to see the thriving growth of Data Mining and Statistics, both under the Data Science umbrella, as we move into an era where the importance of data is so widely recognised. Data Science's time has come and is today recognised as foundational for every organisation on the planet today. They are all on a journey to realising that the analysis of data will empower their business into the future – a challenge to computer science, statisticians and the wider data science research and innovation community – a challenge requiring new foundations for industry and government.

Data Mining continues as the core underlying this significant and growing interest in data, and not least in the Australian community. AusDM offers an opportunity to be exposed to some of the latest research in data mining and to hear of experiences in delivering data mining within industry and government and to network with the researchers and practitioners.

This year you will find coverage of topics ranging from predictive analytics, visualisation, social and network analysis, unsupervised methods, data mining applications, big data, an industry showcase and concluding with a panel discussion on "Opportunities and challenges for statisticians and data miners in the emerging field of business analytics."

This year we also see a new focus on the Government sector with the significantly increasing recognition by all government departments that analytics is at the heart of their business. Specific streams through the conference will be particularly pertinent to the practising data scientist. Delivering AusDM in conjunction and seamlessly with the Australian Statistical Conference will offer unsurpassed access to not only data mining but also statistical developments that will have impact on business.

Such a conference involves many people organising many different aspects. As the conference general chairs we acknowledge and are grateful for the tremendous support and effort of the organising committee and the program committee. Thank you for bringing together a strong program and, we are sure, a great experience for the attendees. Of course the authors and presenters and keynotes are what makes the conference and without your support there would be no conference. Thank you for again choosing to present your latest research and achievements with the AusDM community.

Enjoy the conference. Enjoy learning of new developments in the data sciences. Enjoy gaining a glimpse into the future. Enjoy a future where data drives the intelligent applications that support all aspects of our lives.

Yours Sincerely,

Dharmendra Sharma
University of Canberra

Graham Williams
Microsoft

November 2016

Message from the Program Chairs

Welcome to the 14th Australasian Data Mining Conference in Canberra, Australia.

A total of fifty one (51) papers were submitted to the two main conference tracks (Research Track and Application Track). From these, each paper was rigorously reviewed by three or four reviewers, providing an assessment of the papers' merits. After careful consideration, twenty three (23) papers were selected for inclusion in the final conference proceedings, with an overall acceptance rate of 45%.

In addition, there is a new Industry Showcase Track this year, where, based on evaluation by industry experts, four (4) submissions have been accepted for presentation at the conference.

Our Program Committee members have been pivotal to the success of this conference. Many have worked hard to provide timely reviews that are crucial to ensuring the success of the conference. On behalf of the entire Organising Committee, we express our appreciation to the Program Committee for their cooperative spirit and extraordinary effort. Many members delivered every review requested, and more. It was a true privilege to work with such a dedicated and focused team, many of whom were also active in helping with the publicity of the conference. We also wish to extend our appreciation to all external reviewers relied upon by the Program Committee members; they have played a part of making this conference possible.

Beyond the technical program in the proceedings, the conference has been enriched by many other items. These include the co-location the 23rd Australian Statistical Conference (ASC 2016) and the 9th Australian Conference on Teaching Statistics (OZCOTS 2016), and the availability of keynote speakers from the AusDM, ASC and OZCOTS conferences. We trust these programmes will provide insightful new research ideas and directions.

Lastly, we hope you enjoy the conference as much as we have enjoyed being part of delivering it.

Yours Sincerely,

Yanchang Zhao
Department of Immigration and Border Protection, Australia

Md Zahid Islam
Charles Sturt University

Glenn Stone
Western Sydney University

Kok-Leong Ong
La Trobe University

Warwick Graco
Australian Taxation Office

November 2016

Conference Organisation

Conference Chairs

Dharmendra Sharma, University of Canberra
Graham Williams, Microsoft

Program Chairs (Research Track)

Md Zahid Islam, Charles Sturt University
Glenn Stone, Western Sydney University

Program Chairs (Application Track)

Kok-Leong Ong, La Trobe University, Melbourne
Yanchang Zhao, Department of Immigration and Border Protection, Australia

Program Chair (Industry Showcase)

Warwick Graco, Australian Taxation Office

Tutorial Chair

Andrew Stranieri, Federation University Australia, Mount Helen

Web and Publicity Chair

Ling Chen, University of Technology Sydney

Competition Chairs

Tony Nolan, Australian Taxation Office
Paul Kennedy, University of Technology Sydney

Steering Committee Chairs

Simeon Simoff, Western Sydney University
Graham Williams, Microsoft

Steering Committee Members

Peter Christen, Australian National University, Canberra
Ling Chen, University of Technology Sydney
Md Zahid Islam, Charles Sturt University
Paul Kennedy, University of Technology Sydney
Jiuyong (John) Li, University of South Australia, Adelaide
Kok-Leong Ong, La Trobe University, Melbourne
John Roddick, Flinders University, Adelaide
Andrew Stranieri, Federation University Australia, Mount Helen
Geoff Webb, Monash University, Melbourne
Richi Nayak, Queensland University of Technology, Brisbane
Yanchang Zhao, Department of Immigration and Border Protection and RDataMining.com

Program Committee

Research Track

Adil Baghirov, Federation University
Yee Ling Boo, RMIT University
Jie Chen, The University of South Australia
Xuan-Hong Dang, University of California at Santa Barbara
Rafiqul Islam, Charles Sturt University
Muhammad Marwan Muhammad Fuad, Aarhus University
Yun Sing Koh, University of Auckland
Wei Kang, University of South Australia
Paul Kwan, University of New England
Gang Li, Deakin University
Lin Liu, University of South Australia
Brad Malin, Vanderbilt University
Qinxue Meng, University of Technology Sydney
Parma Nand, Auckland University of Technology
Quang Vinh Nguyen, Western Sydney University
Azizur Rahman, Charles Sturt University
Md Anisur Rahman, Charles Sturt University
Md Geaur Rahman, Charles Sturt University
David Stirling, University of Wollongong
Siamak Tafavogh, Coca Cola Amatil and University of Technology Sydney
Xiaohui Tao, University of Southern Queensland
Dat Tran, University of Canberra
Dinusha Vatsalan, Australian National University
Sitalakshmi Venkatraman, Melbourne Polytechnic
Guandong Xu, University of Technology Sydney
Ji Zhang, University of Southern Queensland
Mengjie Zhang, Victoria University of Wellington

Application Track

Alex Antic, PwC
Chris Barnes, University of Canberra
Nathan Brewer, Department of Human Services
Neil Brittliff, University of Canberra
Adriel Cheng, Defence Science and Technology Organisation
Lianhua Chi, IBM Research Australia
Tania Churchill, Australian Government
Ross Farrelly, Datamilk
Lifang Gu, Australian Taxation Office
Yingsong Hu, Australian Government
Edward Kang, Australian Government
Luke Lake, Department of Human Services
Fangfang Li, Fairfax Media
Geng Li, Apple Inc.
Jin Li, Geoscience Australia
Balapuwaduge Sumudu Udaya Mendis, Australian National University
Kee Siong Ng, AUSTRAC
Tom Osborn, University of Technology Sydney
Martin Rennhackkamp, PBT Group Australia
Nandita Sharma, Australian Taxation Office
Ting Yu, Commonwealth Bank of Australia

Industry Showcase Track

Rohan Baxter, Australian Taxation Office
Klaus Felsche, C21 Directions
Ray Lindsay, Australian Taxation Office
David Stirling, University Of Wollongong

AusDM Sponsors

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

http://www.togaware.com

http://www.westernsydney.edu.au/

Keynotes

Opinion Search Engine

Xue Li

School of Information Technology and Electrical Engineering
The University of Queensland, Australia

xueli@itee.uq.edu.au

Abstract

Social media and social networks are used as a platform for people to share their experiences such as information and opinions about services and consumer products, or to organize and initiate social events. Some questions would be asked in order to understand the social media and social networks: how can we detect and predict the emerging events? How can we understand people's opinions about a current issue, a new product, or an important event? How can we predict the spatial and temporal propagation patterns of online-discussed issues? This talk is to introduce our research work on a social opinion search engine software that is designed to answer the above questions.

Key words: Sentiment analysis, Language modelling, Spatial and temporal word spectrum, Network modality, Graph classification.

1 Introduction

Social opinions may reflect or affect organizational performance. In order to make social opinions searchable and relevant to the organizations who would actively search for them, we have developed a series algorithms and a software tool named, Opinion Search Engine (OSE) for analysing and visualising social media with respect to the organizational performance data. Our perception is that an organization would have its performance data available for prediction and the performance of the organization may be affected by social opinions. According to the reports of PEW Research Centre, the social opinions are now reflected by social media in many aspects of social opinions: trends of pop music, fashions, politics, financial markets, nature disaster responses, sales of products and services. For example, in government elections, people's feelings may affect the swings of votes of political parties; in real estate business, people's attitudes may cause changes to local house pricing; in stock market, people's moods may influence the values of stock shares. Therefore, we need a tool to predict the trend of changes of the organizational performances based on the social opinions that would cause the changes.

2 System Description

Our OSE framework has a few key components which are designed to understand the social opinions with respect to the given objects. In our system, the objects can be anything that the social network users would talk about.

2.1 Function of OSE

In Google Search Engine, the *Big Table* is used as a structure to store the entire WWW. Similarly, we use a Big O-Table to implement a 5-tuple function (***f***) to store the entire social opinions from all social communities. **Opinion Search Engine** (OSE) is implemented as a constrained function (***f***) for finding opinions (instead of finding Web contents):

$$f_l^t(x, y) = z$$

- ***x*** is a set of Social Network Users (**Who**).
- ***y*** is a set of target objects (**What** - Topics, organizations, Product & Services, Events, ...).
- ***z***is a set of opinions (**How** - Positive, Negative, Neutral, Like, Love, Hate, etc).
- ***t*** is a constraint about time point or time period (**When**).
- ***l*** is a constraint about a geo-location (**Where** - an area, a city, a state, ...).

In above, ***x*** can be structured as a graph of **social communities** (Retweeting vs. Friend-of-Friend/follower networks);***y*** can be used to narrow down (partition) the **community graph** of ***x***; ***t*** can be in a time point or a time range; ***l*** can be an area of location on a map. The data for Big O-Table is generated by using a set of machine learning algorithms. Initially a crawler is used to obtain the messages posted on the social networks. Then the classification algorithms are used to obtain the values of function (***f***). For example, there are only about 2% twitters that have the geo-locations available. In order to avoid the sparsity of data, we developed algorithms to reason out the locations for the posted social media messages (Unankard*et al*, 2014a & 2014b).

2.2 Unique Language Model

One of the key techniques of OSE is its usage of unique language models (ULM) for organizations that are talked about by social network users. The idea of ULM is from language modelling in computational

linguistics. We use a temporal language model to quantify the co-occurrences and changes of all terms related to an organization along a timeline. A unique hierarchical language model (ULM) can be established for an organization at a certain time period. In our tests, we achieved the excellent results in the prediction of the Australian Federal Election and the accurate prediction of the Queensland 2015 State Election, as well as the NSW 2015 State Election. As we can always establish a temporal unique language model for a given organization, our approach opens a new way of performance prediction based on social networks. For example, in analysing social opinions for two competitive organizations, ULMs can help decide if an opinion is positive towards one and negative to the other, or both positive, and vice versa.

2.3 Social Communities Profiling

Social network users can be grouped into social communities. When opinions are searched, social communities can be used as an important source to provide comprehensive and quantifiable opinions. Therefore, we need to constantly identify and maintain the dynamics of social communities about their commons, outliers, opinion propagation patterns, and power of influences. We use our ***f*** function (described in Section 2.1) as the input to create a PTO (People-Topic-Opinion) graph for every community that can be identified. There are two perspectives on PTO: for a given topic, we can find the social communities that are related to this topic. Also, for a given community, we can find the topics that it has interests. A PTO graph is then becoming a pool of social opinions. When a predicting task is requested, all PTO graphs will be retrieved in time for the regression-base prediction of the performance based on the combination of the ULM data.

2.4 Spatial-Temporal Spectrum of Social Media

In social networks, different users from different geolocations will post different microblog messages for their local issues on social networks. We invented algorithms to calculate unique features of social media for different geolocations in different time periods. The outcome of this calculation is called a Spatial-Temporal Word Spectrum (STWS) model which is a linguistic ***fingerprint*** of a geolocation on social media (Li, 2015). We use STWS as a baseline to catch the prominent and statistical features of microblogs as a spectral representation of the words used by social network users. As a baseline of the social media, STWS can be used in three aspects: (1) to detect emerging local events; (2) to guess the location of a user; and (3) to detect sentiment of users toward a target object (e.g., performance indicators of an organization, a product or service).

2.5 Drivers of the visualisation

In order to visualise the trends of opinions, we need to know **when, where, what** and **who** are involved in the social media, see Fig. 1. So our framework can be used to visualise the social opinions. The visualisation is presented in terms of six drivers that are controllable by the end user in terms of their properties.

- **Location-Driven**: user can click on the map to select various locations worldwide.
- **Time-Driven:** user can specify the time range for a given location.
- **Event-Driven**: user can specify an event that has happened or is happening related to the organization.
- **Network-Driven**: user can select groups of people from social networks, such as high school students, university students, etc. to see the social communities related to the organization.
- **Rank-Driven:** user can specify ranking criteria of interest, such as the most popular, most interesting, most expensive, the longest times, etc.
- **Content-Driven:** user can select content that s/he is interested in, such as a topic discussed on social networks, an event that is unfolding, etc. Big data fusion techniques are used for the real-time prediction. It offers a comprehensive bird's-eye-view on everything that is happening on the social networks.

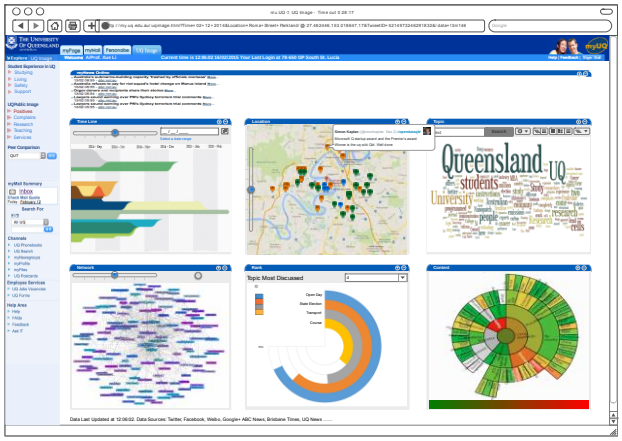


Fig. 1. Dashboard of the Six Drivers for Social Media Visualization

2.6 A three-layer System Architecture

We designed a three-layer architecture to accommodate all the algorithms in our framework, see Fig. 2. Firstly, an opinion-seeker renders a set of key terms that are used to best describe an object, to the system. The performance data of an organization will be used reflect the changes according to online users' opinions. The system is also constantly crawling

social networks (e.g., Twitter or a review report). A large in-house database of up-to-date social media data (e.g., tweets) must be available in the system. At Content Layer, the social network content will be cross-indexed with ability to efficiently access tweets in all different ways. At Network Layer, a graph of the social-network structure is constructed, where the nodes are the social-network users plus their profiles, and the edges represent the relationships between followers. Social communities will be revealed in this way. The statistical information about each social-network user's behaviours will also be recorded. At Spatial-Temporal Layer, a spatial-temporal index is maintained for accessing the results of opinion analysis. As a result, the public social networks will be mirrored by our three-layer data sets and be used to provide information about **what, who, where** and **when** opinions are given.

3 Prediction Models for Different Domain Applications

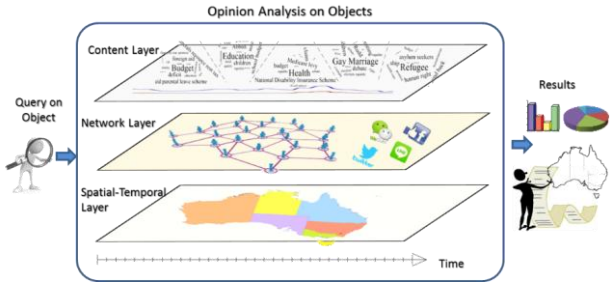


Fig. 2. A three-layer architecture of social media analysis

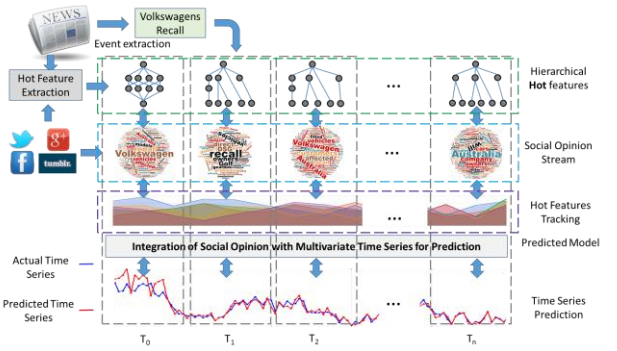


Fig. 3. The integration of social opinions with multivariate time series for prediction.

OSE system can be integrated as a transparent API within a given website to collect social opinions for the social feedback of an organisation. When social opinions are detected with spatial and temporal information, theperformance of an organisation measured by time series can then be predicted with respect to the changescaused by changes of social opinions, i.e. people's feelings, attitudes, and moods. A variety of applications can bebenefited by using this methodology, for example, market studies can be improved by early prediction on prices,or the implementation of government policies to do with environmental protection, immigration, and healthcare canbe better understoodby tracking the changes of social opinions expressed about them.

Performance Indicators (PI) of organisations can be monitored as time series. The PIs are the organizational properties inherent in the origin of the performance. For a given application, we firstly identify the PIs of this organization. Then the *hot* features of the object (e.g., an organization in the current context of discussion) will be discovered from the social media analysis. The *hot* features are those issues that social network users would care about and talk about. Apparently, in reality, PIs may be mismatched with the hot features. Our prediction models will be established based on the learnings for the causal relationships of the PIs and the hot features. The learning is evolutionary based on the streaming data. Fig. 3 shows the idea.

4 Spamming Detection

We consider that sentiment is reactive and opinion is proactive.

Based the models of our framework, we also developed the spamming detection algorithms to detect the social opinions that are untruthfully injected into the social media.

We assume that the spammers would have unique language models when they inject the spamming opinions into the social media. In addition to the traditional sentiment analysis using emotional words in libraries such as SentiWordNet, our ULM can be used toevaluate spamming opinions without detecting for emotional words.

We assume that when people express their opinions, they would choose some particular words to express their ideas in order to form up a specific concept or an attitude towards a specific object such as a product or service, a political party, or an organization. For example, in a political campaign, a certain political party may use a slogan: "Stop the boats", to rally the people who have the same opinion in a government election.

Fig. 4 shows the life cycle of a government election when people express the opinions at different time. When an event happens, people would start talk about it (Unankard, 2015).

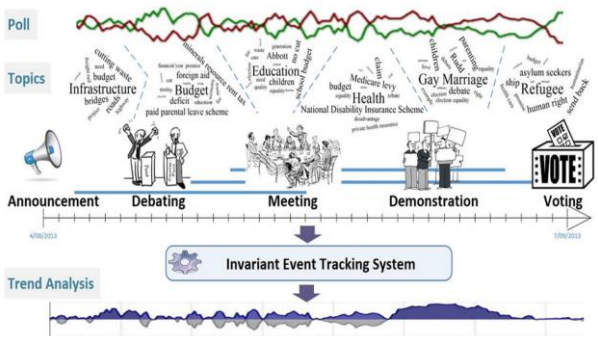


Fig. 4. Tracking different events during an election.

5 Conclusions

Opinion Search Engines provides a transparent plug-and-play software API for websites of organizations to care about the feedback of social opinions for its performances. In this talk we presented our design and implementation of such a system for obtaining social opinions to summarise and predict the performance of the organizations such as the government elections. Our experiences show that the data fusion and machine learning algorithms would play an important role.

6 Bibliography and References

Aggarwal C. C. and Subbian K. (2012). Event detection in social streams. In SDM, 624–635, 2012.

Andrienko, N. and Andrienko, G. (2006). Exploratory analysis of spatial and temporal data: a systematic approach. Springer Science & Business Media.

Bo, H. and P. C. T. Baldwin (2012). Geolocation prediction in social media data by finding location indicative words. Proceedings of COLING 2012: Technical Papers: 1045-1062.

Cambria, E. Schuller, B. Xia, Y. and Havasi, C. (2013). New Avenues in Opinion Mining and Sentiment Analysis, IEEE Intelligent Systems, vol.28, no. 2, pp. 15-21.

Cambria, E. Schuller, B. Liu, B. Wang, H. & Havasi, C. (2013). Knowledge-based approaches to concept-level sentiment analysis. IEEE Intelligent Systems, (2), 12-14.

Cambria, E. Wang, H. and White, B. (2014). Guest editorial: Big social data analysis. Knowledge-Based Systems, (69), 1-2.

Coulthard, M. (2004). Author identification, idiolect, and linguistic uniqueness. Applied linguistics, 25(4), 431-447.

Harrison, C. (2015). Word spectrum: Visualizing Google's BiGram Data. http://www.chrisharrison.net/index.php/Visualizations/WordSpectrum.

Google n-gram (2006). http://googleresearch.blogspot.com.au/2006/08/all-our-n-gram-are-belong-to-you.html.

Jalal, M., et al. (2014). Home location identification of Twitter users. ACM Trans. Intell. Syst. Technol. 5(3): 1-21.

Li, L. Goodchild, M. F. and Xu, B. (2013). Spatial, temporal, and socioeconomic patterns in the use of Twitter and Flickr. Cartography and geographic information science, 40(2), 61-77.

Li, X., *et al* (2015): Spatial and Temporal Word Spectrum of Social Media, SIGKDD 2015 Workshop WISDOM, Sydney, 2015, http://sentic.net/wisdom/2015/li.pdf

Liu, B. (2015). Sentiment analysis mining opinions, sentiments, and emotions, Cambridge University Press.

Stoica, P. and R. L. Moses (2005). Spectral analysis of signals, Pearson/Prentice Hall Upper Saddle River, NJ.

Stephen, R. et al. (2012). Supervised text-based geolocation using language models on an adaptive grid. Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning.

Unankard, S. Li, X. & Sharaf, M. A. (2014a). Emerging event detection in social networks with location sensitivity. World Wide Web Journal (WWWJ), 1-25.

Unankard, S. Li, X. Sharaf, M.A. Zhong, J. and Li, X. (2014b). Predicting elections from social networks based on sub-event detection and sentiment analysis, In WISE, (Web Information System Engineering), Part II, LNCS8787, 1-16.

Unankard, S. Li, X. and Long, G. (2015). Invariant event tracking on social networks. In Database Systems for Advanced Applications, DASFAA2015, 517-521, 2015.

Contributed Papers

Finding Influentials in Twitter: A Temporal Influence Ranking Model

Xingjun Ma[1,2] **Chunpin Li**[2] **James Bailey**[1] **Sudanthi Wijewickrema**[1]

[1] Department of Computing and Information Systems
The University of Melbourne,
Victoria 3010, Australia,
Email: {xingjunm@student., baileyj@, swijewickrem@}unimelb.edu.au

[2] School of Software
Tsinghua University,
Beijing 100084, China,
Email: cli@tsinghua.edu.cn

Abstract

With the growing popularity of online social media, identifying influential users in these social networks has become very popular. Existing works have studied user attributes, network structure and user interactions when measuring user influence. In contrast to these works, we focus on user behavioural characteristics. We investigate the temporal dynamics of user activity patterns and how these patterns affect user interactions. We assimilate such characteristics into a PageRank based temporal influence ranking model (TIR) to identify influential users. The transition probability in TIR is predicted by a logistic regression model and the random walk, biased according to users' temporal activity patterns. Experiments demonstrate that TIR has better performance and is more stable than the existing models in global influence ranking and friend recommendation.

Keywords: social networks, user influence, influence ranking, influentials, PageRank

1 Introduction

As one of the most popular online social networks (OSNs), Twitter allows users to share status, activities and ideas, and follow others for updates. With millions of closely connected users, it has changed the way people communicate and socialize, and has had prominent influence on important historical events, such as the 2016 US presidential election and the 2010 Arabic Spring. Opinion leaders, authorities, and media hubs (or influentials) play an important role in generating and propagating such influence. With a large number of followers, they can spread stories, ideas and product information to massive audiences and enable follow-up discussions. As such, identifying influentials and understanding why and how they influence others has become a hot topic in recent years.

However, identifying influentials from millions of users is a challenging task. First of all, the definition of influence varies according to context and application domain. For example, in viral marketing, a car company may regard authoritative car buyers or advisers as influential users, while a fashion company may be more interested in pop singers. Second, it is hard to find accurate influence measures. As people's interests and behaviours change over time, the influence of a user also varies from time to time.

Existing works in this area have studied user attributes, network structure, and user interactions when measuring user influence. Different from existing works, we focus on the temporal dynamics of user activity pat- terns and how these patterns affect user influence. In this paper, we make the following contributions: 1) we comprehensively investigate the user activity patterns and their influence on user interactions, 2) we incorporate the temporal user behaviour characteristics into a new influence ranking model to identify influential users and 3) we empirically demonstrate that our model has better performance and is more stable than the existing models in global influence ranking and friend recommendation.

The rest of the paper is organized as follows. We review related work in Section 2. Section 3 investigates user statistics, user activity patterns as well as user interaction (response) characteristics. We formulate the response probability between two users in Section 4. In Section 5, we introduce a novel influence ranking model based on the response probability. The proposed model is evaluated in Section 6. Section 7 concludes the paper and discusses future work.

2 Literature Review

In the areas of social science and communication theory, the study of people's individual influence in a community has been an intriguing topic since the 1940s. Lazarsfeld et al. (1948) introduced a two-step flow theory to formulate the part played by people in the flow of mass communications. The theory states that individuals receive media effects indirectly from opinion leaders rather than directly from mass media. In the following decades, opinion leaders or "influentials" became an important part of innovation diffusion (Rogers 1962), communication theory (McQuail 1987), and marketing (Chan & Misra 1990).

Modern views of user influence provide a more in-depth understanding of the diffusion process and interpersonal interactions. Watts & Dodds (2007) reported that large cascades of influence are driven not only by influentials but also by a critical mass of easily influenced individuals. In recent years, the rise of online social media has facilitated empirical exploration and validation of different influence theories.

Motivated by the design of viral marketing strategies and Domingos & Richardson (2001), Richardson & Domingos (2002) were the first to introduce influence modelling in online social networks from a data mining perspective. They proposed a probabilistic model to exploit the expected network value of users during the propagation of influence to others. Kempe et al. (2003) investigated the propagation process of user influence in two propagation models, i.e., Linear Threshold Model and Independent Cascade Model. This work provides the first provable approximation guarantees for greedy approximation algorithms in influence maximization problem. Follow-ups in this direction are Leskovec et al. (2007) and Chen et al. (2010).

Recent works quantify user influence based on user attributes, for example, the number of followers. Kwak et al. (2010) reported that there is a gap in influence inferred from the number of followers and that from the popularity of one's tweets. Cha et al. (2010) conducted an in-depth comparison of three measures of influence: indegreee (i.e., the number of followers), retweets, and mentions. They found that popular users who have high indegree are not necessarily influential, which indicates that the number of followers may not be a good measure of influence in Twitter. This conclusion is also validated in a recent study (Cataldi & Aufaure 2015). There are other works that utilize user attributes to measure user influence such as Leavitt et al. (2009). However, all these works demonstrate that using user attributes alone may not be accurate enough for user influence measurement.

Other works measure user influence by taking both the network structure and user interactions into consideration. One such work is TunkRank[1], a variant of PageRank (Page et al. 1999). The influence is calculated iteratively by the following equation:

$$Influence(X) = \sum_{Y \in Follwers(X)} \frac{1 + p * Influence(Y)}{|Friends(Y)|},$$

It measures a user's influence by the expected number of people who will read a tweet that X tweets, including all retweets of that tweet. $Friends(Y)$ is the set of users that Y follows while $Follwers(X)$ indicates the users that follow X. If Y reads a tweet from X, there is a constant probability p that Y will retweet it. However, by taking p a constant probability, TunkRank assumes the retweet probabilities are the same for any tweets between any users, which may not be a reliable assumption in reality.

Kwak et al. (2010) proposed a method to find influentials by considering both the network structure and the temporal order of information adoption. They assume that followers only adopt the information they are first exposed to and will ignore such information in later exposures. The temporal sequence of information adoption was also discussed in Bakshy et al. (2011) to calculate the influence score for a given URL post. Although these two works consider the chronological order of tweet diffusion, the temporal order in which information is received is not necessarily consistent with the order of information adoption. For example, users would probably read the latest relevant tweet they see when online instead of the first tweet received when offline. Therefore, the temporal pattern of user activity also affects the order of information adoption, which will be further investigated in this paper.

[1] http://thenoisychannel.com/2009/01/13/a-twitter-analog-to-pagerank/

More in-depth investigations of user interactions have also been undertaken in existing literature. Apart from the network structure, Tang et al. (2009) also investigated the topical similarity between two users and proposed Topical Affinity Propagation (TAP) to model the topic-level social influence on large networks. A topic-based model TwitterRank was introduced by Weng et al. (2010) to estimate user influence with pre-computed topic distributions. It is also a variant of PageRank, but with topic-specific random walk. The transition probability from one user to another is defined as:

$$P_t(u,v) = \frac{|\tau_v|}{\sum_{a:\, u\ follows\ a} |\tau_a|} sim_t(u,v),$$

where, τ_v is the number of tweets posted by v, $\sum_{a:\, u\ follows\ a} |\tau_a|$ is the number of tweets posted by all fiends of u, and $sim_t(u,v)$ measures the topic similarity between u and v.

Bi et al. (2014) proposed a Followship-LDA (FLDA) model to integrate both topic discovery and social influence analysis in the same generative process and demonstrated it produced precise results. Katsimpras et al. (2015) recommended a supervised algorithm, i.e., Topic-Specific Supervised Random Walks (TS-SRW), to measure user influence using PageRank with transition probability biased towards influential users. These topical models also overlooked the behavioral characteristics of the interaction itself. Dynamic user activity patterns can also impact user interactions; after all, topic interest is a rather static user profile based on long-term observations.

More recent works demonstrated new influence estimation models in continuous-time diffusion networks and multi-source social networks. Du et al. (2013) proposed a randomized influence estimation algorithm in continuous-time diffusion networks, which can provide a more accurate estimate of the number of follow-ups. Rao et al. (2015) proposed a hierarchical framework to generate an overall influence score by combining user information from multiple networks and communities. In this paper, we measure user influence by incorporating user attributes, network structure, topical similarity, and user activity patterns into a PageRank based ranking model. We formulate the transition probability by the expected number of responses which is predicted by a logistic regression model, and bias the random walk towards more active users based on user temporal patterns.

3 User Activity Analysis

In this section, we undertake an initial analysis to reveal user statistics and temporal activity patterns so as to better understand how user activity patterns affect user influence. The social network used in this paper is a sub-network of Twitter: 7.2K New York users, 751K following links and 3.5M tweets (including 565K retweets and 787K replies).

The sub-network was collected through Twitter Streaming API[2] with region of interest specified to New York City. The API returns approximately 1% of randomly sampled real-time data (tweets and users) with respect to our region-specific query (Morstatter et al. 2013). In this step, 10K New York users were collected. We then extracted the largest connected component (7.5K users) of this sampled network. Finally, we collected these users' profiles and

[2] https://dev.twitter.com/streaming/overview

tweets between 24th December 2013 and 24th January 2014 via the REST API [3]. As users that were extremely inactive have little contribution to the influence analysis, those users with less than 20 tweets were removed to get our final dataset.

3.1 User Statistics

To better understand user statistics, we illustrate the distribution of the number of users over the number of followers, the number of friends, and the number of tweets in Figures 1, 2, and 3 respectively. As can be seen from the figures, the three distributions all follow a power law distribution which is consistent with previous observations (Ritterman et al. 2009, Kwak et al. 2010, Petrovic et al. 2011). Figure 4 shows a positive correlation between the number of followers and the number of friends, which indicates that a user with more friends tends to have more followers.

3.2 Temporal User Activity Patterns

As tweet, retweet, and reply are the three major user activities, we approximate the overall user activity in Twitter by the total number of these three activities of all users. The overall user activity pattern is revealed at two different time granularities: hour of day and day of week. As shown in Figure 5, users are more active from 10:00 a.m. to 23:00 p.m. than in other hours which is consistent with people's daily work-rest routine. The weekly pattern illustrated in Figure 6 reveals that users are more active on Monday and Tuesday and less active on Friday and Saturday. This may imply that people find it hard to focus on their work on Monday and Tuesday after the weekend. A more detailed user activity pattern is depicted in the form of a heat map in Figure 7.

To answer the question of whether everybody has similar activity patterns, we further investigate individual-level activity patterns by clustering them into several clusters using the K-Spectral Centroid (K-SC) algorithm (Yang & Leskovec 2011). K-SC is a shape-based clustering algorithm derived from K-Means and invariant to scaling and shifting. The optimal number of clusters can be found by the Average Silhouette Coefficient (ASC) which measures the intra-cluster cohesion and inter-cluster separation (Rousseeuw 1987). Figure 8 illustrates the three most common patterns ($k = 3$) along with the proportion of users in each cluster.

As can be seen from Figure 8, the three patterns (especially C_3) are very different to each other. More specifically, 55% of users (C_1 and C_2) are more active between 14:00 p.m. and 21:00 p.m. while the remaining 45% (C_3) are more active between 0:00 a.m. to 4:00 a.m. Moreover, 13% (C_2) of users are active for a shorter period of time when compared to other users and they have a significant activity burst at around 17:00 p.m. Overall, it suggests that users follow certain activity patterns and posting tweets when followers are very active may have a better chance of attracting attention.

3.3 Response Behaviour Analysis

In order to provide a more in-depth understanding of how user activity patterns affect user influence, we further explore user behavioral patterns in response activity (including retweet and reply) by considering two metrics: *delay* and *trace*. Suppose user v posted a tweet tw at time t_i, and at time t_j, follower u responded to this tweet. Then, the two metrics can be defined by Definition 1 and 2.

Definition 1 *The delay in a response is the time interval between the tweet and the response.*

$$delay = t_j - t_i$$

Definition 2 *The trace is the number of earlier tweets that the follower u needed to trace back before reading the tweet tw.*

$$trace = |\{tw_k | tw_k \in RC_u \quad and \quad t_i < t_k < t_j\}|,$$

where, RC_u represents all the tweets received by u.

The cumulative distributions of *delay* and *trace* of all responses are illustrated in Figures 9 and 10 respectively. As shown, 72% of the retweets and 83% of the replies occurred within 1 hour after the original tweets were posted, while 78% of the retweets and 85% of the replies had been done by tracing back less than 100 earlier tweets. This indicates that users dislike reading many earlier tweets. Therefore, response activity is time-sensitive and this significantly affects users' influence on each other.

4 Response Prediction and Formulation

As response is an explicit indication that a follower has read the tweet and shows a strong interest to interact, we use responses to measure the inter-user influence. We assume that if a user responds to any tweets posted by a friend, that the user is influenced by this friend. In order to formulate the correlation between response and many other factors such as the number of friends, the number of followers, and activity patterns, we apply predictive models to predict the probability of response for a specific tweet. Frequently used notations are given in Table 1.

Table 1: Symbolic notations

Notation	Description
V	user set ($u, v \in V$)
E	following links (u follows v)
F	friend set
$\mathcal{F}$	responded friends
FL	follower set
T	posted tweets
RC	received tweets
RT	retweets
m	the number of favourites

4.1 Feature Selection

Features used for response prediction are selected based on our findings in user activity analysis, as well as existing works (Petrovic et al. 2011, Guille & Hacid 2012, Cossu et al. 2016). Our 10 selected features can be grouped into 3 categories: user attributes, temporal activity, and topical similarity.

User Attributes:

1. The number of listed times (LI)
   According to (Petrovic et al. 2011), this factor is more powerful in measuring popularity than the number of followers or friends.

[3] https://dev.twitter.com/rest/public

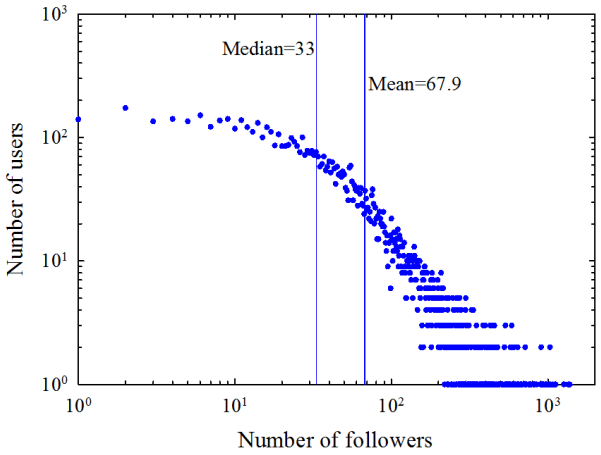


Figure 1: Number of users vs. Number of followers

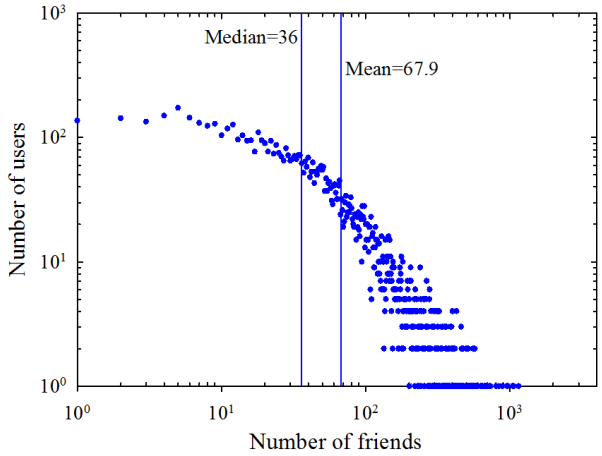


Figure 2: Number of users vs. Number of friends

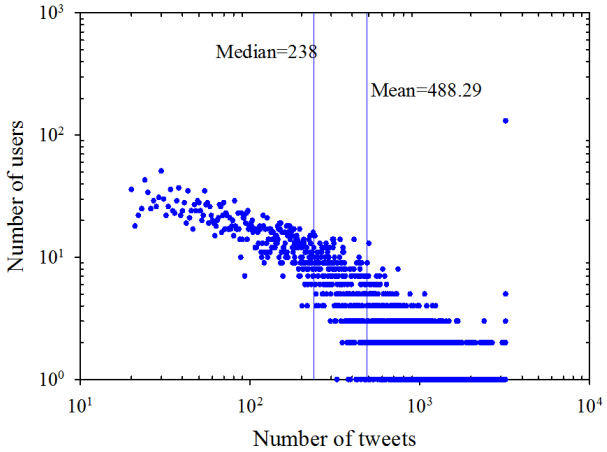


Figure 3: Number of users vs. Number of tweets

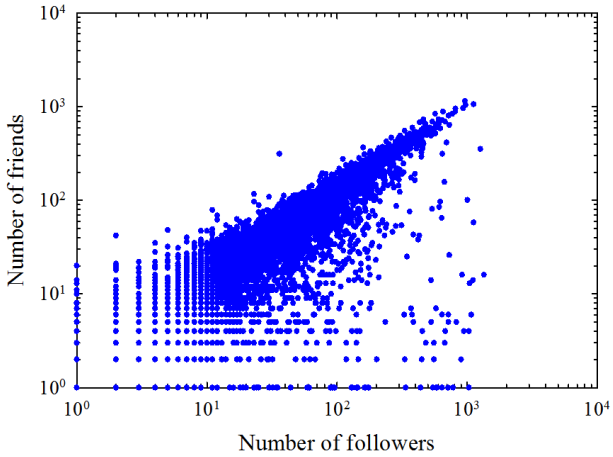


Figure 4: Number of followers vs. Number of friends

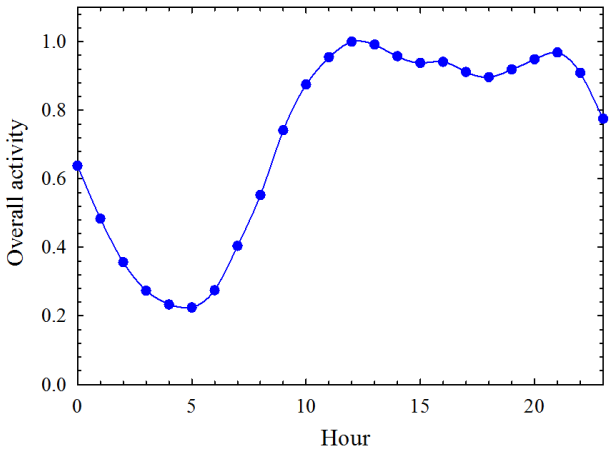


Figure 5: Hourly user activity pattern

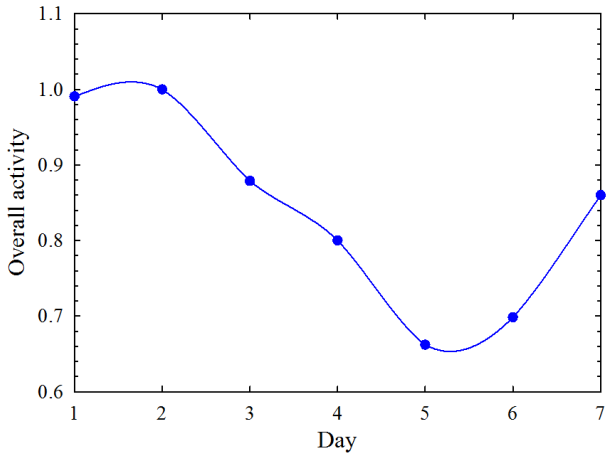


Figure 6: Weekly user activity pattern

2. Favourites received per tweet (FV)

$$FV_v = \frac{m_v}{|T_v|} \quad (1)$$

3. Verified user (VR)

$$VR_v = \begin{cases} 1 & \text{if } v \text{ is verified} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

It has been reported that users are more likely to reponse to verified friends (Petrovic et al. 2011). Verified user has a blue badge next to the name indicating that this account of public interest is authentic [4].

4. Retweet ratio (RR)

[4] https://support.twitter.com

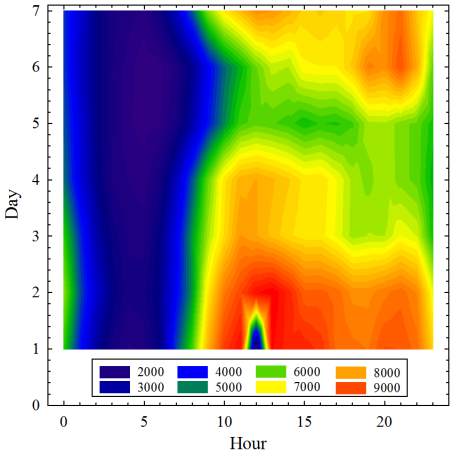


Figure 7: Heat map of user activity

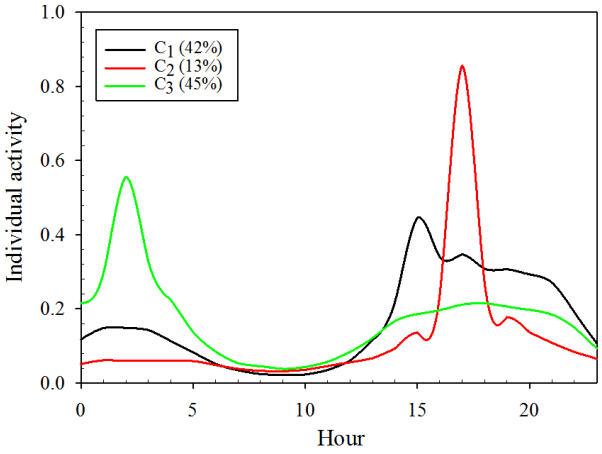


Figure 8: Three common activity patterns

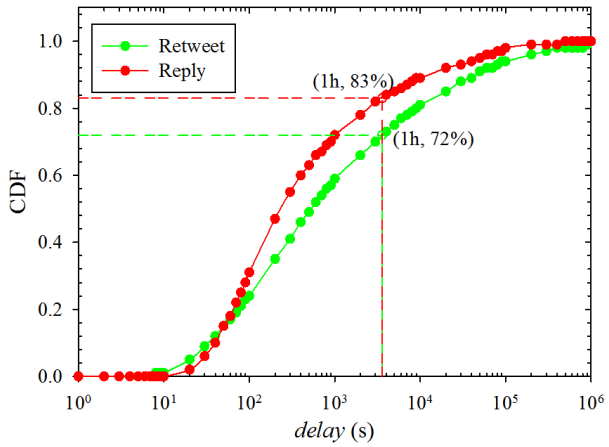


Figure 9: The cumulative distribution of delay

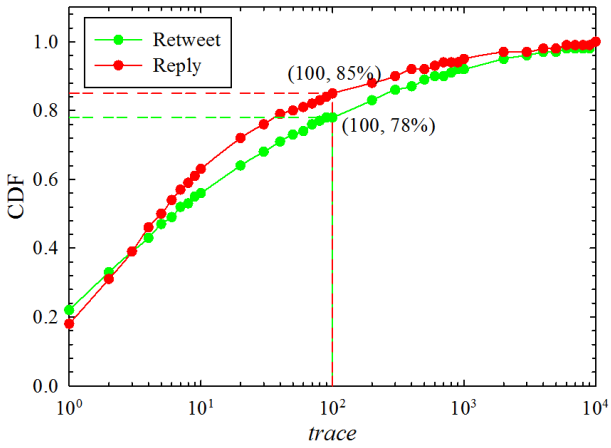


Figure 10: The cumulative distribution of trace

$$RR_v = \frac{|\mathcal{T}_v|}{|T_v|} \tag{3}$$

This feature reflects how actively a user gets involved in the interaction. The higher the RR, the more active the user (Guille & Hacid 2012).

5. Ever responded (RE_{uv})

$$RE_{uv} = \begin{cases} 1 & \text{if } v \in \mathcal{F}_u \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Previous interaction records can be an indicator of how closely two users are related. Here, we define v as a close friend of u if $RE_{uv} = 1$. Otherwise, v is a normal friend of u.

6. Proportion of tweets (PT_{uv})

$$PT_{uv} = \frac{|T_v|}{\sum_{f \in F_u} |T_f|} \tag{5}$$

PT_{uv} is the proportion of v's tweets to all tweets of u's friends. The higher the PT_{uv} is, the more easily v can draw attention from u (Cossu et al. 2016).

Temporal Activity:

1. The number of tweets posted in hour t (N^t)

$$N_v^t = \frac{|T_v^t|}{d_v},\ t \in [0, 23] \tag{6}$$

where, d_v is the number of available days in our observations and is calculated by the time interval between the first and last tweet of v that available in our dataset.

2. User activity at time t (A^t)

$$A_v^t = \frac{|N_v^t|}{\sum_{h=0}^{23} N_v^h},\ t \in [0, 23] \tag{7}$$

A^t represents the probability a user is active in hour t.

3. Joint activity at time t (JA^t)

$$JA_{uv}^t = A_u^t A_v^t,\ t \in [0, 23] \tag{8}$$

JA_{uv}^t is the probability that two users are both active in hour t under the assumption that u and v are independent.

Topical Similarity:

1. Topic similarity (TS)

$$TS_{uv} = \sqrt{2 * D_{JS}(u, v)} \tag{9}$$

$D_{JS}(u, v)$ is the JensenShannon divergence between the topic distributions of two users (Weng et al. 2010).

4.2 Response Prediction

Given two users u and v (u follows v), with v having posted a tweet at time t, the task of response prediction is to predict the probability that u will respond to this tweet. For each tweet, we pair the author with each of his followers to generate an instance. This results in 2.9M instances of which 1.3% are responses and the rest are non-responses. We further subsample a balanced dataset with 66K instances (including 33K responses and 32.9K non-responses) and normalize all features to $[0, 1]$. It is worth mentioning that those tweets with author or follower not in the user set are removed initially.

Based on this dataset, we apply logistic regression (LR), C4.5 and multilayer perceptron (MLP) with one hidden layer (100 nodes) with 5-fold cross validation to predict the probability of response. The accuracy performance is reported in Table 2. As can be seen, C4.5 achieves the best accuracy followed by LR and MLP with similar performance results.

Table 2: The accuracy of response prediction

Classifier	LR	C4.5	MLP
Accuracy	86%	89%	86%

Figure 11 illustrates a few important features ranked by the absolute value of the weight learnt by LR. Feature RE_{uv} (ever responded) dominates the model, in fact, nearly 60% of the responses occurred between users who interacted with each other more than once. It proves that users are more prone to respond to close friends than normal friends.

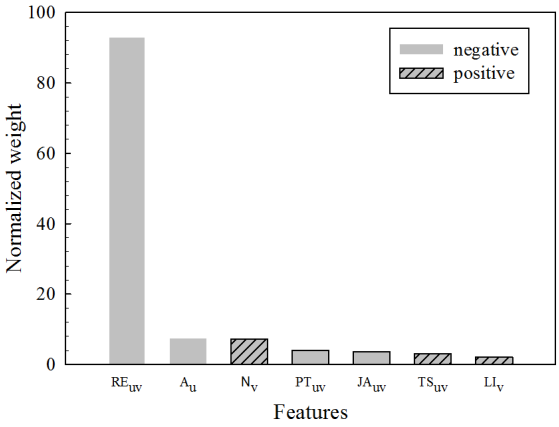


Figure 11: Important features ranked by weight

For the author, the number of tweets (N_v) posted is the most influential factor on response. Since the weight of N_v is positive, posting more tweets will, however, decrease response probability. This is because the predicted probability is averaged to one tweet which decreases when the number of tweets increases. For the follower, the normalized activity (A_u) with a negative weight is the most important factor. It means that the more active the follower is, the more chance there is that the tweet will get a response. This can also be prove by joint activity (JA_{uv}) with a positive weight. Moreover, we find that user attributes such as the number of favourites (FV_v), verified user (VR_v), and retweet ratio of the follower (RR_u) have little impact on response interaction.

4.3 Response Probability Formulation

Although LR is not the best model, the response probability of LR can be easily described in an explicit form as such can be easily integrated into an influence ranking model. There are two benefits of such integration. First, the response probability can be estimated within the influence model, otherwise, it has to be estimated separately by a prediction model. Second, it makes the influence model more flexible in adjusting the weights of different features in different application contexts.

Therefore, we adopt LR to formally define the response probability. Given feature space S, follower u, friend v and a tweet posted by v at time t, the probability that u will respond to this tweet can be defined by:

$$P_{uv}^t = \frac{1}{1 + exp(w_0 + \sum_{i=1}^{|S|} w_i S_i^t)} \tag{10}$$

5 A Temporal Influence Ranking Model

In this section, we introduce a new temporal influence ranking (TIR) model based on PageRank model to estimate user influence so as to identify influentials. TIR is a PageRank based graphic algorithm taking the temporal homogeneity of user activity patterns into consideration. As a response is an explicit indication of user influence of one user on another, we approximate the influence of a friend to a follower by the expected number of responses the friend may receive from the follower. Since response activity is time-sensitive, this approximation is performed for each hour based on integrated hourly response probability between users.

5.1 Model Formulation

For the purpose of simplicity, we denote the directed graph formed by users and following links as $G(V, E)$, where V is the node set with each node being a user, E is the edge set with each edge being a directed link pointing from follower to friend. User influence propagates from one node to another with a certain probability along the edges in the graph.

Given the response probability of a particular tweet, the hourly response probability can be calculated by aggregating the probability of all tweets posted during hour t. To better differentiate normal friend and close friend, we first fix the value of feature RE_{uv} in Equation (10) to 1, then introduce a penalty factor c to penalize normal friends. Thus, the transition probability from u to v in hour t can be defined as:

$$\mathcal{P}_{uv}^t = \begin{cases} cN_v^t P_{uv}^t & \text{if } v \in \mathcal{F}_u \\ (1-c)N_v^t P_{uv}^t & \text{otherwise} \end{cases} \tag{11}$$

where, P_{uv}^t is the response probability of one tweet as defined in Equation (10) with learnt weights and N_v^t is the number of tweets posted by v in hour t as defined in Equation (6). $c \in [0.5, 1]$ is the penalty factor that defines how close friends and normals friends are treated. $c = 0.5$ indicates that close friends and normal friends are considered to be the same while $c = 1$ means normal friends will be considered strangers.

Then we plug the hour-based response probability $\mathcal{P}^t_{uv}$ into the PageRank model where user influence can be calculated iteratively by the following Equation:

$$\vec{R}^t = M \times \vec{R}^t, \tag{12}$$

where, M is a $|V| \times |V|$ transition probability matrix with element M_{uv} defined as the response probability of user u to user v in hour t:

$$M_{uv} = \begin{cases} \mathcal{P}^t_{uv} & \text{if } u \; follows \; v \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

Equation (12) is equivalent to an eigensystem with an eigenvalue of 1. Therefore, it has a solution if and only if it meets three conditions: 1) M is a stochastic matrix, 2) M is irreducible (i.e., M is a strongly connected matrix) and 3) M is non-cyclical (Page et al. 1999). However, these conditions are not satisfied in reality: 1) condition 1 is not satisfied as users with no followers will end up with zero column vectors in M, 2) condition 2 might not be satisfied as the strong connectivity of M is not guaranteed, and 3) cyclical nodes are highly possible in social networks.

To meet the three conditions, transformations should be performed on transition matrix M. First, we normalize M by the sum of each column to make it scale invariant. Then, we add a $|V| \times |V|$ matrix with all elements of the same value $\frac{1}{|V|}$ to M following Equation (14). The damping factor γ represents the probability that a stranger randomly jumps to a user without following the links. As generally assumed, γ in this paper is set to 0.85.

$$\mathcal{M} = \gamma M + (1-\gamma) \times \left[\frac{1}{|V|}\right]_{|V|\times|V|} \tag{14}$$

All three of the above conditions are satisfied after these transformations. As each entry in $\mathcal{M}$ is a non-negative real number and the sum of each column equals to one, $\mathcal{M}$ is a column-stochastic matrix and also a strongly connected matrix. Since each node in $\mathcal{M}$ is connected to other nodes directly, the shortest path of each node to itself equals to one which means $\mathcal{M}$ is also a non-cyclical matrix. Thus, the proposed TIR model can be written in a new form as defined in Equation (15).

$$\vec{R}^t = \gamma M \times \vec{R}^t + (1-\gamma) \times \left[\frac{1}{|V|}\right]_{|V|\times 1} \tag{15}$$

Overall, by utilizing the expected number of response between two users as the transition probability, we bias the random walk in TIR towards more interactive friends and such bias is based on user attributes, temporal activity and topical similarity. We also introduce a penalty factor to further control such bias.

5.2 Discussion

The TIR model has three advantages over existing models. First, it provides more accurate influence estimation. This is because people's interests and behaviours change over time. Such changes can be easily captured by TIR as it evaluates user's influence based on the dynamic information that occurred in a short period of time (days or weeks). As it doesn't rely on the whole Twitter dataset, this also makes it more efficient. Existing models such as TwitterRank depends on massive tweets to get a good estimation. Third, TIR is more flexible. This is because it incorporates a explicit formulation of logistic regression as part of the model to formulate the transition probability. Such formulation can be easily adjusted in different application contexts. For example, if the number of followers is considered to be very important in a context, we can adjust its weight in the formulation accordingly.

In Twitter, the following relationship between follower and friend is asymmetrical (or weak), that is, the friend doesn't have to follow back to his followers. However, in some other social networks such as Facebook, the following relationship is symmetrical (or strong), i.e., the friend has to accept the connection request and follow back. Our model can be easily generalized to those symmetrical networks by taking the symmetrical network as a special case of the asymmetrical network where all friends and followers follow each other.

6 Experiments and Results

To better evaluate TIR, we conduct two different types of experiments to contrast it with existing TunkRank and TwitterRank models. First, we compare their similarities and differences in global influence ranking. Then, we evaluate their performance in a friend recommendation task. It is worth mentioning that the experiments are based on our sampled dataset, not the whole Twitter dataset.

6.1 Global Influence Ranking

The user influence obtained from TIR is an hour-based influence. The global influence can be further calculated via Equation (16) where w_t is the weight of user influence in hour t. w_t can be the overall user activity, in which case the aggregated influence serves as the global influence. Observe that w_t can also be individual activity of a particular user, and in that case, the outcome can be interpreted as the personal perspective influence.

$$\vec{R} = \sum_{t=0}^{23} w_t \vec{R}_t \tag{16}$$

We compare the TIR model with the existing TunkRank and TwitterRank models to examine their similarities and differences in global influence ranking. The penalty factor c of TIR model is set to 0.5, 0.85 and 1 in order to explore its influence to the final ranking. The top 10 influentials are listed in Table 3. As we can see, the top 10 influentials are of different types such as news media (e.g., "ABC", "nprnews and "latimes"), food (e.g., "WholeFoods" and "deverfoodguy"), sports (e.g., "HPbasketball", "BlkSportsOnline" and "YogaArmy") and public figures (e.g., "chrisbrogan" and "bomani_jones"). It's not surprising that news media outnumbers the other types of users as latest news stories are usually well-written and very attractive.

Comparison between $TIR_{c=0.5}$ and $TIR_{c=0.85}$ indicates that TIR with a larger penalty factor makes users with more responsive followers stand out which is an advantage over other models. Take "bomani_jones" for example, 12% of his tweets were retweeted by others which even includes top influentials such as "talkhoops" and "rodimusprime". "MySOdotCom", however, only has 5% retweeted tweets. When the penalty increases, the influence of "bomani_jones" increases while that of "MySOdotCom" decreases.

Furthermore, we calculate the commonly used Kendalls τ rank correlation coefficient (Knight 1966) to measure rank correlations between different models. As shown in Table 4, $TIR_{c=0.5}$ is more similar to the other two models when compared with $TIR_{c=1}$. This is because $TIR_{c=0.5}$ ignores the difference between close friend and normal friend as both TunkRank and TwitterRank do. Also, TIR is more similar to TwitterRank than to TunkRank whether $c = 0.5$ or $c = 1$ which is because both TIR and TwitterRank take topical similarities into consideration.

Overall, with a controllable penalty factor, TIR is more flexible than TunkRank and TwitterRank in global influence ranking. Particularly, TIR with a larger penalty factor can better differentiate those influentials with more responsive followers.

Table 4: Kendall τ rank coefficient

Pairs	τ
TR vs PR	0.49
$TIR_{c=1}$ vs PR	0.28
$TIR_{c=0.5}$ vs PR	0.51
$TIR_{c=1}$ vs TR	0.4
$TIR_{c=0.5}$ vs TR	0.65

6.2 Friend Recommendation

We further investigate the performance of TIR, TunkRank and TwitterRank with respect to friend recommendation (also known as the link prediction) by conducting the same experiment as in Weng et al. (2010). As listed in Table 5, links that need to be predicted are selected based on either friend attribute or similarities between the two users. Each of the eight link sets represents a specific test scenario and contains 30 links randomly selected from all following links regarding its selection criteria. L_{fh}, for example, we first rank all the links based on the number of followers of the friend user, then randomly select 30 links from the top 10% links. Note that the Jenson-Shannon distance (one of the selection criteria in Table 5) is calculated based on the feature vectors of the two users.

The experiment is carried out for each link set L follows the steps below:

1. Take one link out of L: $l(u, v)$;
2. Randomly select 10 users who u doesn't follow as the test candidate set C;
3. Remove link $l(u, v)$ from graph G and denote the new graph by $G^{'}$;
4. Apply ranking model on $G^{'}$ and calculate:

$$Q(l) = |\{v^{'}|v^{'} \in C, Rank(v^{'}) > Rank(v)\}|$$

5. Repeat 1 - 4 for all links in L and calculate the average $Q(l)$.

As the purpose is to recommend friends for u, the personal perspective influence ranking instead of the global ranking is applied here for both TwitterRank and TIR. Since the ground truth is "u follows v", the higher the $Q(l)$ the better the performance. As illustrated in Figure 12, the TIR model demonstrates better performance and more stability than TunkRank and TwitterRank for most of the scenarios. More specifically, TIR achieves better performance than TwitterRank in 7 test scenarios and outperforms TunkRank in 6 scenarios, especially in L_{fl} and L_{tl} where TunkRank and TwitterRank both show the worst performance.

Observe that TIR is outperformed by TunkRank in L_{fh}. The reason is that TunkRank fully relies on the network structure which leaves it to the other extreme when recommending friends with few followers (L_{fl}). TwitterRank is better than TIR in L_{rr} which suggests that users who followed each other share similar topic interests even though they may have few interactions.

The influence of the penalty factor c varies between scenarios. As shown in Figure 13, c only has a significant influence in TIR when $c \in [0.95, 1.0]$. It increases the performance in recommending friends with few followers (L_{fl}) which indicates that it is likely that a user follows a friend who has few followers because he is a close friend. When recommending friends who either follow back (L_{rr}) or not (L_{ur}), a higher c decreases the performance which means users follow each other are not necessarily likely to interact with each other. However, the number of tweets a friend has doesn't affect his interactions with followers as c shows little impact in L_{th} and L_{tl}.

7 Conclusion and Future Work

In this paper, we first investigated the temporal characteristics of user activity and found that user activities display certain patterns either for overall user activity or individual level activity. Moreover, we found that users tend to ignore tweets posted one hour ago and don't like to read many earlier tweets. Such characteristics suggest that user influence is time-sensitive and highly affected by user activity patterns.

As response is an explicit indicator of user influence on others, we further formulated the response probability using a logistic regression model. Based on the formulated response probability, we proposed a PageRank based new influence ranking model (i.e., TIR) to estimate hourly user influence by taking hourly response probability as transition probability and introduced a penalty factor to bias the random walk towards close friends. Our method (TIR) was evaluated against two existing models: TunkRank and TwitterRank, in global influence ranking and friend recommendation separately. Experimental results demonstrated that TIR is more flexible than TunkRank and TwitterRank, and can better differentiate those influentials with more responsive followers. Results in friend recommendation substantiated the claim that TIR is more accurate and stable in a number of scenarios.

The next step of our research is to develop a web service to provide hourly based influentials for other applications. Meanwhile, TIR can also be used to develop useful extensions for Twitter such as finding the most influential friends for individual users or providing advice on when to post in order to attract more responses. Moreover, we will adapt our work to heterogeneous social networks so as to find influentials across multiple online social networks.

References

Bakshy, E., Hofman, J. M., Mason, W. A. & Watts, D. J. (2011), Everyone's an influencer: quantifying influence on twitter, *in* 'Proceedings of the fourth

Table 3: Top 10 globally ranked users

Rank	$TIR_{c=0.5}$	$TIR_{c=0.85}$	$TIR_{c=1}$	TwitterRank	TunkRank
1	WholeFoods	WholeFoods	bomani_jones	XboxSupport	PATisDOPE
2	MySOdotCom	bomani_jones	InTheBleachers	Foxmental_X	nprnews
3	nprnews	MySOdotCom	ABC	MySOdotCom	bigmarkspain
4	bomani_jones	greensboro_nc	wsbtv	denverfoodguy	MySOdotCom
5	greensboro_nc	nprnews	NBCNews	chrisbrogan	chrisbrogan
6	XboxSupport	ABC	rodimusprime	CHRISVOSS	FastCompany
7	denversolarguy	InTheBleachers	ajc	PATisDOPE	CoryBooker
8	denverfoodguy	XboxSupport	HPbasketball	denversolarguy	WholeFoods
9	jilevin	SpitToonsSaloon	talkhoops	nprnews	latimes
10	YogaArmy	PRideas	BlkSportsOnline	bigmarkspain	ABC

Table 5: Selected test scenarios

Link Set	Rank User	Selection Criteria
L_{fh}	friend	the number of followers (high 10%)
L_{fl}	friend	the number of followers (low 10%)
L_{th}	friend	the number of tweets (high 10%)
L_{tl}	friend	the number of tweets (low 10%)
L_{dh}	both	Jenson-Shannon distance (high 10%)
L_{dl}	both	Jenson-Shannon distance (low 10%)
L_{rr}	both	Followed each other
L_{ur}	both	Only the follower followed the friend

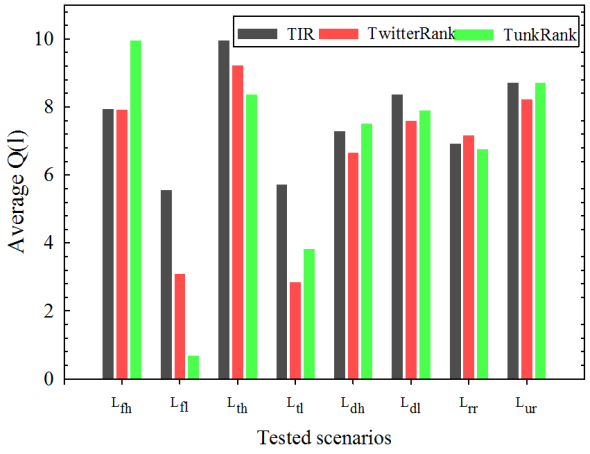


Figure 12: Performance in friend recommendation

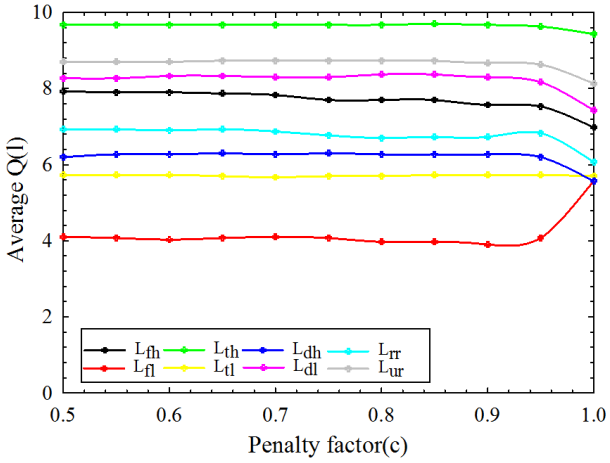


Figure 13: The influence of penalty factor c to TIR

ACM international conference on Web search and data mining', ACM, pp. 65–74.

Bi, B., Tian, Y., Sismanis, Y., Balmin, A. & Cho, J. (2014), Scalable topic-specific influence analysis on microblogs, *in* 'Proceedings of the 7th ACM international conference on Web search and data mining', ACM, pp. 513–522.

Cataldi, M. & Aufaure, M.-A. (2015), 'The 10 million follower fallacy: audience size does not prove domain-influence on twitter', *Knowledge and Information Systems* **44**(3), 559–580.

Cha, M., Haddadi, H., Benevenuto, F. & Gummadi, P. K. (2010), 'Measuring user influence in twitter: The million follower fallacy.', *ICWSM* **10**(10-17), 30.

Chan, K. K. & Misra, S. (1990), 'Characteristics of the opinion leader: A new dimension', *Journal of advertising* **19**(3), 53–60.

Chen, W., Wang, C. & Wang, Y. (2010), Scalable influence maximization for prevalent viral marketing in large-scale social networks, *in* 'Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 1029–1038.

Cossu, J.-V., Labatut, V. & Dugué, N. (2016), 'A review of features for the discrimination of twitter users: application to the prediction of offline influence', *Social Network Analysis and Mining* **6**(1), 1–23.

Domingos, P. & Richardson, M. (2001), Mining the network value of customers, *in* 'Proceedings of the

seventh ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 57–66.

Du, N., Song, L., Gomez-Rodriguez, M. & Zha, H. (2013), Scalable influence estimation in continuous-time diffusion networks, *in* 'Advances in neural information processing systems', pp. 3147–3155.

Guille, A. & Hacid, H. (2012), A predictive model for the temporal dynamics of information diffusion in online social networks, *in* 'Proceedings of the 21st international conference on World Wide Web', ACM, pp. 1145–1152.

Katsimpras, G., Vogiatzis, D. & Paliouras, G. (2015), Determining influential users with supervised random walks, *in* 'Proceedings of the 24th International Conference on World Wide Web', ACM, pp. 787–792.

Kempe, D., Kleinberg, J. & Tardos, É. (2003), Maximizing the spread of influence through a social network, *in* 'Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 137–146.

Knight, W. R. (1966), 'A computer method for calculating kendall's tau with ungrouped data', *Journal of the American Statistical Association* **61**(314), 436–439.

Kwak, H., Lee, C., Park, H. & Moon, S. (2010), What is twitter, a social network or a news media?, *in* 'Proceedings of the 19th international conference on World wide web', ACM, pp. 591–600.

Lazarsfeld, P. F., Berelson, B. & Gaudet, H. (1948), 'The peoples choice: how the voter makes up his mind in a presidential campaign.'.

Leavitt, A., Burchard, E., Fisher, D. & Gilbert, S. (2009), 'The influentials: New approaches for analyzing influence on twitter'.
URL: *http://www.webecologyproject.org/wpcontent/uploads/2009/09/influence-report-final.pdf*

Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J. & Glance, N. (2007), Cost-effective outbreak detection in networks, *in* 'Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 420–429.

McQuail, D. (1987), *Mass communication theory: An introduction* ., Sage Publications, Inc.

Morstatter, F., Pfeffer, J., Liu, H. & Carley, K. M. (2013), 'Is the sample good enough? comparing data from twitter's streaming api with twitter's firehose', *arXiv preprint arXiv:1306.5204* .

Page, L., Brin, S., Motwani, R. & Winograd, T. (1999), 'The pagerank citation ranking: bringing order to the web.'.

Petrovic, S., Osborne, M. & Lavrenko, V. (2011), Rt to win! predicting message propagation in twitter., *in* 'ICWSM'.

Rao, A., Spasojevic, N., Li, Z. & Dsouza, T. (2015), Klout score: Measuring influence across multiple social networks, *in* 'Big Data (Big Data), 2015 IEEE International Conference on', IEEE, pp. 2282–2289.

Richardson, M. & Domingos, P. (2002), Mining knowledge-sharing sites for viral marketing, *in* 'Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 61–70.

Ritterman, J., Osborne, M. & Klein, E. (2009), Using prediction markets and twitter to predict a swine flu pandemic, *in* '1st international workshop on mining social media', Vol. 9, ac. uk/miles/papers/swine09. pdf (accessed 26 August 2015), pp. 9–17.

Rogers, E. M. (1962), *Diffusion of innovations.*, Free Press.

Rousseeuw, P. J. (1987), 'Silhouettes: a graphical aid to the interpretation and validation of cluster analysis', *Journal of computational and applied mathematics* **20**, 53–65.

Tang, J., Sun, J., Wang, C. & Yang, Z. (2009), Social influence analysis in large-scale networks, *in* 'Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 807–816.

Watts, D. J. & Dodds, P. S. (2007), 'Influentials, networks, and public opinion formation', *Journal of consumer research* **34**(4), 441–458.

Weng, J., Lim, E.-P., Jiang, J. & He, Q. (2010), Twitterrank: finding topic-sensitive influential twitterers, *in* 'Proceedings of the third ACM international conference on Web search and data mining', ACM, pp. 261–270.

Yang, J. & Leskovec, J. (2011), Patterns of temporal variation in online media, *in* 'Proceedings of the fourth ACM international conference on Web search and data mining', ACM, pp. 177–186.

Towards an accurate social media disaster event detection system based on deep learning and semantic representation

Zhihong Lin[1] Huidong Jin[2] Bella Robinson[2] Xunguo Lin[3]
[1]Institute of Policy and Management, Chinese Academy of Sciences, Beijing 100190, China
[2]Data61, CSIRO, GPO Box 664, Canberra ACT 2601, Australia
[3]Branch of Strategy, Quality & Intelligence, Civil Aviation Safety Authority, GPO BOX 2005, Canberra ACT 2601, Australia
linzhihong1991@gmail.com Warren.Jin@csiro.au Bella.Robinson@csiro.au
Xunguo.Lin@casa.gov.au

Abstract

As a new source of real-time and first-hand information for emergency management, social media plays an increasingly important role in disaster event detection and monitoring. Message classification is a core module of a social media disaster event detection system and its accuracy affects the validity of such a system. Traditional document classification methods, such as using Support Vector Machine (SVM) to classify a TF-IDF matrix, do not take account of the semantic meaning and sequential information of words within a message, which if used should improve accuracy.

This paper provides examples of integrating the semantic and sequential information of words into the classification of Chinese microblog messages. A deep learning model, the convolutional neural network, embodied by semantic word vectors based on Chinese Wikipedia, is used. When compared to the traditional SVM classifier, the proposed classification techniques improve the performance of microblog classification significantly, which should result in a better social media disaster event detection system.

Keywords: Convolutional neural networks, Word2vec, Tensorflow, Message classification, Earthquake, Disaster event detection

1 Introduction

In recent years, the problem of nature disasters has become more and more severe. Natural disasters affected 141 million people and resulted in US$110 billion total damage worldwide in 2014, referring to the latest annual report on humanitarian crises and assistance from the United Nations Office for the Coordination of Humanitarian Affairs (OCHA, 2015). China was the most affected country with 58 million people impacted, followed in orders by the Philippines, India, Burkina Faso and Sri Lanka. The biggest disaster events in China in 2014 in terms of cost were earthquakes, which resulted in losses of US$5 billion (OCHA, 2015).

Since natural disasters cause significant property loss and death or injury to people around the world, it is important to enhance the ability and effectiveness of emergency management (OCHA, 2015). As mentioned in literature (Robinson et al., 2014), one of the pressing challenges while managing an emergency or crisis event is the collection and communication of reliable, relevant and up-to-date information. Timely, accurate and effective messages play a vital role for disaster response. For emergencies, making rapid and effective decisions needs such information. Losses may be reduced by providing relevant information to both rescue organizations and potential victims.

As mentioned in literature (Bai et al., 2015), social media has been recognized as an emerging new source of information for emergency managers (Hughes et al., 2014, Olteanu et al., 2015, Thelwall and Stuart, 2007). Twitter in particular is an important channel of communication to source content from people experiencing disasters and for emergency services agencies to inform the public of what is going on. One important characteristic of Twitter is its real-time nature. For example, when an earthquake occurs, people make Twitter posts related to the earthquake, which enables prompt detection of the earthquake occurrence. Another important characteristic of Twitter is its direct nature. For example, Olteanu et al. (2015) found that on average 12% of Tweets during natural disasters events were from eyewitnesses.

Since social media can provide a great deal of real-time first-hand information during disaster events, researchers have developed many disaster event detection systems worldwide mainly based on social media, including: “Did You Feel it”[1], “Toretter” (Sakaki et al., 2013, Sakaki et al., 2010), “Twicident” (Abel et al., 2012), “Tweet4act” (Chowdhury et al., 2013), “CrisisTracker” (Rogstadius et al., 2013), “Ushahidi platform”[2] (used by volunteers during the Haiti earthquake and Hurricane Sandy), “Twitter Earthquake Detector” (Earle et al., 2012),

[1] http://earthquake.usgs.gov/data/dyfi/

[2] http://www.ushahidi.com/

Class	Original microblog message	Translation
(+)	又地震了~	Earthquake again~
(+)	刚刚地震了，有震感，好害怕	Just now quaked, felt vibration, very afraid
(+)	哪里发生地震了？	Where did the earthquake occur?
(-)	5.12汶川地震纪念日	Memorial day of 5.12 Wenchuan Earthquake
(-)	地震发生之前通常有三个先兆……	There are usually three omens before the earthquake……
(-)	齐鲁证券人事地震……	A personnel shake-up of Qilu securities company……

Table 1: Example of positive (+) and negative (-) messages containing the keyword "earthquake" (地震)

"Emergency Situation Awareness" (Cameron et al., 2012), "EARS" (Avvenuti et al., 2014). These systems provide a large amount of useful information to disaster management. For example, Toretter (Sakaki et al., 2013) detects earthquakes promptly and notification is delivered much faster than Japan Meteorological Agency (JMA) broadcast announcements.

China is the world's most populous country and among the top largest countries by land area. Recent rapid economic and technological developments have resulted in more Chinese people having access to computers or smart phones/devices, which are used increasingly to exchange information via microblogging services. In June 2015, the number of users of microblogging services in China reached 204 million, with nearly 79.4% of users (approximately 162 million people) accessing their accounts via smart phones (CNNIC, 2016). This large user group provides an unprecedented opportunity to study Chinese microblogging for the purposes of situation awareness for disaster events.

The most influential Chinese microblogging service similar to Twitter is Sina Weibo, which had more than 156 million active users per month and more than 69 million active users per day in 2014 (Wang et al., 2014). Some social media disaster event detection systems based on Weibo messages have been developed, such as the SWIM system (Robinson et al., 2014).

One of the core parts of these social media disaster event detection systems is message classification, which determines whether a message, that includes a disaster keyword, is truly related to a current disaster event. Take earthquake as an example, out of all the messages including keyword "earthquake", positive messages may talk about the occurrence of a current earthquake or people's feeling of the earthquake. Negative messages may include for example memorial comments about a former earthquake, general-talk about common sense things to do during earthquake or use of the metaphoric meaning of "earthquake". Some typical message examples related to earthquake are listed in Table 1.

Compared to news and the other formal text, microblogs are very short, which makes the extracted features sparse. These short messages commonly focus on only one topic. In addition, microblogs are often lacking formal expression and standard grammar. People love to use a lot of abbreviations, internet words, acronyms and dialect in microblogs. External knowledge is often required to understand the meaning of words and phrases within microblogs. Thus, microblog message classification presents several additional challenges in comparison with document classification (Pang et al., 2015).

One common method of microblog message classification is to: first pre-process and obtain the unigrams of a message, then transform the unigrams into a TF-IDF matrix and finally use traditional classifiers, such as Support Vector Machines (SVM), to classify the message (Bai et al., 2015). However, this bag-of-word model does not take the sequential information of words, such as phrases, into consideration and the sparsity of TF-IDF matrix will likely affect the accuracy of the classifier. In the traditional method, we may use bigrams and trigrams to acquire phrase information. However, this will generate a sparser TD-IDF matrix and increase the operation time exponentially. In addition, this method cannot represent the semantic meaning of words. For example, two words, such as "腿软 (debility of the legs)" and "害怕 (fear)", would be regarded as two different words even though they are often semantically similar in Chinese. Sophisticated topic modelling techniques go beyond bag-of-words and are able to capture the semantic meaning of short messages (Du et al., 2010, Du et al., 2012), but are often quite time consuming on model training.

Deep learning models have achieved remarkable results in computer vision (Krizhevsky et al., 2012), speech recognition (Graves et al., 2013), and very recently in text mining (Kim, 2014).

A Convolutional Neural Network (CNN) is a type of deep learning network which is suitable for extracting sequential and local features, since it utilizes layers with convolving filters that are applied to local features (LeCun et al., 1998). Originally invented for computer vision, CNN models have recently been shown to achieve impressive results on the practically important task of sentence categorisation (Kalchbrenner et al., 2014, Kim, 2014, dos Santos and Gatti, 2014, Goldberg, 2015, Iyyer et al., 2015, Wang et al., 2015). CNNs can capitalise on distributed representations of words by first converting the words comprising each sentence into a vector, forming a matrix for a sentence to be used as input (see Figure 1). To achieve good performance, the models do not necessarily need to be complex. For example, Kim proposed a simple one-layer CNN that achieved state-of-the-art results across several datasets (Kim, 2014). The very strong results achieved with this comparatively simple CNN architecture have suggested that it may serve as a drop-in replacement for widely used models, such as SVM or logistic regression.

The word vector representations play an important role in the CNN model. Word vectors, wherein words are projected from a sparse, 1-of-V encoding (where V is the vocabulary size, which could be around 100,000) onto a

lower dimensional vector space via a hidden layer, are essentially feature extractors that encode semantic features of words in their dimensions (Kim, 2014). The typical characteristic of word vectors is that semantically similar words would be similar to each other by calculating cosine similarity among the vectors of words. Generally, the semantic word vector representations could be learned from a large corpus by using the word2vec model (Mikolov and Dean, 2013, Mikolov et al., 2013) or Glove Model (Pennington et al., 2014).

Motivated by improving the accuracy of microblog message classification, we need to take advantage of the semantic representation and sequential information of words. Thus, we build a convolutional neural network embodied by semantic word vectors for message classification.

First, in order to make the raw Chinese microblog messages suitable for analysis, we conduct pre-processing procedures, such as word segmentation, conversion of traditional Chinese to simplified Chinese, and the removal of stop words and other meaningless characters. Then, in order to obtain the semantic meaning of words, we use the word2vec model to learn the word vector representations from a large corpus. We used the Chinese Wikipedia as an example for this paper. Finally, we build the convolutional neural network embodied by the pre-learned word vectors to classify the Chinese microblog messages. The results of both static and non-static models are presented, which are better than the baseline SVM classifier.

The contributions of the paper are summarised as follows:

- The paper provides examples of integrating the sematic and sequential information of words into the classification of Chinese microblog messages by using deep learning model embodied by word vectors learned from Chinese Wikipedia.
- The proposed classification techniques in this paper improve the accuracy of microblog classification significantly, which should result in a better social media disaster event detection system.

This paper is organized as follows. In the next section, we introduce the techniques used for our message classification, such as CNN and the word2vec model. In Section 3, we describe the datasets we used and the data pre-processing procedures. The results and comparisons

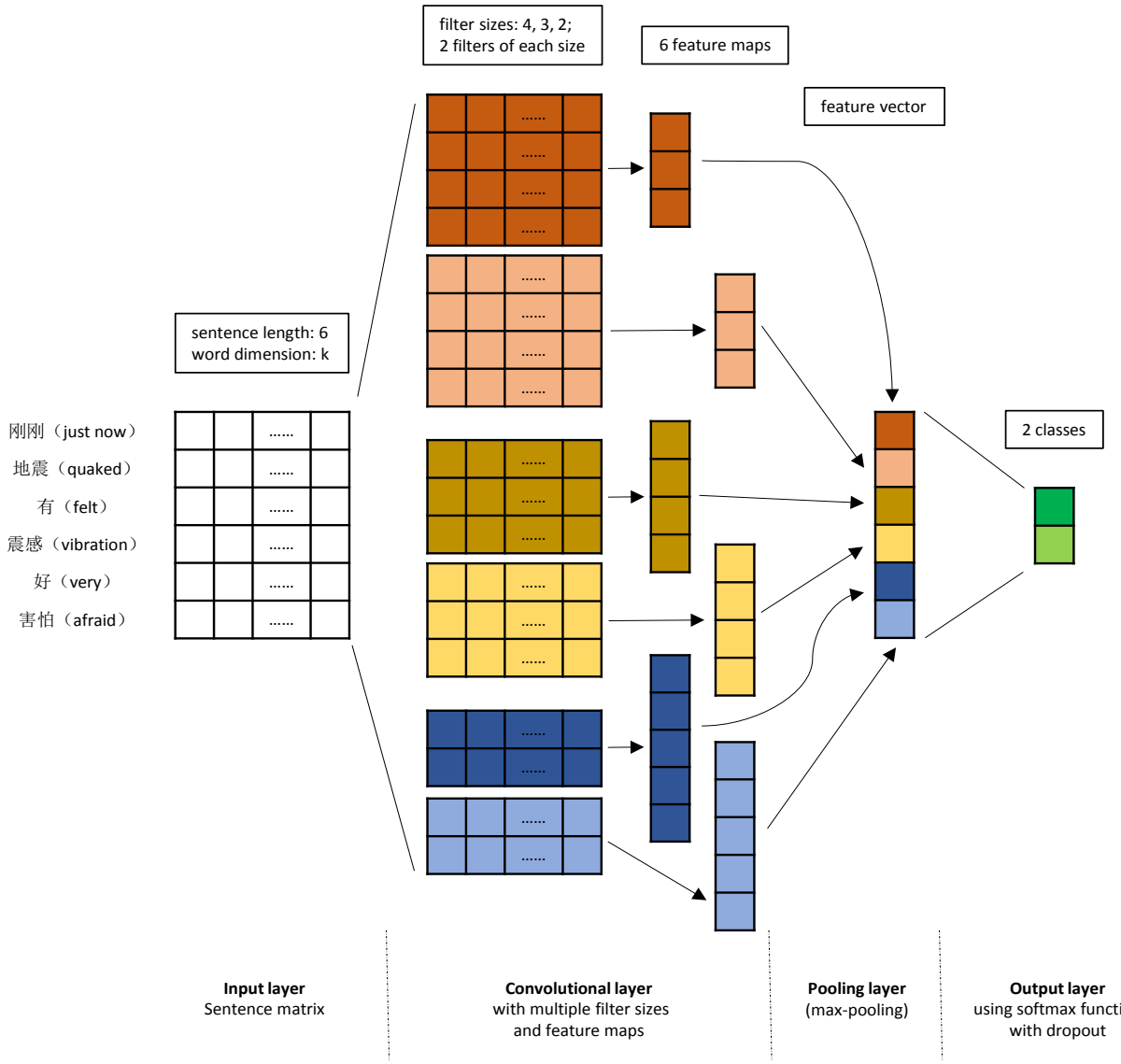


Figure 1: Example of convolutional neural networks for microblog message classification

are shown in Section 4. We provide a discussion in Section 5 and conclude the work in Section 6.

2 Proposed classification techniques

The CNN we employ in this paper is illustrated in Figure 1, which is adopted from Kim (2014) for our classification applications. The first layer represents words into low-dimensional vectors. The second layer performs convolutions over the word vectors with several filter sizes, for example, sliding over 2, 3, or 4 consecutive words at a time. In the third layer, we max-pool the result from the convolutional layer into a long feature vector. Finally, we add dropout regularization, and classify the feature vector in the output layer using softmax function.

2.1 Convolutional Neural Networks

In the following paragraphs of this section, we will describe the Convolutional Neural Network (CNN) model of Kim (2014) which we use as the basis of our CNN approach.

2.1.1 Input layer

We begin with a sentence that has been segmented into words, and then convert sentences to a sentence matrix, whose rows are word vector representations of each word. These might be random vectors or, e.g., outputs from a trained word2vec model.

Suppose that the dimension of word vector is *k* and the length of a given sentence is *s*, the sentence matrix could be denoted by $X \in R^{s\times k}$. The dimensionality of this sentence matrix is $s \times k$, like $6 \times k$ example in Figure 1. Let $x_i \in R^k$ be the k-dimensional word vector corresponding to the ith word in a microblog message or sentence. For example, "刚刚 (just now)" is the first word in this message in Figure 1. This sentence could be represented as

$$X = x_1 \oplus x_2 \oplus \cdots \oplus x_s \quad (1)$$

where $\oplus$ is the concatenation operator.

As illustrated in Figure 1，following Kim (2014), we then effectively treat the sentence matrix as an 'image', and perform convolution on it via linear filters. In most text sentences, there is inherent sequential structure in the data, such as "刚刚 (just now)" followed by "地震 (quaked)". Because rows represent discrete symbols (namely, words), it is reasonable to use filters with widths equal to the dimensionality of the word vectors (i.e., *k*). Thus, we can simply vary the 'height' of the filter, i.e., the number of adjacent rows considered jointly. We also call the "height" of the filter as the size of the filter. For example, in Figure 1, "刚刚 (just now)" $\oplus$ "地震 (quaked)" could be considered as a feature when filter size is equal to two.

2.1.2 Convolutional layer

In this layer, we conduct convolution operation with multiple filter sizes (multiple filters of each filter may be used), then use an activation function general multiple feature maps.

We use $X[i{:}\,j]$ to denote the sub-matrix of the sentence matrix X from row *i* to *j*. Suppose that there is a filter $w \in R^{hk}$, which contains $h \times k$ parameters to be estimated. When we apply the filter on a window of words from x_i to x_{i+h-1}, i.e, the matrix $X[i{:}\, i+h-1]$, an output o_i of the convolution operator is obtained.

$$o_i = w \cdot X\left[i : i+h-1\right] \quad (2)$$

where $i = 1, \cdots, s-h+1$, and $\cdot$ is the dot product between the filter and sub-matrix. An activation operation is also need in this layer. A feature could be produced by Equation (3).

$$c_i = f\left(o_i + b\right). \quad (3)$$

Here $b \in R^{s-h+1}$ is a bias term and f is a non-linear function. We will use the hyperbolic tangent as the activation function like Kim (2014). After we apply the filter to every *h* words, including the convolution operator and the activation operator, a feature map $c \in R^{s-h+1}$ is generated.

$$c = [c_1, c_2, \cdots, c_{s-h+1}] \quad (4)$$

2.1.3 Pooling layer

According to the Equation (4), if the length of the sentence is fixed (zero-padding strategy used where necessary), the length of a feature map is also affected by the filter size. Thus, a pooling function is applied to each feature map to induce a fixed-length vector (Zhang and Wallace, 2015). A commonly-used one is max pooling operation (Collobert et al., 2011) which take the maximum value $\hat{c} = max\{c\}$ as the feature corresponding to this particular filter. The idea is to capture the most important feature, the one with the highest value, for each feature map (Kim, 2014). This pooling scheme naturally deals with variable sentence lengths to general a fix-length feature vector.

By using the max pooling strategy, one feature is extracted from one filter. To obtain multiple features, we simply use multiple filters (i.e., using varying window sizes like 2, 3, or 4). These features could be represented as $z = [\hat{c}_1, \cdots, \hat{c}_m]$ (note that here we have *m* filters).

2.1.4 Output layer

In the final output layer, these features produced by the pooling operation are passed to a function whose output is the probability distribution over labels. This function could be a softmax function as mentioned in literature (Kim, 2014). In our case, there are only two classes of output, so the softmax function could be simplified to a logistic regression function. Given the feature vector $z = [\hat{c}_1, \cdots, \hat{c}_m]$, output unit y could be represented as:

$$y = \frac{1}{1+e^{-(w\cdot z+b)}} \quad (5)$$

2.2 Regularisation

For regularization, Kim (2014) adopted two kinds of regularisation in his CNN, including dropout regularization and l_2-norms regularisation. However, Zhang and Wallace (2015) found that the l_2-norms regularisation has relatively little effect on the performance of the model. Thus, in this paper, in order to simplify the CNN model, we only use the dropout regularisation.

We employ dropout on the penultimate layer (Hinton et al., 2012). Dropout prevents co-adaptation of hidden units

by randomly dropping out, i.e., setting to zero, a proportion p of the hidden units during forward backpropagation. That is, given the penultimate layer feature vector $z = [\hat{c}_1, \cdots, \hat{c}_m]$, instead of using Equation (5), for output unit y in forward propagation, dropout uses

$$y = \frac{1}{1 + e^{-(w \cdot (z \circ r) + b)}} \quad (6)$$

where $\circ$ is the element-wise multiplication operator and $r \in R^m$ is a 'masking' vector of Bernoulli random variables with probability p of being one as Kim (2014) suggested. Gradients are back-propagated only through the unmasked units. At test time, the learned weight vectors are scaled by p such that $\hat{w} = pw$, and $\hat{w}$ is used (without dropout) to score unseen sentences.

2.3 Pre-trained word vectors

For English words, we could have made use of publicly available pre-trained word vectors, such as those produced by Mikolov et al. by training on 100 billion words of Google News[3]. For Chinese words, we were unable to find word vectors publicly and therefore had to create Chinese word vectors ourselves by training a word2vec model on a large Chinese corpus.

The main idea of the word2vec model is to predict the surrounding words in a window of length m for every word. Using a large amount of unannotated plain text, the word2vec learns the relationships between words automatically. The output is a set of vectors; one vector per word. The vector representations express the semantic meaning of words and are very good at encoding dimensions of similarity.

In 2013, Mikolov (together with his colleagues at Google) released the source code of word2vec, an unsupervised algorithm for learning the meaning behind words, followed by the development of many software tools for word2vec. In this paper, we use the word2vec module of Gensim Python package[4] to train our Chinese word vectors. As for the Chinese corpus, we downloaded the publicly available Chinese Wikipedia[5] on July 14th, 2016. This corpus contains around 260,000 articles which have already had all punctuation removed.

After pre-processing and training the model (embedding size: 150; window size: 5; minimum count: 5), we obtained approximately 440,000 word vectors. In our word vector representations, we found some typical features of word2vec. First, the good semantic similarity is found between relevant words. For example, the similarity between the vector of "害怕 (fear)" and the vector of "担心 (worry)" is 0.76. Secondly, the linear relationship is maintained. For example, the vector of "中国 (China)" minus the vector of "北京 (Beijing)" is similar to the vector of "日本 (Japan)" minus the vector of "东京 (Tokyo)".

3 Data preparation

3.1 Data Description

Our datasets were constructed by selecting targeted subsets of public messages that have been collected by the SWIM system from the Chinese Microblog service, Sina Weibo, since 2014. Some of these messages have been labelled manually.

Training dataset

The training dataset contains 2847 messages. It consists of 1430 positive messages that truly indicate the occurrence of an earthquake and 1417 negative messages that include keyword "earthquake" but are not related to a current earthquake (See Table 1). The training dataset can be regarded as the balanced dataset.

Testing Dataset

The testing dataset contains 1494 messages: 227 positive messages and 1267 negative messages. We only choose a part of data samples from the testing dataset to test the performance of our final model.

Both the training and testing dataset were labelled by the same native Chinese speakers according to the same standard (See the examples of Table 1), while they were collected from two different time periods.

3.2 Data pre-processing

Data pre-processing is an important task that must be performed before training classification models. We process our raw microblog messages as follows. The procedure refers to the steps of processing Chinese text in the literature (Robinson et al., 2014).

Step1: Segmentation based on dictionary.

The main difficulty with processing Chinese text is the lack of whitespace between words. Automatic word segmentation on Chinese text has been an active research topic for many years (Nie et al., 1996, Foo and Li, 2004, Gao et al., 2005) resulting in numerous software tools. Examples include the Stanford Word Segmenter[6], the IK Analyzer[7], and ICTCLAS[8] (Zhang et al., 2003). For our classification experiments we used the jieba segmentation package[9] and its default dictionary in Python to segment our messages.

Step 2: Convert the Traditional Chinese to the Simplified Chinese.

Simplified Chinese characters are standardized Chinese characters prescribed in the List of Commonly Used Characters in Modern Chinese for the use in mainland China. The government of the People's Republic of China in mainland China has promoted them for use in printing since 1950s and 1960s.However, the Traditional Chinese is still used in Taiwan, Hong Kong and Macau, as well as in overseas Chinese communities outside Southeast Asia[10] The difference is mostly with the characters, as the name suggests, the Simplified Chinese characters being simpler.

[3] https://code.google.com/archive/p/word2vec/

[4] https://radimrehurek.com/gensim/

[5] https://dumps.wikimedia.org/zhwiki/latest/zhwiki-latest-pages-articles.xml.bz2

[6] http://nlp.stanford.edu/software/segmenter.shtml

[7] https://github.com/wks/ik-analyzer

[8] http://ictclas.nlpir.org/

[9] https://github.com/fxsjy/jieba

[10]https://en.wikipedia.org/wiki/Simplified_Chinese_char-acters

However, both the Simplified and the Traditional Chinese use the same grammar and sentence.

. The main method of dealing with the Traditional Chinese text is to initially convert it to the Simplified Chinese. For example, 'heat' in the Simplified Chinese is 热 while in the Traditional Chinese it is 熱. We used opencc[11] tool to convert the Traditional Chinese to the Simplified Chinese.

Step 3: Remove extraneous punctuation and characters.

For the CNN model, we only remove some extraneous punctuation and characters such as "_", "「", "⊕", "っ" and keep some important punctuation including user mention, hash tag, question mark and exclamation mark. For the SVM classifier, since we will calculate meta information of messages including important punctuations later, all the punctuation and extraneous characters would be removed.

Step 4: Remove the stop words.

Stop word removal is a common pre-processing step for text analysis where a stop word list can be predefined or learned. There are five popular Chinese stop words lists predominantly in use (Zhuang et al., 2012): Harbin Institute of Technology (includes 263 symbols/ punctuation characters and 504 Chinese words); Baidu (includes 263 symbols/ punctuations, 547 English words and 842 Chinese words); Sichuan University Machine Intelligent Lab (includes 975 Chinese words); Chinese stop word list (a combination of the previous three mentioned above, includes 73 symbols/ punctuations, 1113 Chinese words and 9 numbers); Kevin Bouge Chinese (includes 125 Chinese words). These lists have different features with none considered authoritative. For our work, since we have already removed extraneous punctuation and characters, we simply removed words using the Kevin Bouge stop word list.

The optional step (only for the SVM classifier).

Calculate meta information of messages, such as the character count, word count, user mention count, hash tag count, question mark count and exclamation mark count.

4 Results

4.1 Baseline result

To provide a point of reference for the CNN results, we first report the performance achieved using SVM for message classification. A range of message features were explored for the SVM classifier: sentence unigrams after pre-processing (1227 features after removal of stop words and low-frequency words) and other message features, including the character count, word count, user mention count, hash tag count, question mark count and exclamation mark count.

We used two kernels, a linear kernel and a radial kernel, to classify messages. Since the training dataset is balanced, we only report the accuracy of 10-fold cross validation for both classifiers (See Table 2).

The average accuracy of the radial kernel SVM is greater than that of the linear kernel SVM. To check whether this difference is statistically significant, we conducted a one-tail paired-sample *t* test. The *t* value is 2.378 and *p* value is 0.020, which indicates that the accuracy of radial kernel SVM classifier is significantly greater than that of the linear kernel one. Thus, we adopt the result of the radial kernel SVM classifier for our comparison below.

4.2 Improved result

In this section, we experimented with two CNN models: the CNN static model and the CNN non-static model and compared the cross validation results of the CNN models with the baseline SVM result. Both models are briefly described below.

The CNN static model:

A model with pre-trained vectors from word2vec. All words including the unknown ones that are randomly initialised are kept constant and only the other parameters of the model are learned

The CNN non-static model:

This is the same as the static model, however the pre-trained vectors are further tuned during learning.

We used Tensorflow (Abadi et al., 2016) to run the CNN training and evaluating programs in Python. Tensorflow is .an open source software library for numerical computation using data flow graphs developed by the Google Brain Team within Google's Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research[12],

The word vectors described in Section 2.3 were used as pre-trained vectors. Our training dataset contains 11510 different words after pre-processing, of which only 7824 words could be found in the pre-trained word vectors. The other words are randomly initialised with a 150 dimensional vector.

Other hyper parameters settings are filter sizes of 2, 3 and 4, a dropout of 0.5 and 128 filters per filter size. The results of the training programs are shown in the following accuracy plots.

kernel	1	2	3	4	5	6	7	8	9	10	Ave	SD
linear	0.870	0.860	0.874	0.856	0.839	0.842	0.845	0.839	0.873	0.846	0.854	0.014
radial	0.884	0.853	0.849	0.873	0.870	0.860	0.880	0.870	0.874	0.888	0.870	0.013

Table 2: 10-folds cross validation accuracy of SVM

[11]https://code.google.com/archive/p/opencc/wikis/Install.wiki

[12] https://www.tensorflow.org/

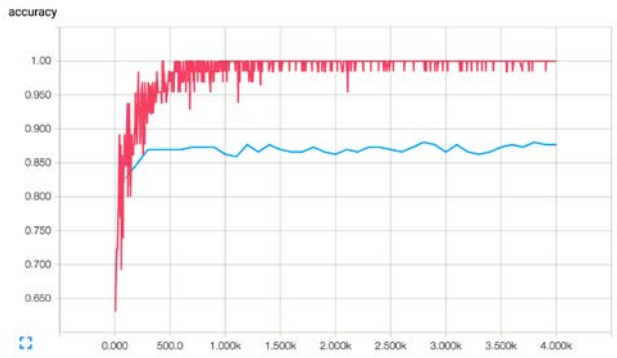


Figure 2: Training accuracy plot of the CNN-static model (red is training data, blue is 10% testing data).

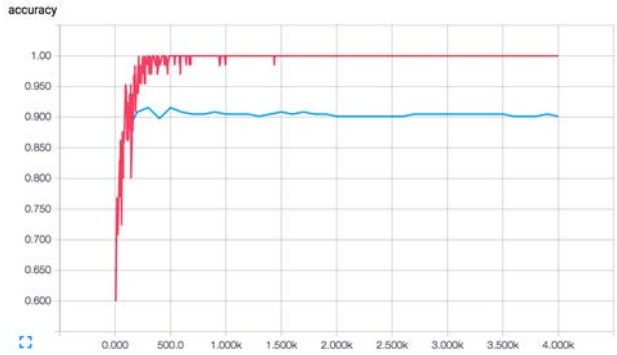


Figure 3: Training accuracy plot of the CNN-nstatic model (red is training data, blue is 10% testing data).

Figure 2 shows the accuracy of every training step for the CNN static Model and Figure 3 shows the accuracy of every training step accuracy for the CNN non-static model. We found that the results tend to stabilise after one or two thousand steps. We therefore made the assumption that 4000 steps are sufficient for training the model.

For each model, we conducted 10-fold cross validation. The mean and standard deviation of the 10-fold cross validation accuracy are presented in Table 3.

model	Mean of Accuracy	SD of Accuracy
static	0.893	0.010
non-static	0.907	0.014

Table 3: 10-folds cross validation accuracy of CNN

The average accuracy of the static model is less than that of the non-static one. In order to check whether this difference is statistically significant, we conduct a one-tail paired-sample *t* test between these two results. The *t* value is 3.422 and *p* value is 0.004, which indicates that the accuracy of the non-static model is significantly greater than that of the static model.

We also compared the accuracy of both CNN classifiers with the baseline SVM classifier. The one-tail paired sample *t*-tests show that the *p*-values is 0.001 for CNN-static against the radial kernel SVM and is 0.0001 for the CNN-non-static against the radial kernel SVM.

These results indicated that both CNN classifiers achieved significantly higher accuracies than that of the SVM classifier. Figure 4 displays the comparison of the accuracies achieved by these three models. It is clear that the accuracy plots for both CNN models are generally higher than that of the SVM model.

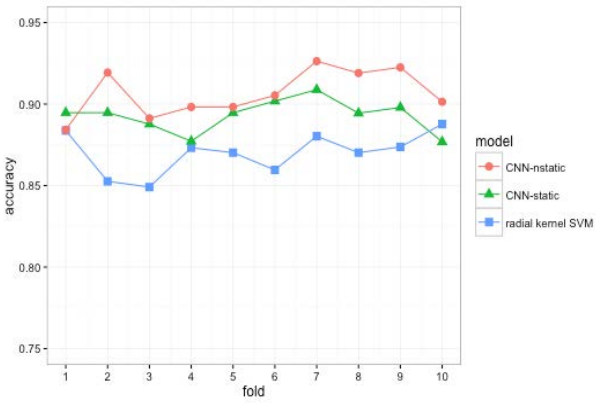


Figure 4: 10-folds cross validation accuracies of the three models.

4.3 Result of new messages testing by using CNN-non-static model

As shown in Section 4.2, the CNN non-static model has the best cross validation accuracy. To test the generalisation ability of the non-static CNN classifier, we used the non-static CNN model trained in Section 4.2 to classify a test dataset. Since our model was trained on a balanced training dataset, we randomly choose 200 positive and 200 negative messages from the testing dataset to test its generality. The results based on 10 independent runs are summarised in Table 4.

	Accuracy	Recall	Precision	F1
Mean	0.916	0.953	0.888	0.919
SD	0.012	0.005	0.018	0.011

Table 4: Testing result of non-static CNN model

Table 4 shows that the CNN non-static classifier we built produces a promising classification result with regard to its generalisation ability, when applied to a balanced testing dataset.

5 More experimental results and discussion

The results in Section 4.3show that our model performs well on a balanced testing dataset. However, in reality, most real-world datasets are, if not extremely, unbalanced. Among all of the messages containing the keyword "earthquake" we collected, the ratio of the number of negative messages to that of positive messages is more than 10.

To simulate a real-word dataset, we randomly choose a subset of the whole testing dataset to form skewed testing datasets as listed in Table 5.

Dataset type (negative-positive ratio)	Number of negative messages	Number of positive messages
1	200	200
2	400	200
3	600	200
4	800	200
5	1000	200

Table 5: Different types of skewed datasets

We randomly generated each type of skewed datasets ten times and tested the classification performance of our CNN model. The F1 and its 95% confidence interval

against the change of negative-positive ratio of the datasets are shown in Figure 5. When the skew ratio is 1, the F1 index is around 0.92. However, when the skew ratio increases to 5, the F1 index falls to around 0.75. It is obvious that the F1 decreases with the increasing skew ratio of dataset. This result shows that the performance of our model deteriorates as the dataset becomes more skewed.

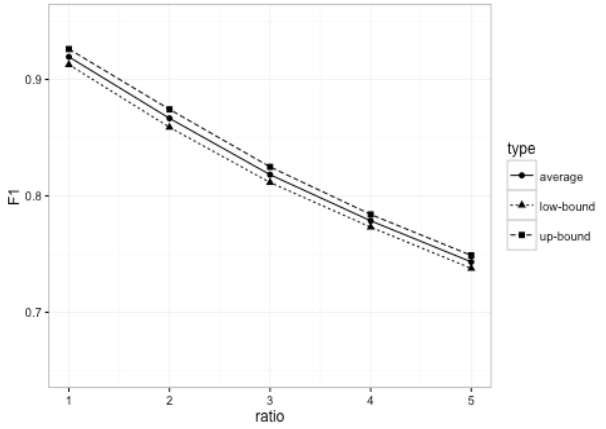


Figure 5: F1 and its 95% confidence interval with the change of the negative-positive ratio

Another issue we encountered is that for our training dataset, we only found 7824 words (68% of all words) in the pre-trained word vectors from Chinese Wikipedia. The other 3686 words were not present in the pre-trained word vectors. Using so many randomly initialised word vectors may reduce the accuracy of the CNN model. This phenomenon suggests that a Chinese Wikipedia corpus is not sufficient for word vector training when applied to the microblog message classification task.

The above mentioned issues will be the focus of our future work.

6 Conclusion

A new microblog message classification method has been presented in this paper, motivated by developing a more accurate and prompt social media disaster event detection systems. We have combined semantic and sequential information of words for Chinese microblog message classification by using a convolution neural network model (CNN) embodied by the pre-trained word vectors. The word vectors were trained on Chinese Wikipedia by using the word2vec algorithm in the Python Gensim package. The CNN model training was implemented using the Tensorflow open source software.

Compared to the widely used SVM model, the proposed classification techniques in this paper improve the cross validation accuracy of the microblog message classification significantly. It also shows good performance when classifying a balanced testing dataset.

However, when it comes to an unbalanced or skewed testing dataset, which is often encountered in reality (Pang et al., 2015), the performance may deteriorate depending on the skewness of the dataset, since our model was trained on a balanced dataset. In addition, only 68% words of the training dataset could be found in the word vectors trained on our currently used corpus, which may affect the accuracy of the CNN model. Therefore, we plan to address the unbalanced issues, perhaps by using an unbalanced training dataset or adding new features to the model, and introduce a new corpus in the future.

7 References

ABADI, M., AGARWAL, A., BARHAM, P., BREVDO, E., CHEN, Z., CITRO, C., CORRADO, G. S., DAVIS, A., DEAN, J. & DEVIN, M. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.

ABEL, F., HAUFF, C., HOUBEN, G.-J., STRONKMAN, R. & TAO, K. Twitcident: fighting fire with information from social web streams. Proceedings of the 21st International Conference on World Wide Web, 2012. ACM, 305-308.

AVVENUTI, M., CRESCI, S., MARCHETTI, A., MELETTI, C. & TESCONI, M. EARS (earthquake alert and report system): a real time decision support system for earthquake crisis management. Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014. ACM, 1749-1758.

BAI, H., LIN, X., ROBINSION, B. & POWER, R. 2015. Sina Weibo Incident Monitor and Chinese Disaster Microblogging Classification. Journal of Digital Information Management, 13, 157-161.

CAMERON, M. A., POWER, R., ROBINSON, B. & YIN, J. Emergency situation awareness from twitter for crisis management. Proceedings of the 21st International Conference on World Wide Web, 2012. ACM, 695-698.

CHOWDHURY, S. R., IMRAN, M., ASGHAR, M. R., AMER-YAHIA, S. & CASTILLO, C. Tweet4act: Using incident-specific profiles for classifying crisis-related messages. 10th International ISCRAM Conference, 2013.

CNNIC. 2016. The 36th Statistical Report on Internet Development in China [Online]. http://www.apira.org/data/upload/The36thSurveyReport_oTmyiO.pdf. [Accessed 9 Aug 2016].

COLLOBERT, R., WESTON, J., BOTTOU, L., KARLEN, M., KAVUKCUOGLU, K. & KUKSA, P. 2011. Natural language processing (almost) from scratch. Journal of Machine Learning Research, 12, 2493-2537.

DOS SANTOS, C. N. & GATTI, M. Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts. COLING, 2014. 69-78.

DU, L., BUNTINE, W. & JIN, H. 2010. A segmented topic model based on the two-parameter Poisson-Dirichlet process. Machine learning, 81, 5-19.

DU, L., BUNTINE, W., JIN, H. & CHEN, C. 2012. Sequential latent Dirichlet allocation. Knowledge and information systems, 31, 475-503.

EARLE, P. S., BOWDEN, D. C. & GUY, M. 2012. Twitter earthquake detection: earthquake monitoring in a social world. Annals of Geophysics, 54.

FOO, S. & LI, H. 2004. Chinese word segmentation and its effect on information retrieval. Information Processing & Management, 40, 161-190.

GAO, J., LI, M., WU, A. & HUANG, C.-N. 2005. Chinese word segmentation and named entity recognition: A pragmatic approach. Computational Linguistics, 31, 531-574.

GOLDBERG, Y. 2015. A primer on neural network models for natural language processing. arXiv preprint arXiv:1510.00726.

GRAVES, A., MOHAMED, A.-R. & HINTON, G. Speech recognition with deep recurrent neural networks. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, 2013. IEEE, 6645-6649.

HINTON, G. E., SRIVASTAVA, N., KRIZHEVSKY, A., SUTSKEVER, I. & SALAKHUTDINOV, R. R. 2012. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580.

HUGHES, A. L., PETERSON, S. & PALEN, L. 2014. Social media in emergency management. Issues in Disaster Science and Management: A Critical Dialogue Between Scientists and Emergency Managers. FEMA in Higher Education Program.

IYYER, M., MANJUNATHA, V., BOYD-GRABER, J. & DAUMÉ III, H. Deep unordered composition rivals syntactic methods for text classification. Proceedings of the Association for Computational Linguistics, 2015.

KALCHBRENNER, N., GREFENSTETTE, E. & BLUNSOM, P. 2014. A convolutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188.

KIM, Y. 2014. Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

KRIZHEVSKY, A., SUTSKEVER, I. & HINTON, G. E. Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 2012. 1097-1105.

LECUN, Y., BOTTOU, L., BENGIO, Y. & HAFFNER, P. 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86, 2278-2324.

MIKOLOV, T., CHEN, K., CORRADO, G. & DEAN, J. 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

MIKOLOV, T. & DEAN, J. 2013. Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems.

NIE, J., BRISEBOIS, M. & REN, X. On Chinese text retrieval. Proceedings of the 19th annual international ACM SIGIR Conference on Research and Development in Information Retrieval, 1996. ACM, 225-233.

OCHA. 2015. World Humanitarian Data and Trends [Online]. http://www.unocha.org/humanity360/. [Accessed 15 Aug 2016].

OLTEANU, A., VIEWEG, S. & CASTILLO, C. What to expect when the unexpected happens: Social media communications across crises. Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing, 2015. ACM, 994-1009.

PANG, G., JIN, H. & JIANG, S. 2015. CenKNN: a scalable and effective text classifier. Data Mining and Knowledge Discovery, 29, 593-625.

PENNINGTON, J., SOCHER, R. & MANNING, C. D. Glove: Global Vectors for Word Representation. EMNLP, 2014. 1532-43.

ROBINSON, B., BAI, H., POWER, R. & LIN, X. Developing a Sina Weibo incident monitor for disasters. Australasian Language Technology Association Workshop 2014, 2014. 59-68.

ROGSTADIUS, J., VUKOVIC, M., TEIXEIRA, C., KOSTAKOS, V., KARAPANOS, E. & LAREDO, J. A. 2013. CrisisTracker: Crowdsourced social media curation for disaster awareness. IBM Journal of Research and Development, 57, 411-413.

SAKAKI, T., OKAZAKI, M. & MATSUO, Y. Earthquake shakes Twitter users: real-time event detection by social sensors. Proceedings of the 19th international conference on World wide web, 2010. ACM, 851-860.

SAKAKI, T., OKAZAKI, M. & MATSUO, Y. 2013. Tweet analysis for real-time event detection and earthquake reporting system development. IEEE Transactions on Knowledge and Data Engineering, 25, 919-931.

THELWALL, M. & STUART, D. 2007. RUOK? Blogging communication technologies during crises. Journal of Computer‐Mediated Communication, 12, 523-548.

WANG, N., SHE, J. & CHEN, J. How big vs dominate Chinese microblog: a comparison of verified and unverified users on sina weibo. Proceedings of the 2014 ACM Conference on Web Science, 2014. ACM, 182-186.

WANG, P., XU, J., XU, B., LIU, C., ZHANG, H., WANG, F. & HAO, H. Semantic clustering and convolutional neural network for short text categorization. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, 2015. 352-357.

ZHANG, H., YU, H., XIONG, D. & LIU, Q. HHMM-based Chinese lexical analyzer ICTCLAS. Proceedings of the 2nd SIGHAN Workshop on Chinese Language Processing, 2003. Association for Computational Linguistics, 184-187.

ZHANG, Y. & WALLACE, B. 2015. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. arXiv preprint arXiv:1510.03820.

ZHUANG, T., WANG, P. & CHENG, Q. 2012. Temporal Related Topic Detection Approach on Microblog. Journal of Information Resources Management, 3, 40-46.

TSIM: Topic-based Social Influence Measurement for Social Networks

Asso Hamzehei[1,3] **Shanqing Jiang**[1] **Danai Koutra**[2] **Raymond K. Wong**[1] **Fang Chen**[1,3]

[1] School of Computer Science and Engineering
University of New South Wales,
Sydney, Australia
Email: assoh,wong@cse.unsw.edu.au,
shanqing.jiang@student.unsw.edu.au

[2] University of Michigan
Ann Arbor, Michigan, USA,
Email: dkoutra@umich.edu
[3] Data61, CSIRO, Sydney, Australia
Email: fang.chen@data61.csiro.au

Abstract

Social science studies have acknowledged that the social influence of individuals is not identical. Social networks structure and shared text can reveal immense information about users, their interests, and *topic-based* influence. Although some studies have considered measuring user influence, less has been on measuring and estimating topic-based user influence. In this paper, we propose an approach that incorporates network structure, user-generated content for topic-based influence measurement, and user's interactions in the network. We perform experimental analysis on Twitter data and show that our proposed approach can effectively measure topic-based user influence.

Keywords: Topic-based social influence, Social networks analysis, influence measurement

1 Introduction

Although social influence has been an area of interest for researchers in sociology and more recently in computer science, still there is no agreement on its definition. A very early definition for influential people is "individuals who were likely to influence other persons in their immediate environment" (Katz 1957). Social influence has either been studied to identify influential users (opinion leaders or authorities), topical or topic-based influential users (Riquelme 2015).

Social science studies, e.g.(Katz & Lazarsfeld 1955), have acknowledged the fact that the social influence of individuals is not identical. Katz (Katz 1957) introduced three main factors that are related to an individual's social influence such as: Who one is, what one knows, and whom one knows. The individual's social influence can be much more easily observed on social media while it is confirmed that the social influence factors are similar in social networks to those in the real society (Libai et al. 2010, Eccleston & Griseri 2008). For example, Eirinaki et al (Eirinaki et al. 2012) introduced two factors (popularity and activity) as factors related to social influence on Online Social Networks (OSN).

One of the main measures studied for influence is information diffusion which measures how important a user is in spreading information in the network. This is equivalent to identify central and hub nodes in the network (Jin & Wang 2013, Hajian & White 2011). Opinion leaders and discussion starters also have been studied as a measure of social influence (Jabeur et al. 2012). A user's position in the network (Jin & Wang 2013), content (Hu et al. 2013), and activities (Pal & Counts 2011) have been also studied as influence measures. Another aspect of studied influence has been the scale of affected users by a post on social network or intensity of emotional and cognitive impact (McNeill & Briggs 2014).

According to (Probst et al. 2013), influential users have different influences on different topics and a very influential user is not necessarily influential on all topics. It is indicated in (Kardara et al. 2015) that topic-based influence measures are more effective and functional than the global ones. One of the differences of topic-related influence studies to network structure analysis is that it takes the posts' (e.g., tweets) content into account. When we consider user influence on topics, no longer the whole network needs to be analyzed, which improves the performance of measures.

However, there are drawbacks and shortcomings in the topic-based influence studies. In most of the existing works, they have aimed at making influential user detection more effective in retrieving the top N users only. Less effort is dedicated in discriminating influential from non-influential users. Also, approaches that uses supervised learning (e.g., SVM) suffer from

their dependency on labeled data, which is extremely expensive to prepare for the immense data of social networks. Another considerable issue in these studies is their approach evaluation. This is a difficult task as influence is subjective. More importantly, prediction of user influence is remained as a problem to address in the state-of the-art.

Topic-based user influence measurement and identification are important challenges and the focuses of this paper. This task is significantly important for different applications such as marketing, election campaigns, or recruiting employees for a company. In this work, we measure topic-based user influence on observed topics in which they have shown their interests by posting in social networks. Our approach, called TSIM (Topic-based Social Influence Measurement), incorporates network structure, user generated contents, users history of activities, and network users engagement in user's activity. Our approach represents users with their topic interests and their social influence on each observed topic.

In more detail, our contributions are:

- We propose a novel topic-based influence measurement approach to integrate the user-topic relationships, topic content information, and social connections between users into the same principled model.
- Instead of considering user-to-user influence and global user influence, the proposed model considers individuals influence and interests in a topic, which gives the capability of predicting one's influence on a new topic.
- Finally we have prepared a unique dataset from real-world social networks for testing and evaluating the proposed approach that contains all the social media related metadata.

The remainder of this paper is organized as follows. We first discuss existing approaches for topic-based influence analysis in Section 2. We then present the background in Section 3. Next, we define the research problem, and then propose our approach and algorithms in Section 4. We describe our dataset and discuss the results in Section 5. Finally, we conclude the paper in Section 6.

2 Related Work

One of the main approaches to study user influence in social networks has been through network structure as well as user's position and connectivity in the network. The traditional centrality measures such as closeness and betweenness are measured for users, to discover how well connected a user is to the rest of users in the network and whether a user is acting as a hub (Romero et al. 2011). The major adopted algorithms for network structure based influence measurement include PageRank (Haveliwala 2002) and HITS (Kleinberg 1999). Numerous works have applied PageRank algorithm variations on social network graph to rank user influence according to the network structure. An example of PageRank algorithm variations is the work by Kwak et al (Kwak et al. 2010), in which they ranked users by applying PageRank on follower/following graph in Twitter (along with number of followers and number of retweets). The network structure is relatively static compared to the activities of users in social networks. Some studies have included the social network related meta data (in case of Twitter, the meta data are retweets, mentions, and likes) (Hajian & White 2011).

Topic-based Influence. Following the influence studies (overall user influence) on social networks, less studies have shed light on topic-based influence. More recently, topic-based influence studies have combined content of user posts with link-based metrics. Haveliwala (Haveliwala 2002) proposed a topic-sensitive extension of PageRank to rank query results in regards to the query topics. The idea of topic-sensitive PageRank was later used and adjusted for social networks such as Twitter for ranking topic-based user influence. Topical authorities were also studied in (Pal & Counts 2011) by Pal et al. They proposed a Gaussian-based ranking to rank users efficiently. They used probabilistic clustering to filter feature space outliers and showed that mentions and topical signals are more important features in ranking authorities. In (Kong & Feng 2011), Kong et al intended to identify and rank users that are posting quality tweets. They defined a topic-based high quality tweet with the author's topic-specific influence, topic related author's behavior. They applied their proposed metric on graph of following and retweets. Xiao et al (Xiao et al. 2014) aimed at detecting topic related influential users by looking at hashtag user communities where hashtags are pre-identified from news keywords. They proposed RetweetRank an MentionRank as content-based and authority-based influential users. Similarly, (Hu et al. 2013) worked on detecting topical authorities with the assumption that retweeting propagates topical authority. Montangero and Furini (Montangero & Furini 2015) also measured Twitter topic-based user influence where they identify topics by hashtags. Although hashtags can reveal the tweet's topic correctly, over 80% of tweets do not have hashtags. These results are neglecting the majority of tweets and can mislead a topic-based user influence, as 4 out of 5 of her tweets are not considered for measuring her influence. In (Cataldi & Aufaure 2015), they estimated Twitter user influence for topics of conversations based on PageRank. For that purpose they build a topic information exchange graph to take the information diffusion and degree of information shared into account for user influence estimation. They manually considered seven topic categories and later assign each tweet to those categories through an n-gram model. However, their approach is unable to identify topics in the lower level of the main categories. For example, if someone is detected as influential in the sports category we do not know which sport the influence belongs to. In (Weng et al. 2010), they offered TwitterRank, a PageRank extension, that measures user influence by calculating topical similarities of users and their network connections. For topic identification, they used the unsupervised text categorization technique, LDA, by aggregating

Table 1: Key notations

Symbol	Description
t	a topic
$F_{i,j}$	influence of user u_i in t_j
$F_f(i,j)$	follower strength influence measure for user i in t_j
$F_a(i,j)$	activity influence measure for user u_i in t_j
$F_e(i,j)$	engagement influence measure for user u_i in t_j
$F_c(i,j)$	centrality influence of measure for user u_i in t_j
X_i	rank of user u_i calculated through PageRank
I^{op}	operation of identifying authorities
L	asymmetric adjacency matrix representing directed edges
$P_{i1,i2}$	probability of user u_{i1} engage with user u_{i2}
$D_{out}(i)$	number of user that points to user u_i in graph G
N_d	number of words in document d

all tweets of a user into a document. Although this approach is presented as topic-sensitive, this approach cannot discriminate the user influence for the topics. In (Sung et al. 2013) they proposed another extension of PageRank, and unlike (Weng et al. 2010), it does not need predefined topics for topic-based user influence. In (Cano et al. 2014) a PageRank-based user influence rank algorithm introduced that the user links have weights based on their topics of interest similarities. In (Liu et al. 2014), their topic-based influence framework considers retweet frequency and link strength. The link strength is estimated by poisson regression-based latent variable model on user's frequency of retweeting each other. In (Welch et al. 2011), they found out that topical relevance is better detectable through the retweet link rather than following links. They used two variations of PageRank algorithm to on retweet and following graphs for that purpose. In a recent work by Katsimpras et al (Katsimpras et al. 2015), they proposed a supervised random walk algorithm for topic sensitive user ranking. As it is obvious from the algorithm name, it needs labeled data which is not very practical in many cases specially with the volume of social networks.

It is worth mentioning that similar works exist that are only after the identification of global influencers instead of influencers for specific topics. An example of such works is (Barbieri et al. 2012) where they extended the Linear Threshold Model and Independent Cascade Model to be topic-aware, the topics are still obtained based on the network structure, while totally ignoring the valuable content information.

3 Background

Next, we give preliminaries for Probabilistic Topic Modeling and Pagerank.

3.1 Probabilistic Topic Modeling

Given a set of documents denoted by $D = [d_1, \ldots, d_q]$, Topic Modeling generates a set of t topics denoted by $\mathcal{T} = [t_1, \cdots, t_j]$. Each topic is related to a weighted representation over m words denoted by $t_j = [w_1 \cdots w_m]$, where w_j is the weight representing the contribution of word w_m to topic t_j. Probabilistic topic modeling , such as Latent Dirichlet Allocation (LDA), represents a low dimensional space of corpus by detecting a set of latent topics. The basic idea of Probabilistic Topic Modeling is having a Z hidden variable for each word's co-occurrence in the collection of documents. Z can range among j topics where each topic is a distribution over a fixed vocabulary. Given a corpus, a document may contain multiple topics and the words are assumed to be generated by those topics. A probabilistic topic model can be generated over a process as follows:

1. Obtain a distribution over topics to generate a document (in LDA this distribution is drawn from a Dirichlet distribution with a corpus-specific hyperparameter α)
2. Then for each word to be generated;
   (a) Assign topics by drawing upon the document-specific distribution over topics
   (b) Finally, generate a word from distribution of topics over words in dictionary, which means words of each document come from a mixture of topics.

We aim to use probabilistic topic modeling to represent items as a set of topics and also detect social network users interest by applying topic modeling on their timelines.

3.2 PageRank

PageRank is a webpages ranking algorithm that calculate rank X_i for vertex v_i based on the rank of other vertices in the graph that point to vertex v_i. Assume $G(\mathbb{V}, E)$ denotes a directed graph, where the set $\mathbb{V}$ of vertices consists of i users and users relationships are the edges set E. Considering u_i as a user equal to vertex v_i in the graph G, the directed edge $(i1, i2)$ exists if user u_{i1} is connected to user u_{i2}. The directed vertices of the graph G contained in the asymmetric adjacency matrix $L = (L_{i1,i2})$, where $L_{i1,i2} = 1$ if $u_{i1} \rightarrow u_{i2}$ and $L_{i1,i2} = 0$ otherwise. Out-degree $D_{out}(i)$ is the number of users that points to user u_i.

$$X_i = \sum_{(j,i) \in E} D_{out}(j)^{-1} X_i \tag{1}$$

The above equation is a recursive function that gives any vertex points to vertex v_i, a fraction of the rank inversley.

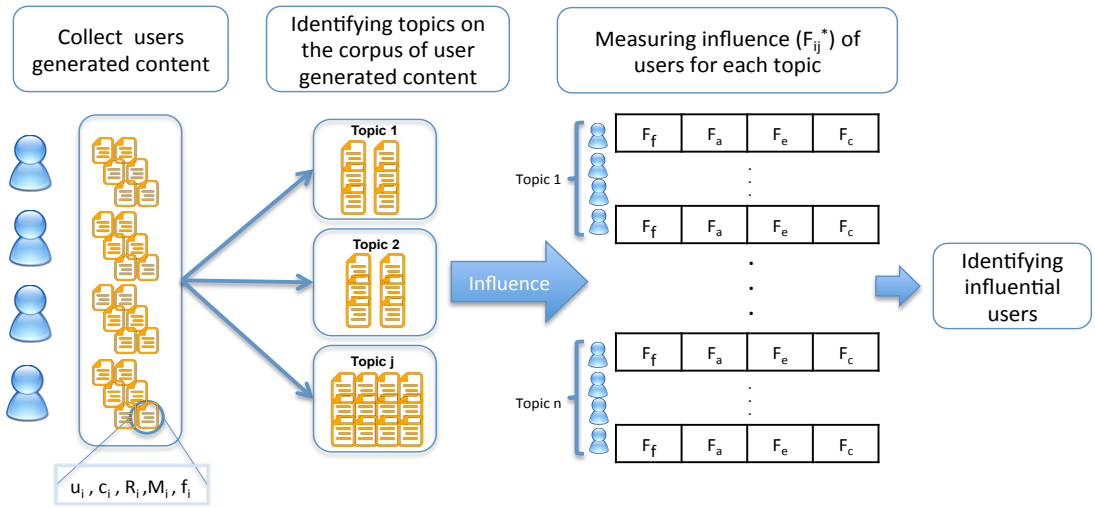


Figure 1: TSIM WorkFlow.

In PageRank, each out-going link from v_i is weighted by $1/o_i$, thus every node has the same total out-going weights. Each node has a total of one vote. PageRank uses an idea that a "good" node should connect to or be pointed to by other "good" nodes. However, instead of mutual reinforcement, it adopts a web surfing model based on a Markov process in determining the scores:

$$x = I^{op}(x) \tag{2}$$

where the I^{op} is an authority that is pointed to by many hubs and the I^{op} operation is defined to be

$$I^{op}(.) = L^T D_{out}^{-1} \equiv P^T. \tag{3}$$

This amounts to rescale the adjacency matrix L such that each row is sum-to-one. Thus, $P = (P_{i1,i2})$ is a stochastic matrix, since $\sum_v P_{i1,i2} = 1, P_{i1,i2} \geq 0.P_{i1,i2}$ represents the probability of a web surfer making a transition from webpage v_{i1} to v_{i2} . Starting from any webpage v_i, a surfer goes to any one of the hyperlinked webpages with equal probability $1/o_i$.

At any moment, millions of people are using the social networks. PageRank assumes the users follow the random surfing model in viewing and engaging with the rest of network. They will reach the equilibrium (stationary) distribution under general conditions. If a node has a high probability in the equilibrium distribution, that means more nodes will point to that node. Therefore, the equilibrium distribution of users in social network is a measure of a node's importance, which is the authority score in PageRank. The equilibrium distribution x is determined by

$$P^T x = \lambda x \tag{4}$$

and x satisfies $\sum_k x(k) = 1$. One can obtain the solution iteratively. Note that $\lambda = 1$ if the Markov process has an equilibrium distribution x. PageRank models two types of random jumps on the Internet.

(i) Link-tracking jump: a user often follows other users in the network by simply clicking on them; this is modeled by $L^T D_{out}^{-1}$.

(ii) Link-interrupt jump: a user sometimes observe to engage with a user that they are not already connected to each other. PageRank models such link-interrupt jump with a simple uniform distribution $(1 - \alpha)/n$. The full stochastic matrix of transition probability is

$$P^T = I^{op}(.) = \alpha L^T D_{out}^{-1} + (1 - \alpha)(1/n)ee^T \tag{5}$$

where $\alpha = 0.8 \sim 0.9$. Here $e = (1, 1, , 1)^T$; thus ee^T is a matrix of all 1's (Arasu et al. 2002).

4 Topic-based Social Influence Measurement

4.1 Problem Definition

Assume $G(\mathbb{v}, E)$ denotes a social network graph, where users are the vertex set $\mathbb{V}$ and users relationships are the edges set of E. Assume that users publish a set of texts $D = [d_1, d_2, \ldots, d_q]$, and talk about different topics $\mathcal{T} = [t_1, t_2, \ldots, t_j]$. Each user text (post) d_q holds one or more topics and receives engagement from other users by replying, liking, or republishing it. The engagement of other users in a post can reveal the influence of that particular post among its audience.

We denote $A = \{a_{ii'}\}$ as the $n \times n$ matrix which shows the social ties among users in the social network G. For the pair of users i and i', $a_{ii'} \in [0, 1]$ shows the weight of the relationship between users u_i and $u_{i'}$, which we treat as the influence of user i on user i' (the higher the value of $a_{ii'}$, the higher the corresponding influence). The matrix A is not symmetric, as the influence of user i on user i' is not necessarily equal to influence of user i' on user i. We also assume that user post is visible to all users in G.

Quantifying the topic-based influence of each user based on social ties and other users' engagement in social networks, we can identify the influence of user i on topic j, represented as F_{ij}. Then we have matrix $F = [F_{ij}]_{i \times j}$ that represents influence of all the users in all identified topics.

Table 2: A sample from the influence matrix *before* aggregating the 4 influence measures of user u_i in topic t_j.

User	*Topic1*	*Topic2*	*Topic3*	*Topic4*
vnfrombucharest	[0, 0, 0, 0]	[0, 0, 0, 0]	[0.1, 0.02, 0.51, 0.008]	[0, 0, 0, 0]
CharlieDataMine	[0.12, 0.01, 0.34, 0.16]	[0, 0, 0, 0]	[0.1, 0.014, 0.511, 0.163]	[0.28, 0.198, 0.38, 0.16]
sepehr125	[0, 0, 0, 0]	[0, 0, 0, 0]	[0, 0, 0, 0]	[0, 0, 0, 0]
sDataManagement	[0.12, 0.02, 0.35, 0.67]	[0.05, 0.04, 0.15, 0.67]	[0.6, 0.042, 0.512, 0.674]	[0, 0, 0, 0]
yisongyue	[0, 0, 0, 0]	[0, 0, 0, 0]	[0, 0, 0, 0]	[0.07, 0.05, 0.27, 0.08]

Table 3: A sample from the influence matrix *after* aggregating the 4 influence measures of user u_i in topic t_j.

User	*Topic1*	*Topic2*	*Topic3*	*Topic4*
vnfrombucharest	0	0	0.159	0
CharlieDataMine	0.157	0	0.197	0.254
sepehr125	0	0	0	0
sDataManagement	0.29	0.227	0.457	0
yisongyue	0	0	0	0.117

4.2 Our Approach

To measure social influence on an observed topic in a social network, we propose TSIM, Topic-based Social Influence Measure, which measures topic-based individuals influence in social networks. In a nutshell, our model contains two main phases:

- Identifying topics on social networks according to users generated contents, and
- measuring individuals influence for the detected topics.

Figure 1, shows our approach's work flow. First, we collect user-generated content from the social media. For each user-generated content, we collect related information such as list of users that have re-published the content (R_i)and list of users that have engaged in that content (M_i) as well as metadata connected to the content. We identify the topics by applying probabilistic topic modeling, LDA, on all the user-generated text. Each topic contains a set of posts with all their related information and metadata, such as; content, replies, and republishing. For each tuple of $(user_i, topic_j)$, we measure the influence of user u_i on topic t_j as F^*_{ij} which comprises of four measures F_f, F_a, F_e, F_c. The details of influence measurement shown in Algorithm 1 and Section 4.3. The measure F^*_{ij} identifies user influence for the identified topics and users can be ranked according to their F^*_{ij} score for each topic.

4.3 Influence Measurement

We define social influence in a social network as importance of a user in the social network graph, user's activities, and involvement of others in the user's posts. Social influence can be analyzed through different modalities network structure and user's position in the network, scale of a user's post diffusion in the network, a user's activities and engagement in the social network, and message content that a user broadcast in the network(Embar et al. 2015).

From the network structure, we identify influence related attributes, such as user friends and centrality of user in the social network. From the content of broadcasted text, we can identify one or more topics, thus, the influence of that user on different aspects. For instance, in Twitter, a post can contain user mentions, receive replies, and get retweeted by other users. All this information can reveal social influence of a user.

Algorithm 1 Influence Measurement

Input: List of topics, collection of user posts for each topic, interaction graphs, number of friends of each user.
Output: Matrix of user influence on each topic.

1: **for** topic in *topics* **do**
2: **for** user in *users* **do**
3: $F_f(i) \leftarrow \#friends$
4: $F_a(i,j) \leftarrow \sum_{d_i \in D_t} \delta(d_i)$
5: $F_e(i,j) \leftarrow \sum_{d_i \in D_t} (\delta(R_i) + \delta(M_i))$
6: $F_c(i,j) = PR(u_i, G(D_t))$
7: $F^*_{ij} \leftarrow$
8: aggregation of $F_f(i,j), F_a(i,j), F_e(i,j), F_c(i,j)$
9: Return matrix of user influence on topics

Let denote D_t as the set of collected texts related to topic t_j from the set of topics $\mathcal{T}$. Each text d_i contains a set of attributes as $(u_i, c_i, R_i, M_i, f_i)$ where u_i is the author of the text, c_i is the text, R_i is the list of users republished the text, M_i is the list of mentions for that text, and f_i is the number of followers of the text author.

We define the following dimensions for measuring social influence of a user on a topic as following:

Follower scale: This measure depicts the number of friends a user has in the network. This value is constant across all topics for a user and is independent of topics. It shows the strength of social ties of a user. Although the number of social connections can be an indicative of influence, it does not carry information on any specific topic. The following influence measures are more topic-specific.

Topic Activity: This measure captures topic-related activities of a user. $F_a(i,j)$ denotes influence of user u_i in terms of activities related to topic t_j and we define it as:

Table 4: A sample of topics and their 5 top influencers measured by our proposed topic-based influence measurement system.

Deep Learning		*Text Mining*		*Programming Languages*		*Artificial Intelligence*	
Screen Name	*F*	*Screen Name*	*F*	*Screen Name*	*F*	*Screen Name*	*F*
kdnuggets	0.63	randal_olson	0.62	analyticbridge	0.70	analyticbridge	0.57
analyticbridge	0.49	analyticbridge	0.55	randal_olson	0.49	ML_toparticles	0.55
deeplearning4j	0.33	jmgomez	0.53	DataScienceCtrl	0.41	DataScienceCtrl	0.37
KirkDBorne	0.31	IBMbigdata	0.51	BernardMarr	0.35	IBMbigdata	0.24
DataScienceCtrl	0.31	kdnuggets	0.49	eddelbuettel	0.34	kdnuggets	0.21

Table 5: Table 4 continued- a sample of topics and their 5 top influencers measured by our proposed topic-based influence measurement system.

NLP-BigData		*Neural Networks*		*Social Networks*		*R and Stats*	
Screen Name	*F*	*Screen Name*	*F*	*Screen Name*	*F*	*Screen Name*	*F*
jmgomez	0.51	kdnuggets	0.93	analyticbridge	0.58	kdnuggets	0.62
randal_olson	0.48	KirkDBorne	0.43	kdnuggets	0.56	analyticbridge	0.54
analyticbridge	0.45	smolix	0.34	mjcavaretta	0.47	randal_olson	0.47
stanfordnlp	0.43	mapr	0.32	CharlieDataMine	0.44	DataScienceCtrl	0.36
bigdata	0.36	mjcavaretta	0.30	jure	0.35	paulblaser	0.36

Table 6: Table 5 continued- a sample of topics and their 5 top influencers measured by our proposed topic-based influence measurement system.

Recommender Systems		*BigData-Hadoop*		*Database*		*Visualization*	
Screen Name	*F*	*Screen Name*	*F*	*Screen Name*	*F*	*Screen Name*	*F*
xamat	0.70	analyticbridge	0.69	analyticbridge	0.58	analyticbridge	0.90
analyticbridge	0.54	mapr	0.60	randal_olson	0.55	DataScienceCtrl	0.45
kdnuggets	0.40	BernardMarr	0.58	IBMbigdata	0.36	hmason	0.42
jmgomez	0.36	odbmsorg	0.56	OracleAnalytics	0.32	KirkDBorne	0.35
KirkDBorne	0.32	infochimps	0.53	MarkLogic	0.30	paulblaser	0.31

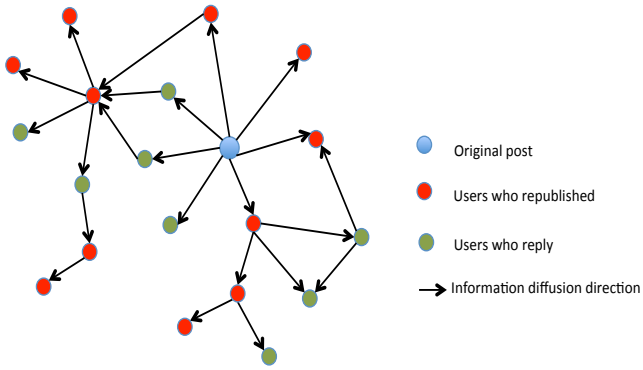


Figure 2: Retweet and Mention Graph.

$$F_a(i,j) = \sum_{d_i \in D_t} \delta(d_i) \tag{6}$$

where $\delta(d_i)$ is 1 if d_i belongs to texts set for topic t and is 0 otherwise. It intuitively measures the volume of topic t_j-related activities of user u_i.

Topic-based Attractiveness: This measure indicate how other users are attracted to user$i$'s post. It takes other users' feedback on user u_i's activities into account. We define it as

$$F_e(i,j) = \sum\nolimits_{d_i \in D_j} (\delta(R_i) + \delta(M_i)) \tag{7}$$

where $\delta(R_i)$ is the number of times d_i is republished by other users and $\delta(M_i)$ is the number of mentions or replies of d_i.

Network centrality: Centrality of a user is another indicator of her influence in a social network. PageRank was introduced first for ranking webpages for search engines, and can be used here to calculate topic-specific centrality of users in the social graph. Figure 2 shows an interaction graph of users on a post generated by user 1. To that end, we perform PageRank on the induced graph of interactions on a specific topic t_j. The interaction graph is a better representative of the topical relevance of two users rather than friendship graph (Welch et al. 2011). We denote it as:

$$F_c(i,j) = PR(u_i, G(D_t)) \tag{8}$$

where $G(D_t)$ is a graph corresponding to users over documents set D_t for topic t. $PR(u_i, G(D_t))$ indicates the PageRank score of user u_i in the graph $G(D_t)$. In this work, we reconstruct the interaction graph, (e.g., retweet and mention graphs from Twitter), to measure topic specific centrality of users by PageRank.

Aggregating Influence Scores: The four influence measures described above F_f, F_a, F_e, F_c will be aggregated to form a single influence score F^* for user u_i in topic t_j. For the first attempt, we averaged the measures which gives every measure the same share in the overall influence score. For the future works, we investigate other methods for aggregating the measures.

5 Results and Experiments

In this section, we discuss the the details of conducted experiments. It includes the data and the influence measurement performed by our proposed method.

5.1 Dataset

To validate our proposed method, we collected a unique dataset from Twitter using the Twitter Search API. We targeted the Machine Learning domain and identified core 500 users that have mentioned machine learning as a keyword in their profile description. To choose the users, we selected a set of machine learning users as seeds and crawled among their friends and friends of friends for other machine leaning-related users. For the prepared list of users, we gathered their timeline tweets which for most of the users covers their tweets for the last 5 years. For each tweet, we also, collected the related meta-data such as the list of users who have replied to each tweet (mention list) and the list of users who have retweeted each tweet (retweet list). The final dataset contains 101,363 tweets with their related metadata, mention lists, and retweet lists. The network that is built on retweet list contains 301870 nodes.

5.2 Evaluation

Our experiments contains a main task of user influence measurement on the identified topics from the tweet corpus.

We evaluate the measured user influence through expert opinion and user citations on the topics that the user has published in scientific conferences and journals. We collected publications through Google scholar for validation. The community of study is intentionally chosen as researchers then we are able to cross-validate our results through the users influence in research community measured by topics of their publications and citations.

5.3 Topic-based Influence Measurement

Next, we proceed with identifying topics from the collection of all tweets and then measuring influence. The number of topics generated by LDA can affect the quality of features that will be used in TSIM. We determined a number of topics through cross validation that we could receive higher recall in user influence prediction. In our proposed approach, we perform probabilistic topic modeling for identifying the topics in the tweets dataset.

The user tweets gathered from their timelines, belong to the identified topics with a probability. We set the probability threshold to 0.1 to consider whether a tweet belongs to a topic. Each tweet is mapped to at least one topic. Now that for each topic we have a collection of related tweets with their mention and retweet lists, we can measure user influence for them. In Section 4.3, we defined influence based on 4 measures; follower strength, activity, engagement, and network centrality. Follower strength will be taken from the number of users follow the user u_i on Twitter. Activity represents the number of tweets user u_i has in topic t_j. Engagement is the sum of number of mentions and retweets for all of user u_i's tweets in topic t_j. For measuring network centrality, we build the retweet graph for each topic separately from the corresponding retweet list and measure centrality of that user node through PageRank algorithm. Table 2 shows a small sample of the 4 calculated measures of topic-user influence. The zero scores mean that user u_i did not have any tweet for that corresponding topic. The non-zero scores are normalized to lie in the range of [0,1] and higher score means higher influence for that topic. The measured influence scores are aggregated and a sample of aggregated scores is shown in Table 3.

Tables 4, 5, and 6 show the top 5 influencers for selected topics. The sample of topics presented in the tables contain machine learning topics such as Neural Networks, Deep Learning, Big Data, Social Networks, Text Mining, NLP, Database, Visualization, and more specific topics such as Hadoop. For the task of validation of the influence results, there is no standard method in the literature to validate the algorithm output. One of the reasons we have chosen the machine learning and data science community on Twitter as our community of study was the wide availability of experts in the domain that allows us to verify the identified influential users through our algorithm. We manually verify the top topic-based influential users through expert opinions, their Twitter, and Google scholar accounts. For example, for the topic NLP, Stanford NLP group appeared in the top 5 influential accounts on Twitter. For "Recommender Systems" topic, Xavier Amatriain, who is known for his works on recommender systems, received a high influence score. Also, for the topic "Neural Networks", Alex Smola was in the top 5 influencers who have extensively published on neural network topic. In the topic "Social Networks", Jure Leskovec, who is well-known in the social networks community, was among the top influencers.

5.4 Implications and Applications

This section describes the real-world implication and applications of our model. Identifying topic-based influential users is similar to the problem of finding experts and authorities. Spotting the elite group of users for topics can improve available systems such as search engines. The query result for both contents and users can be returned and ranked using the score provided by our system.

One of the main applications of this work is in Marketing. Marketing campaigns can be implemented through the influential users in the related topic to have more productive and cost effective campaign. Influential users act as hubs in the network and have a central position in the network in terms of information diffusion, also they attract and engage more users into their conversations.

Our model is able to detect the new and surprising topics. This capability gives the strength to our model that works in real-world and detect new topics and related influential users. As a result, there wouldn't be a need for manually defining the topics

and consequently the recent and new topic would not be missed. TSIM, also can be applied to detect topics at what period get viral and who are influential in those topic in different period of time.

6 Conclusions

In this study, we have presented a Topic-based Social Influence Measurement, TSIM, to measure topic-based user influence in social networks. We have identified topics from user posts on social networks, and measured each user's influence on each topic. TSIM is then used to calculate user influence for the observed topics. Our main contributions include:

- the proposal of a effective method to measure topic-based influence for social network users
- opening a new discussion for user influence prediction in social networks that has not been explored in the literature.

Finally, we have tested TSIM using a unique dataset that we collected from Twitter, which we are making it available online.

In future work, we are interested to measure topic-based user influence over time, and study how influence changes over time. Prediction of user influence on unobserved topics is also currently under our investigation. We will also investigate other methods to combine influence measures.

References

Arasu, A., Novak, J., Tomkins, A. & Tomlin, J. (2002), Pagerank computation and the structure of the web: Experiments and algorithms, *in* 'WWW', pp. 107–117.

Barbieri, N., Bonchi, F. & Manco, G. (2012), Topic-aware social influence propagation models, *in* 'ICDM', pp. 81–90.

Cano, A. E., Mazumdar, S. & Ciravegna, F. (2014), 'Social influence analysis in microblogging platforms–a topic-sensitive based approach', *SWJ* **5**(5), 357–372.

Cataldi, M. & Aufaure, M.-A. (2015), 'The 10 million follower fallacy: audience size does not prove domain-influence on twitter', *KAIS* **44**(3), 559–580.

Eccleston, D. & Griseri, L. (2008), 'How does web 2.0 stretch traditional influencing patterns', *IJMR* **50**(5), 591–161.

Eirinaki, M., Monga, S. P. S. & Sundaram, S. (2012), 'Identification of influential social networkers', *IJWBC* **8**(2), 136–158.

Embar, V. R., Bhattacharya, I., Pandit, V. & Vaculin, R. (2015), Online topic-based social influence analysis for the wimbledon championships, *in* 'KDD', pp. 1759–1768.

Hajian, B. & White, T. (2011), Modelling influence in a social network: Metrics and evaluation, *in* 'PASSAT', pp. 497–500.

Haveliwala, T. H. (2002), Topic-sensitive pagerank, *in* 'WWW', pp. 517–526.

Hu, J., Fang, Y. & Godavarthy, A. (2013), Topical authority propagation on microblogs, *in* 'CIKM', pp. 1901–1904.

Jabeur, L. B., Tamine, L. & Boughanem, M. (2012), Active microbloggers: identifying influencers, leaders and discussers in microblogging networks, *in* 'SPIRE', pp. 111–117.

Jin, X. & Wang, Y. (2013), 'Research on social network structure and public opinions dissemination of micro-blog based on complex network analysis', *JNW* **8**(7), 1543–1550.

Kardara, M., Papadakis, G., Papaoikonomou, A., Tserpes, K. & Varvarigou, T. (2015), 'Large-scale evaluation framework for local influence theories in twitter', *Information Processing Management* **51**(1), 226–252.

Katsimpras, G., Vogiatzis, D. & Paliouras, G. (2015), Determining influential users with supervised random walks, *in* 'WWW', pp. 787–792.

Katz, E. (1957), 'The two-step flow of communication: An up-to-date report on an hypothesis', *Public opinion quarterly* **21**(1), 61–78.

Katz, E. & Lazarsfeld, P. F. (1955), *Personal Influence, The part played by people in the flow of mass communications*, Transaction Publishers.

Kleinberg, J. M. (1999), 'Authoritative sources in a hyperlinked environment', *JACM* **46**(5), 604–632.

Kong, S. & Feng, L. (2011), A tweet-centric approach for topic-specific author ranking in micro-blog, *in* 'ADMA', pp. 138–151.

Kwak, H., Lee, C., Park, H. & Moon, S. (2010), What is twitter, a social network or a news media?, *in* 'WWW', pp. 591–600.

Libai, B., Bolton, R., Bügel, M. S., De Ruyter, K., Götz, O., Risselada, H. & Stephen, A. T. (2010), 'Customer-to-customer interactions: broadening the scope of word of mouth research', *JSR* **13**(3), 267–282.

Liu, X., Shen, H., Ma, F. & Liang, W. (2014), Topical influential user analysis with relationship strength estimation in twitter, *in* 'ICDM', pp. 1012–1019.

McNeill, A. R. & Briggs, P. (2014), Understanding twitter influence in the health domain: A social-psychological contribution, *in* 'WWW', ACM, pp. 673–678.

Montangero, M. & Furini, M. (2015), Trank: ranking twitter users according to specific topics, *in* 'CCNC', pp. 767–772.

Pal, A. & Counts, S. (2011), Identifying topical authorities in microblogs, *in* 'WSDM', pp. 45–54.

Probst, F., Grosswiele, D.-K. L. & Pfleger, D.-K. R. (2013), 'Who will lead and who will follow: Identifying influential users in online social networks', *BISE* **5**(3), 179–193.

Riquelme, F. (2015), 'Measuring user influence on twitter: A survey', *arXiv:1508.07951* .

Romero, D. M., Galuba, W., Asur, S. & Huberman, B. A. (2011), Influence and passivity in social media, *in* 'PKDD', pp. 18–33.

Sung, J., Moon, S. & Lee, J.-G. (2013), *The Influence in Twitter: Are They Really Influenced?*, pp. 95–105.

Welch, M. J., Schonfeld, U., He, D. & Cho, J. (2011), Topical semantics of twitter links, *in* 'WSDM', pp. 327–336.

Weng, J., Lim, E.-P., Jiang, J. & He, Q. (2010), Twitterrank: Finding topic-sensitive influential twitterers, *in* 'WSDM', pp. 261–270.

Xiao, F., Noro, T. & Tokuda, T. (2014), 'Finding news-topic oriented influential twitter users based on topic related hashtag community detection', *JWE* **13**(5-6), 405–429.

Regression classifier for Improved Temporal Record Linkage

Yichen Hu **Qing Wang** **Dinusha Vatsalan** **Peter Christen**

Research School of Computer Science,
The Australian National University,
Canberra ACT 0200, Australia
Email: {yichen.hu,qing.wang,dinusha.vatsalan,peter.christen}@anu.edu.au

Abstract

Temporal record linkage is the process of identifying groups of records which are collected over long periods of time, such as census databases or voter registration databases, that represent the same real-world entities. These datasets often contain temporal information for each record, such as the time when a record was created, or the time when it was modified. Unlike traditional record linkage, which treats differences between records from the same entity as errors or variations, temporal record linkage aims to capture records from entities where the details of these entities change over the time.
This paper proposes a temporal record linkage approach that learns the probabilities for attribute values of records to change within different periods of time, which extends an existing temporal approach *decay model*. The proposed method uses a regression based machine learning model to predict decay with sets of attributes, where attribute values in each set could affect the decay of others. Our experimental results show that the proposed approach results in generally better recall than baseline approaches on real-world datasets.

Keywords: Data matching, entity resolution, record linkage, temporal data

1 Introduction

Record linkage (also known as data matching, entity resolution, and duplicate detection) identifies records that refer to the same real-world entity (Christen 2012*a*). Record linkage is being used in many application domains, such as linking patient data for disease outbreak detection or clinical trails in the health industry, credit checking and fraud detection in the finance industry, and constructing population databases for social science research (Kum et al. 2014). Challenges in record linkage are caused by the lack of unique identifiers (such as national identifier number), dirty data (such as misspellings and missing values), legitimate updates over time (such as changes in last name and address), and the lack of informative attributes (many datasets do not have gender and/or date of birth. e.g. no gender or date of birth in publication datasets).

Record linkage generally involves the following steps: data preprocessing (such as unifying data structure and cleansing datasets), blocking/indexing (grouping records into blocks, where records with a certain similarity are grouped into the same block), comparison and classification (comparing pairs of records in each block to decide if they are a match), and evaluation (Christen 2012*a*). This paper focuses on record pair comparison and classification, but we will also briefly discuss blocking/indexing.

Record linkage has been studied extensively in the past few decades. However, until recently, most works in this field did not use any temporal information available in datasets (Li et al. 2012). Records of the same entity can be collected over a long period of time (multiple years or even decades, such as census data in Australia collected every five years). During such periods the attribute values of an entity are likely to change, such as job position, living address, and potentially last name. Traditional record linkage methods often assume records that are highly similar are most likely to be belonged to the same entity. These techniques do not perform well on temporal records, because many entities have changed some of their attribute values over time. For example, when a person has changed his or her last name and address over a few years, their earlier records can be linked by mistake to records of a different person who has the same last name and/or address.

Temporal record linkage tries to address the above issues by using temporal information (such as the time-stamp when a record was created) from each record as a special type of attribute. These time-stamps can be used to sort records by time order, calculate time distance between records, and therefore provide potential for new record linkage approaches (examples to be discussed in Section 2). To be used in temporal record linkage, a dataset should contain temporal data for each record, such as date entered (for registration dataset), date being published (for publication datasets), and date being collected (for datasets collected by taking snapshots of databases at different times).

This paper extends an existing temporal linkage approach called *decay model* (Li et al. 2012). The *decay model* learns the probability for an attribute to change over time (*disagreement decay*), and the probability for an attribute to share the same value among different entities over time (*agreement decay*), and then uses these decays to compute and adjust the weight given to each attribute. The sum of the adjusted attribute weights is used to calculate the similarity between a pair of records and decide if they are a match or non-match based on a similarity threshold (Christen 2012*a*). We integrate a regression model (such as linear regression) into

the *decay model*. The regression model uses multiple support attributes to calculate the decay of a main attribute whose decay is affected by the values of those support attributes. The calculated decays are more reflective to each entity's specific situation. For example, a person's gender can affect the likelihood of changes in their last name, so when we calculate a decay of last name with gender as additional input, we can produce a gender sensitive decay for last name. We choose linear regression model because it is commonly used in parameter estimation and prediction (Krueger et al. 2015).

2 Related Works

We discuss related works in three areas: generic record linkage techniques, temporal models, and temporal linkage techniques.

Generic record linkage techniques: Generic record linkage techniques refer to the class of algorithms that do not consider temporal information explicitly. Record-wise or cluster-wise similarity comparison functions are sometimes treated as *black-box*. The similarities produced by these function are used to conduct linkage tasks.

Benjelloun et al. (2009) proposed three variations of the Swoosh algorithms (G-Swoosh, R-Swoosh, and F-Swoosh), which handle the record linkage problem by merging each pair of matched records into a new record. Merged records are removed and the newly created record is added to the input dataset. The matching and merging behaviors and criteria are defined by the user. We use F-Swoosh in our approach to link records.

Kim & Lee (2010) proposed an algorithm which uses Locality-sensitive hashing (LSH) (Indyk & Motwani 1998) to iteratively group records to clusters. LSH approximately compares the Jacarrd similarity between records, and place records that are similar to each other to the same cluster (so that any pair of records from that cluster will have a similarity higher than a certain threshold). Records placed into the same cluster are merged into a new record, and the algorithm conducts LSH again in an attempt to merge more records, until an user defined termination condition is met. We use a LSH based blocking technique which we will explain in Section 4.1.

Li et al. (2012) proposed a clustering technique, which computes the similarity between a record and a cluster of records. If a record has higher similarity to another cluster than its current one, this record will be reassigned to the new cluster. In the case where a record's similarity to any cluster is lower than a user-defined threshold, a new cluster is created with this record as its member. The adjustment process is repeated until the result converges or oscillates on the number of clusters.

Temporal models in record linkage: Temporal models are used to adjust attribute-wise or record-wise similarities with temporal information. A temporal model is often learned from a training dataset (which is the case for all of the approaches below). The learned model is then applied in attribute or record pair comparison process, to adjust the finial similarity score.

Li et al. (2012) proposed a similarity measure which considers the probability of an attribute's value to change over a certain time period (the probability is learned from training data). The algorithm calculates a disagreement rate (the probability for an attribute to change within a certain time interval) and an agreement rate (the probability for an entity's attribute value to be the same with a different entity within a time interval). The two rates are used to adjust the weight of each attribute.

More recently, Li et al. (2015) proposed a temporal model which learns the probability for each attribute value to be changed to another attribute value over a certain time period. However, this approach is restricted to attributes whose values might change, such as job positions (for example, the position 'technician' can be changed to 'manager').

Christen & Gayler (2013) adapted the approach by Li et al. (2012), which adjusts the temporal model iteratively using linkage results produced. The difference between this algorithm and the original one is that the original algorithm only learns the temporal model from training data, whereas this algorithm continuously trains the temporal model using linkage results produced by itself.

Chiang et al. (2014*a*) proposed an algorithm which learns the probability of an attribute's value to recur at different time intervals. For each value of an attribute, the algorithm constructs a transition history and uses the history to calculate the probability for a value to recur. These recurrence probabilities are used to adjust similarity weighting of record pairs to improve entity resolution quality.

Temporal linkage techniques: Unlike generic record linkage techniques, temporal linkage techniques use temporal information that is available in a dataset. Unlike temporal models, temporal linkage techniques do not adjust the way in which similarities are calculated between records, but they adjust record comparisons such as the order of comparisons between records, or between clusters of records.

Chiang et al. (2014*b*) proposed a clustering algorithm which processes records in two phases using different temporal models. In the first phase, the algorithm greedily groups records into clusters and creates temporal signatures for each of the clusters. In the second phase, the algorithm calculates the similarities between clusters, and adjusts these similarities using the temporal signatures, to decide if two clusters need to be merged.

Li et al. (2015) recently proposed a temporal clustering algorithm which identifies data-sources that are likely to be well-updated (accurately describe the current state of entities), and then uses these well-updated sources to create the initial clustering before using other data-sources that are not fresh.

3 Notation and Problem Statement

We now provide the notation we use in this paper and define the problem we aim to tackle.

Entity: Given a domain with a set of entities $\mathbf{E}$, where each entity $e \in \mathbf{E}$ is described by a set of attributes $\mathbf{A}$.

Record: Let $\mathbf{R}$ be a set of records, $r \in \mathbf{R}$ refers to a record with a time-stamp t. Each record r has a list of attribute values $[a_1, a_2, ..., a_k]$, where the value of an attribute $A \in \mathbf{A}$ in a record r is denoted as $r.A$. Every record $r \in \mathbf{R}$ must belong to exactly one entity $e \in \mathbf{E}$. The entity that associated with a record r is denoted as $r.e$.

Attribute values of an entity e can change over time, where each change (update) is represented by a record r with a time-stamp t and attribute value(s) that is different from the previous record. For example, let r_1, r_2 be two records belonging to e (in another word, $r_1.e = r_2.e$). If $r_1.t < r_2.t$ and

$\exists A \in \mathbf{A} : r_1.A \neq r_2.A$, then we say that the value of attribute A of entity e was changed between two time-stamps $r_1.t$ and $r_2.t$.

Training dataset: Given a training dataset $\mathbf{C}$ in the form of a set of clusters of records. Each cluster $C \in \mathbf{C}$ contains a set of records $\{r_1, r_2, ...\}$ that represents an entity e from the domain. All records in a cluster C represent the same entity, and all records referring to the same entity are in the same cluster C.

Problem statement: Temporal record linkage is the problem of grouping a set of records $\mathbf{R}$ into a set of clusters $\mathbf{C}'$. Ideally, for each created cluster $C' = \{r_1, r_2, ...\}$, and $C' \in \mathbf{C}'$, C' represents an entity $e_j \in \mathbf{E}$. All records in a cluster $C' \in \mathbf{C}'$ belong to the same entity: $\forall r \in C' \rightarrow r.e = e_j, e_j \in \mathbf{E}$. All records that are belonging to the same entity are in the same cluster: $\forall r_1 \in \mathbf{R}$, $\forall r_2 \in \mathbf{R}$, $r_1.e = r_2.e \leftrightarrow \exists C' \in \mathbf{C}'(r_1 \in C' \wedge r_2 \in C')$.

Our work addresses the temporal record linkage problem using a weighting strategy which adjusts the weight of each attribute. The objective of our work is to improve the quality of linkage. Such a weighting strategy is also called a temporal model, which is trained by a training dataset $\mathbf{C}$.

Temporal model training: Given a training dataset $\mathbf{C}$ with ground truth, the problem of training a temporal model is to build a statistical model for attribute weighting. This model can adjust the weight of each attribute A when a pair of records are being compared. The adjusted weights reflect the temporal characteristics of the whole dataset.

Our work addresses the temporal model training problem with a machine learning approach, by using a regression model to train and predict parameters for temporal model. Given an attribute A from a record r, and a time distance Δt, there exists a probability $p^{\neq}$ that A changes its current value within time distance Δt. Records used to train the model can thus be created with two class values: *changed* or *unchanged* using training data $\mathbf{C}$.

Training data generation: Given a cluster $C \in \mathbf{C}$ where $C = \{r_1, r_2, ...\}$. Given a record r_i in C and a time distance Δt, we can find all records $r' \in C - \{r_i\}$ where $r'.t - r_i.t \leq \Delta t$ holds. If there exists a record r' such that $r'.A \neq r_i.A$, a record for training can be created with class value *changed*, with Δt and attribute values of r_i as features. Similarly, if every record r' satisfies the condition: $r'.A = r_i.A$, a record for training with class value *unchanged* is created.

4 Framework

Figure 1 presents a high-level overview of our record linkage process. With a set of records $\mathbf{R}$ as input, a blocking method first places records into smaller sets, and only compares records within each set (there can be overlaps between these sets, i.e. each record can be inserted into more than one set). The reason for using a blocking method is to cluster records that are similar to each other into smaller sets, therefore improve the scalability and reduce computational cost for linkage approaches that have a running time growing exponentially by the size of dataset. The similarity threshold for threshold based blocking methods is usually defined by the user (Christen 2012*b*). The linkage technique computes the similarities between the records within those blocks (Christen 2012*a*). The criteria for two records (or clusters of records) to link are defined by specific linkage technique, such as by

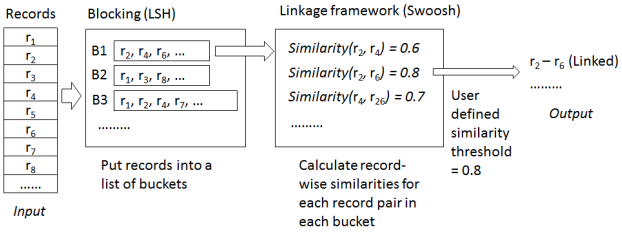


Figure 1: Overview of our linkage approach. Given a set of records as input, our blocking approach places the records in blocks (each record can be placed in multiple blocks). For each block, a linkage framework calculates similarities of record pairs and links records according to certain criteria, such as a user defined similarity threshold.

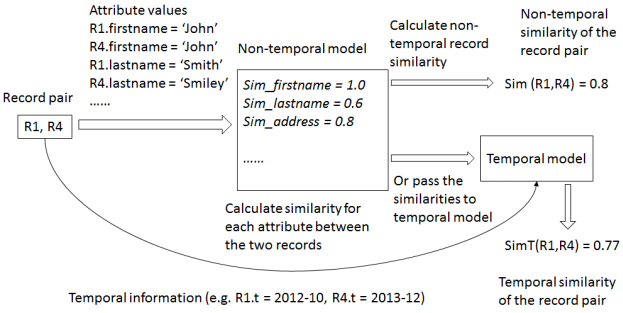


Figure 2: Temporal and non-temporal similarity values of a given pair of records. The difference is that a temporal model combines temporal information with attribute similarities to calculate the record-wise similarity.

a user-defined similarity threshold. In this paper, we use LSH (Locality Sensitive Hashing) (Indyk & Motwani 1998) for blocking, and F-Swoosh (Benjelloun et al. 2009) as linkage technique. Both will be further explained in detail in the following sections.

Figure 2 shows how a pair-wise record similarity is produced with and without a temporal model, as well as the problem domain this paper will address. The temporal model takes attribute-wise similarities of a record pair as input, as well as temporal information related to the record pair (such as the dates when the records were added to the dataset). Same as the a non-temporal model, the final output of a temporal model is a similarity score for a pair of records, where these similarity scores are adjusted by the temporal information.

4.1 Blocking: Locality-Sensitive Hashing

Blocking methods create smaller record sets based on the original dataset, where records in each block possibly refer to similar entities. Only those records within the same blocks are compared with each other. Conceptually, a blocking method can be understood as a linkage method with a low threshold which is only concerned with recall and not precision.

We use LSH to create blocks. LSH is an approximation algorithm used to cluster similar texts. The texts in the same cluster will have a high Jaccard similarity (computed by shingling a text into a set of q-grams) above a user defined threshold (Rajaraman & Ullman 2011). We use LSH as the blocking method based on the observation that a single attribute value

Algorithm 1 Blocking with LSH

Input:
- A set of records: $\mathbf{R}$
- A list of lists that contains attributes for blocking: $\mathbf{L}_A$
- A similarity threshold t_s
- Maximum block size $size_{max}$

Output:
- Sets of record ids, each set is a block: $\mathbf{B}$

```
1: B = set()
2: //For each list from L_A, produce a set of blocks
3: for l_A in L_A do
4:    //Create a hashtable where each
5:    //record-reference links to a list of block keys
6:    D_I = hashtable()
7:    //Generate block-keys for each record
8:    for r in R do
9:       gramset = set()
10:      for A in l_A do
11:         //Get the q-grams of the attribute value
12:         //q is defined by the user for each attribute
13:         for gram in GetQGrams(r.A, q) do
14:            gramset.add(gram)
15:      blockkeys = LSHBucketKeys(gramset, t_s)
16:      D_I[r.recordid] = blockkeys
17:   D_b = hashtable()
18:   //Group record-references by block-keys
19:   for recordid, blockkeys in D_I do
20:      for key in blockkeys do
21:         if key not in D_b then
22:            D_b[key] = set()
23:         D_b[key].add(recordid)
24:   //Remove over-sized blocks
25:   for block in D_b.values() do
26:      if length(block) >= size_max then
27:         D_b.remove(block)
28:   B = B ∪ D_b.values()
29: return B
```

(such as first name), or the concatenation of a list of attribute values (such as the string concatenation of first name, last name, and zip code), can be considered as a text string. If we consider a string value as a blocking key value of a record, we can put similar records into the same block by comparing their blocking key values using LSH. LSH is chosen as our blocking approach because it is scalable and efficient, performing reasonably good when comparing it to other blocking approaches (Wang et al. 2016).

After blocks of records are produced by LSH, we remove the over-sized blocks by a user defined maximum block size ($size_{max}$), to further reduce the computational cost. $size_{max}$ was introduced as we observed that some blocks can be very large due to commonly shared attribute values (such as first name 'David'). As a result, these large blocks significantly increased the overall run-time of the algorithm. A similar approach was used in suffix array based blocking (de Vries et al. 2011).

Algorithm 1 describes the way we use LSH for blocking in our work. From lines 8 to 16, a list of blocking keys is created for each record and hashed into the hashtable $\mathbf{D}_I$ by its record ID. The function $GetQGrams()$ takes a text input value and an integer q (Ukkonen 1992). The value of q decides the length of q-grams which will be created by shingling the text input into q-grams. The function $LSHBucketKeys()$ takes a set of q-grams and a similarity threshold to produce LSH buckets. We treat the $LSHBucketKeys()$ function as a blackbox in this work as its internal mechanism is dependent on the implementation of LSH. Each bucket key uniquely identifies a block. For example, if the hashing of a record results in ten keys, this record is hashed into ten blocks by LSH. From lines 19 to 23, $\mathbf{D}_I$ is converted to another hashtable $\mathbf{D}_b$ where each blocking key is mapped to a list of record IDs (each list is a block). In lines 25 to 27, the algorithm applies the $size_{max}$ parameter and removes over-sized blocks.

Algorithm 2 The Swoosh algorithm (conceptual)

Input:
- A set of records: $\mathbf{R}$

Output:
- A set of records, each record represents an entity: $\mathbf{E}$

```
1: E = ∅
2: while R ≠ ∅ do
3:    currentRecord = R.getFirstItem()
4:    R.removeFirstItem()
5:    buddy = null
6:    for r' in E do
7:       if IsMatched(currentRecord,r') then
8:          buddy = r'
9:          break
10:   if buddy == null then
11:      E.append(currentRecord)
12:   else
13:      merged = Merge(currentRecord, buddy)
14:      E.remove(buddy)
15:      R.append(merged)
16: return E
```

4.2 Linkage: F-Swoosh

Swoosh is an entity resolution approach which compares records according to features (a feature is a set of attributes) selected by the user (Benjelloun et al. 2009). A pair of records are merged into a new record when one of their features meets the matching criteria provided by the user. The two original records are removed after a new record is created by merging. Algorithm 2 is a basic description of the algorithm. As a version of Swoosh algorithm, F-Swoosh is optimised with various hashtables and feature operations. In practice we use F-Swoosh for matching, but here we use the algorithm of R-Swoosh to present this approach as R-Swoosh is simpler and more straightforward. The function $IsMatched(r, r')$ in line 7 is a function provided by the user which compares two records and decides if they are a match. The $Merge(r, r')$ function in line 13 is provided by the user as well, which decides how to merge each attribute of two records r and r' (possible ways to handle attribute merging include keeping the latest value, or creating a list of all values (Benjelloun et al. 2009)).

4.2.1 Tracking Merged Records

While F-Swoosh merges records that likely belong to the same entity, we also keep track of the time (which can be a year, such as 2012, or a date, such as 20-10-2011) when each attribute value was originally generated. Each attribute value is stored in the form of a tuple: (*attribute_value*, *updated_date*), and each attribute can have one-to-many attribute values for a given record after multiple merges.

Example: Let record r_1 = ['Tom', 'Bruce', '21 May Street', '2012-12'] (in the format of [*first name*, *last name*, *address*, *time-stamp*]), record r_2 = ['Tom', 'Steven', '21 May Street', '2013-06']. The new record created by merging r_1 and r_2 will be: [[('Tom','2012-12'), ('Tom','2013-06')], [('Bruce','2012-12'), ('Steven','2013-06')], [('21 May Street','2012-12'), ('21 May Street','2013-06')]].

4.3 Temporal Models

Temporal models refer to the models that adjust record pair similarity using temporal information (such as time related rules and patterns) learned from a dataset. In this paper, all temporal models used are built from a training dataset.

The *decay model* calculates disagreement decay and agreement decay of each attribute, and uses them to calculate the similarity of a record pair (Li et al. 2012). Disagreement decay and agreement decay both describe attribute characteristics learned from training data, indicating the probability for an attribute to change within a time distance, and the probability for an attribute to be shared by multiple entities within a time distance, respectively. Time distance refers to the difference between two time-stamps. Time distance is measured by a time unit which is defined by the user, such as days, years, or hours.

A *life span* l refers to the time distance between the time-stamps of two values. An attribute value's life span is *full* when the value has a date when it is start to be used, and another date when it is changed to another value. The time distance between the first date and the second date is a *full life span*, denoted as l_f. Similarly, if the attribute's value does not change between two time-stamps, the time distance between the two time-stamps is a *partial life span*, denoted l_p.

For example, assume an entity with three different last names over five records, with time-stamps in the form of [year-month]: 'Taylor' (2011-10) → 'Taylor' (2011-12) → 'Spire' (2012-12) → 'Spire' (2013-10) → 'Wright' (2015-10). The time distance between the first and the third record is one full life span with a length of 14 months (2011-10 to 2012-12), and the distance between the third and the fifth record is another full life span with a length of 34 months (2012-12 to 2015-10). Note in this example, month is being used as a time unit but it is not necessary for all datasets. From the example above, the time distance between the first and the second record is a partial life span with a length of 2 months (the time distance between 2011-10 and 2011-12), and the time distance between the third record and the forth record is another partial life span with a length of 10 months.

$\bar{L}_f$ denotes the list of all full life spans of an attribute A for all entities. Similarly, $\bar{L}_p$ denotes the list of all partial life spans of an attribute A for all entities.

Definition 4.1. (Disagreement Decay $d^{\neq}$): Let Δt be a time distance, $A \in \mathbf{A}$ be a single valued attribute. The disagreement decay of A over time Δt, is the probability that an entity changes its value of A within a time distance Δt (Li et al. 2012).

$$d^{\neq}(A, \Delta t) = \frac{|\{l \in \bar{L}_f | l \leq \Delta t\}|}{|\bar{L}_f| + |\{l \in \bar{L}_p | l \geq \Delta t\}|} \tag{1}$$

Equation 1 calculates the disagreement decay $d^{\neq}$ given an attribute A and a time distance Δt. $\bar{L}_f$ is the set of full life spans of attributes values of attribute A. The time unit of Δt, is defined by the user, which can be days, months, or years.

Definition 4.2. (Agreement Decay $d^{=}$): Let Δt be a time distance, $A \in \mathbf{A}$ be a single valued attribute. The agreement decay of A over a time distance Δt, is the probability that two different entities share the same value for A within Δt. (Li et al. 2012)

$$d^{=}(A, \Delta t) = \frac{|\{l \in \bar{L} | l \leq \Delta t\}|}{|\bar{L}|} \tag{2}$$

Equation 2 calculates the agreement decay $d^{=}$. $\bar{L}$ is a list of life spans. For each record from a training dataset, if it has the same attribute value with another record which belongs to a different entity, the time distance between the two records is added to $\bar{L}$. If no entity has the same attribute value, a life span with length ∞ is added to $\bar{L}$.

We use agreement decay and disagreement decay to calculate w_A (weight per attribute), as shown in Equation 3. Then, weights are used to calculate the pair-wise similarity between two records. As shown in Equation 4, $s_d(r, r')$ denotes the decay adjusted similarity between two records r and r'. s_a refers to the similarity between a pair of attribute values.

$$w_A(s_a, \Delta t) = 1 - s_a \cdot d^{=}(A, \Delta t) - (1 - s_a) \cdot d^{\neq}(A, \Delta t) \tag{3}$$

$$s_d(r, r') = \frac{\sum_{A \in \mathbf{A}} w_A(s_a(r.A, r'.A), |r.t - r'.t|) \cdot s_a(r.A, r'.A)}{\sum_{A \in \mathbf{A}} w_A(s_a(r.A, r'.A), |r.t - r'.t|)} \tag{4}$$

5 Our Approach

This section discusses our temporal model and training strategy in detail.

5.1 Disagreement Probability

Disagreement probability is a concept introduced in this work. It has a similar definition as disagreement decay (as shown in Equation 1), but is modified to make it easier to be used with a machine learning model. From Equation 5, we can see that the only difference between $d^{\neq}_{prob}$ and $d^{\neq}$ is that the divisor no longer decreases with an increasing Δt. We use $d^{\neq}_{prob}$ instead of $d^{\neq}$ because the equation of $d^{\neq}_{prob}$ has a divisor that is fixed for each entity. This problem is therefore intuitively easier to fit into a machine learning classifier that when a life span l is encountered, we can immediately decide if it is lower than a Δt and create a training instance with the attribute values associated to the life span l.

$$d^{\neq}_{prob}(A, \Delta t) = \frac{|\{l \in \bar{L}_f | l \leq \Delta t\}|}{|\bar{L}_f| + |\bar{L}_p|} \tag{5}$$

$d^{\neq}_{prob}$ is normalized into the range $[0, 1]$, and then used as weights to adjust respective attribute-wise similarities, as shown in Equation 6. s_p denotes the adjusted similarity between a pair of records, and the function $s_a(a, a')$ returns the similarity between a pair of attribute values. The specific value comparison function for an attribute is defined by the user, which returns a similarity measure in the range [0,1]. These comparison functions can be approximate string similarity functions, such as edit-distance, Jaro-Winkler, etc. (Christen 2012*a*).

$$s_p(r, r') = \sum_{A \in \mathbf{A}} \frac{1 - d^{\neq}_{prob}(A, |r.t - r'.t|)}{\sum_{A' \in \mathbf{A}} 1 - d^{\neq}_{prob}(A', |r.t - r'.t|)} \cdot s_a(r.a, r'.a) \tag{6}$$

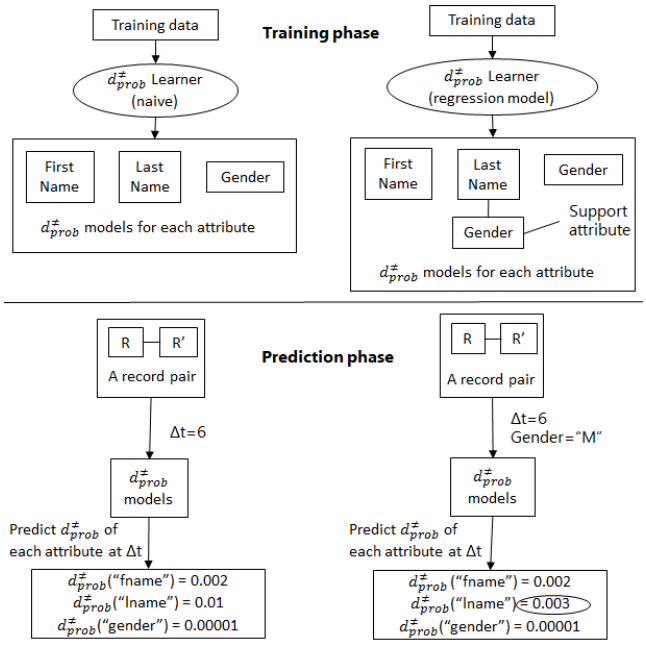


Figure 3: Disagreement probability training and prediction with decay model (left) and regression model (right). We can see the model on the right takes a value from one support attributes as input, *gender* in this case, and calculates a disagreement probability value for a specific subset of entities (males in this example). Whereas the model on the left side calculates a disagreement probability value regardless to the entity's gender. The number in the black circle shows that a different disagreement probability value is calculated for attribute *last name*.

5.2 A Regression Model

From the previous equations we can see that agreement decay, disagreement decay, and disagreement probability are calculated with only one attribute independently each time. For example, when the disagreement decay of attribute *last name* is calculated, the temporal model calculates the overall probability for an entity to change its last name within a given interval Δt. However, the probability of an entity to change its last name is often associated with gender and age. The disagreement decay for last name, calculated without considering its gender and age group, will for example be too high for older male entities, and too low for younger female entities, because it is rare for an older male person to change last name, but common for a young female to change last name due to marriage.

Support attributes are assigned to certain attributes when the value of a support attribute may affect the probability of the attributes to change. For example, when building a model which predicts the probability for attribute *address* to change, attribute(s) such as *gender* and *age* can be used to make the prediction more accurate. Support attributes are selected by the user based on empirical knowledge, however they can also be selected by a feature selection strategy (Blum & Langley 1997).

Figure 3 compares the difference between the original learning strategy (left) and the learning strategy using a machine learning model (right), with attribute *gender* used as a support attribute for attribute *last name*.

Algorithm 3 shows how we use a machine learning model to predict disagreement probability. In line 6, the function $getAttValues(r, L_A)$ extracts a list of attribute values from a record r according to an attribute list L_A. The attribute values are important features later used to create training instances. In lines 9-12, a training instance with class value 0, with Δt and $supportValues$ as features is created. Since partial life spans indicate no value change, we use 0 to denote *non-change*. In lines 16-17, in the case that a change happen within a Δt, we create a training instance with class value 1 which denotes *change*. The training instances are then sent to the machine learning model defined by the user which can be used to predict $d^{\neq}_{prob}$ for an attribute A.

Algorithm 3 Temporal model training with disagreement probability

Input:
- A list of clusters: **C**, each cluster $C \in \mathbf{C}$ contains a list of records that belong to an entity
- An attribute to build temporal model: A_1
- A list of support attributes: L_A
- A machine learning algorithm: *model*

Output:
- A trained temporal model

```
1:  train = ∅
2:  for C in C do
3:      start = 1
4:      while start ≤ |C| do
5:          end = start +1
6:          supportValues = getAttValues(C[start], L_A)
7:          while C[start].a_1 == C[end].a_1 and end ≤ |C| do
8:              end = end + 1
9:          if end > |C| then
10:             Δt = C[|C|].t − C[start].t + 1 // Partial life span
11:             for i = 1 to Δt_max do
12:                 train.add([0, Δt, supportValues])
13:         else
14:             Δt = C[end].t − C[start].t // Full life span
15:             for i = 1 to max_Δt do
16:                 if Δt ≥ i then
17:                     train.add([1, Δt, supportValues])
18:                 else
19:                     train.add([0, Δt, supportValues])
20:         start = end
21: // Train the model with the accumulated training data
22: model.fit(train)
23: return model
```

Figure 4 shows the different decay values calculated for attribute *last name* using disagreement decay $d^{\neq}$ (see Equation 1) and disagreement probability $d^{\neq}_{prob}$ predicted by a regression model which was trained using Algorithm 3. Attribute *gender* was used as the support attribute. We can see that with our approach, lower disagreement probabilities are calculated for entities with male gender, whereas higher disagreement probabilities are calculated for entities with female gender, for attribute *last name*. This makes sense as female entities change last name more often than male entities due to marriages.

5.3 Combine Disagreement Probability with Agreement Decay

Disagreement probability can be modified into the same form as disagreement decay by normalising it: $d^{\neq}_{nprob}(A, \Delta t) = \frac{d^{\neq}_{prob}(A,\Delta t)}{max(d^{\neq}_{prob}(A))}$. where $max(d^{\neq}_{prob}(A))$ is the maximum disagreement probability over all Δt. Using Equation 3 and Equation 4 above, the temporal similarity for our model can be calculated by substituting $d^{\neq}$ with $d^{\neq}_{nprob}$.

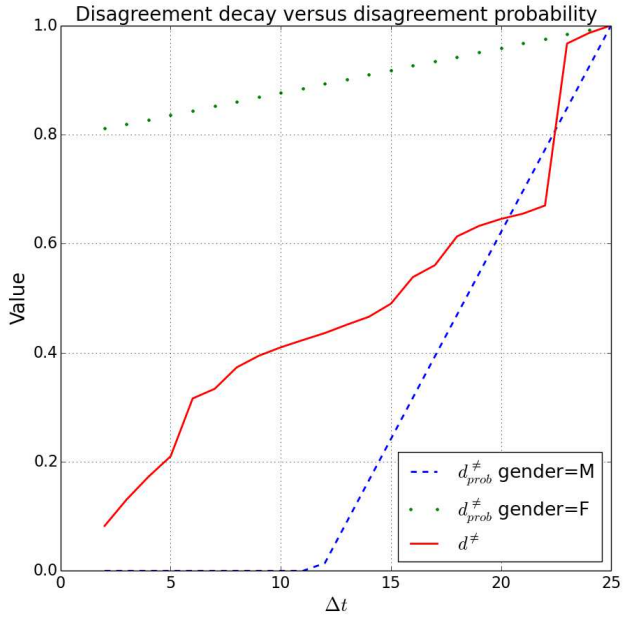


Figure 4: Different decay values for attribute *last name* calculated using disagreement decay $d^{\neq}$ and disagreement probability $d^{\neq}_{prob}$ predicted by a regression model. Attribute *gender* was used as the support attribute.

Table 1: A summary of snapshots of the NCVR database and temporal records created from each snapshot

Snapshot (year-month)	Number of records	Records added	Records updated	Temporal records
2011-10	6,233,683.00	5,660,833	0	5,660,833
2011-12	6,981,774.00	285,091	50,181	335,272
2012-02	6,974,887.00	40,370	40,187	80,557
2012-04	7,054,734.00	82,370	82,496	164,866
2012-06	7,090,377.00	38,792	87,326	126,118
2012-08	7,134,332.00	47,928	86,418	134,346
2012-10	7,310,212.00	183,858	182,609	366,467
2012-12	7,524,471.00	221,922	354,053	575,975
2013-02	7,251,818.00	34,607	83,552	118,159
2013-04	7,268,064.00	28,878	43,423	72,301
2013-06	7,291,726.00	27,293	26,511	53,804
2013-08	7,325,036.00	35,382	32,528	67,910
2013-10	7,358,266.00	35,683	54,264	89,947
2013-12	7,388,104.00	33,495	50,440	83,935
2014-02	7,391,221.00	28,336	32,974	61,310
2014-06	7,453,901.00	69,002	120,422	189,424
2014-08	7,490,428.00	38,250	50,923	89,173
2014-10	7,539,857.00	53,438	86,702	140,140
2014-12	7,608,324.00	72,576	211,607	284,183
2015-02	7,111,324.00	29,247	43,555	72,802
2015-04	7,129,997.00	24,578	50,048	74,626
2015-06	7,156,299.00	31,320	88,298	119,618
2015-08	7,198,755.00	45,845	58,748	104,593
2015-10	7,244,629.00	50,579	71,549	122,128
2015-12	7,272,428.00	46,563	74,142	120,705
2016-02	7,313,555.00	45,498	77,645	123,143

6 Experiments

In this section we describe datasets, methods and measures used in our experiment. Then we present and discuss the experimental results.

6.1 Experimental Settings

we describe the characteristics of the datasets we used in the experiments and the methods, measures, and other implementation details of the experiments.

Table 3: Testing datasets

Dataset name	Number of entities	Number of records	Entities with more than one records
Avery	13,707	15,464	1,604
Buncombe	221,106	282,947	49,490
Cherokee	25,450	28,715	2,875
Gates	9,396	10,504	1,006
Guilford	410,661	515,878	85,192
Montgomry	18,686	21,544	2,539
Avery_L2	1,604	3,361	1,604
Buncombe_L2	49,490	111,331	49,490
Cherokee_L2	2,875	6,140	2,875
Gates_L2	1,006	2,114	1,006
Guilford_L2	85,192	190,409	85,192
Montgomry_L2	2,539	5,397	2,539

Temporal datasets: The temporal datasets we used in this paper are from the North Carolina Voter Registration (NCVR) datasets collected every two months[1]. The datasets have ground truth (entity identifiers) available for all entities. Because the raw datasets are collected in the form of snapshots of databases, we preprocessed them to refine their temporal aspects. Only records that describe changes of an entity are selected into the temporal dataset.

If an entity never has changes in its attribute values, only the earliest record will be selected for that entity. If we sort records of an entity by increasing time-stamp values, the first record of an entity is an *add*, since it indicates the time that the entity was created in that dataset. Any following record of that entity, where a record has any attribute values (except age and time-stamp) that are different from the last record, then this record is considered as *update*.

Table 1 summarizes the number of records in each database snapshot and the number of temporal records created from them. A total of 8,336,205 unique entities were added to the datasets at different point of time.

Example: As shown in Table 2, an entity $e1$ with name "William Crawford Taylor" was added to the snapshot at 2011-10, but only two updates occurred over the next four snapshots: middle name was changed from 'Rose' to 'Louise' at 2012-02, and last name was changed from 'Taylor' to 'Clark' at 2012-06.

The temporal dataset of entity $e1$ will only have the initial record and the two records when the updates occurred, as shown in the left side of the table. The second entity, $e2$, used another first name at 2011-12 and 2012-02, then was changed back. This entity will have three temporal records, one for the first record of the entity at 2011-10, two for the two updates. For the third entity $e3$ whose name "Michelle Mary Lee" has never been changed across the five snapshots, this entity will only have one record in the temporal dataset as there was no update happened.

Above 80% of entities are in the same situation as the third entity $e3$ in the NCVR datasets, indicating updating election information is not very common for most people.

Only a subset of datasets and attributes were used in our work. These attributes were used in the test: last name, middle name, first name, name suffix, residential address, age, and sex code. Sex code has been used as a support attribute for last name, middle name, and residential address. We choose these attributes as they are commonly seen in different datasets, comparing to other attributes, such as race code, party code, and phone number. We did not use

[1] http://dl.ncsbe.gov/

Table 2: Sample records in a temporal dataset

	Raw records				Temporal records				
Entity	**Date**	**First name**	**Last name**	**Middle name**	**Date**	**First name**	**Last name**	**Middle name**	**Action**
e1	2011-10	William	Taylor	Rose	2011-10	William	Taylor	Rose	Add
e1	2011-12	William	Taylor	Rose					
e1	2012-02	William	Taylor	Louise	2012-02	William	Taylor	Louise	Update
e1	2012-04	William	Taylor	Louise					
e1	2012-06	William	Clark	Louise	2012-06	William	Clark	Louise	Update
.....									
e2	2011-10	David	Edward	Jr	2011-10	David	Edward	Jr	Add
e2	2011-12	Dave	Edward	Jr	2011-12	Dave	Edward	Jr	Update
e2	2012-02	Dave	Edward	Jr					
e2	2012-04	David	Edward	Jr	2012-04	David	Edward	Jr	Update
e2	2016-06	David	Edward	Jr					
.....									
e3	2011-12	Michelle	Lee	Mary	2011-12	Michelle	Lee	Mary	Add
e3	2012-02	Michelle	Lee	Mary					
e3	2012-04	Michelle	Lee	Mary					
e3	2012-06	Michelle	Lee	Mary					
.....									

the full datasets for testing at this stage as the proposed approaches are still being tuned and testing on the full datasets is time consuming. The results from selected subsets are well informative so far. We aim to test on the full datasets in the future.

Training dataset. The NCVR temporal dataset of county 'Alexander' was used as a training dataset. Which has 33,995 records from 27,725 entities, and 5,403 entities have at least two records.

Testing datasets: Temporal datasets of six counties were selected from NCVR as testing datasets, where each county has two versions of temporal datasets: the original temporal dataset (named by the county's name) and a refined temporal dataset (L2 dataset) where every entity has at least two records, as shown in Table 3. The original temporal datasets of NCVR have a low percentage of entities who have at least two records, which means the majority of records are not linkable. L2 versions of datasets were created by extracting records from entities with at least two records from its respective original temporal dataset, to test the algorithm's performance when all of the records are linkable.

The reason to create one L2 dataset for each county is to test the algorithm's performance in a distinct data environment where most entities have multiple records.

6.1.1 Implementation

We implemented all algorithms in Python 2.7, and the experiments were conducted on a server with 64-bit Intel Xeon (2.4 GHz) CPUs, 128 GBytes of memory and running Ubuntu 14.04.

We implemented four algorithms which are being discussed below. All of the algorithms above were implemented on the R-Swoosh clustering framework (Benjelloun et al. 2009). Blocks were generated using LSH as discussed in Section 4.1, with pairs completeness greater than 99%, which means greater than 99% of records can be correctly linked with an ideal linkage technique. The same set of blocks was used by the four algorithms. For the regression model, we used linear regression model from sklearn python package with default settings. [2]

For string attributes, the similarity of a pair of attribute values was calculated using the Jaro-Winkler string comparison function (Christen 2012*a*). The similarity of a pair of age values was calculated as: $s_{age} = \frac{1}{|age_1 - age_2|+1}$. The similarity threshold used by all algorithms was 0.8, which means record pairs with similarity equal to or above this threshold are matches (same entity) and below are non-matches. This threshold is arbitrarily chosen. Future experiments can be done with different similarity thresholds.

- No model. A baseline approach with no temporal model. Weights of attributes were not adjusted by a temporal model.
- Decay model (*Decay*). A baseline approach using the temporal model proposed by Li et al. (2012). The algorithm calculates a disagreement rate (the probability for an attribute to change within a certain time interval) and agreement rate (the probability for an entity's attribute value to be the same as other different entities within a time interval), and uses the two rates to adjust the weight of each attribute.
- Disagreement probability regression model (*Disprob*). A temporal model uses a regression model to predict disagreement probability, and reduces the weight of attributes when its predicted disagreement probability is high. The weights of attributes are normalized so that the sum of attribute weights is always 1.
- Disagreement probability plus agreement decay regression model (*Mixed*). With a disagreement probability being predicted in the same way as the method above, the mixed method also calculates agreement decay from the decay model. Disagreement probability and agreement decay are combined to adjust the weight of each attribute.

Measures. Let *res* be a linkage result in the form of clusters of records that are matching, *stand* be the ground truth that *res* corresponds to, which is also in the form of clusters of records. Pair-wise precision $(Precision) = \frac{|res \cap stand|}{|res|}$, pair-wise recall $(Recall) = \frac{|res \cap stand|}{|stand|}$, and $F_1 = \frac{2 * Precision * Recall}{Precision + Recall}$.

6.2 Experimental Results

We compared the four approaches (two baseline and two proposed approaches) on the 12 testing datasets, using precision, recall, and F1. The 'Alexander' dataset was used as the training dataset.

[2] http://scikit-learn.org

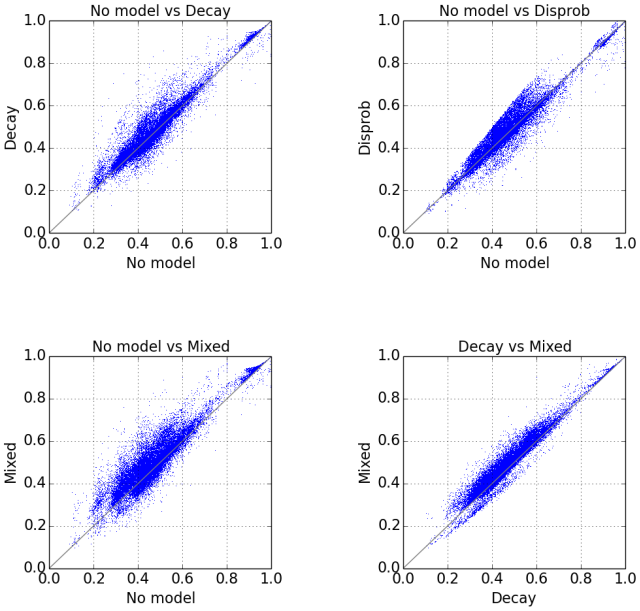
Table 4: Linkage results on original temporal datasets

Dataset	Avery			Buncombe			Cherokee		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
No model	**0.8830**	0.9350	**0.9083**	**0.7495**	0.9517	**0.8386**	**0.8607**	0.9198	**0.8893**
Decay	0.6857	0.9579	0.7992	0.5740	0.9604	0.7185	0.6177	0.9394	0.7453
Disprob	0.8449	0.9126	0.8774	0.7407	0.9315	0.8252	0.8000	0.8896	0.8425
Mixed	0.6421	**0.9802**	0.7759	0.5158	**0.9754**	0.6747	0.5687	**0.9658**	0.7159
Dataset	Gates			Guilford			Montgomry		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
No model	**0.8840**	0.9326	**0.9076**	**0.7100**	0.9421	**0.8097**	**0.8904**	0.9272	**0.9085**
Decay	0.7101	0.9581	0.8157	0.5741	0.9531	0.7166	0.7269	0.9450	0.8217
Disprob	0.8346	0.8998	0.8660	0.7072	0.9219	0.8004	0.8492	0.9002	0.8740
Mixed	0.6654	**0.9737**	0.7905	0.5037	**0.9707**	0.6632	0.6682	**0.9698**	0.7912

Table 5: Linkage results on L2 temporal datasets

Dataset	Avery_L2			Buncombe_L2			Cherokee_L2		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
No model	0.9841	0.9339	0.9584	0.8922	0.9517	0.9210	**0.9907**	0.9195	0.9538
Decay	0.9766	0.9568	0.9666	0.8491	0.9604	0.9013	0.9730	0.9421	0.9573
Disprob	**0.9882**	0.9126	0.9489	**0.9297**	0.9315	**0.9306**	0.9863	0.8894	0.9353
Mixed	0.9597	**0.9787**	**0.9691**	0.7875	**0.9757**	0.8716	0.9564	**0.9680**	**0.9621**
Dataset	Gates_L2			Guilford_L2			Montgomry_L2		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
No model	**0.9887**	0.9334	0.9603	0.8538	0.9428	0.8961	**0.9832**	0.9269	0.9542
Decay	0.9701	0.9589	0.9645	0.8319	0.9540	0.8888	0.9657	0.9456	0.9555
Disprob	0.9856	0.8998	0.9407	**0.9074**	0.9225	**0.9149**	0.9814	0.9005	0.9392
Mixed	0.9619	**0.9737**	**0.9677**	0.7597	**0.9716**	0.8527	0.9457	**0.9701**	**0.9578**

Figure 5: Similarity scatter plot between the four approaches. Each plot shows the similarities produced by the two approaches for each of the record pairs. We can see that *decay* and *mixed* tend to produce a higher similarity than *no model*, and *mixed* tends to produce a higher similarity than *decay*.



Results based on original datasets: Table 4 shows the results on the six original temporal datasets (shown in Table 3). We observed that the performance of a linkage approach drops significantly when the size of a dataset increases. This is expected, as the larger a dataset is, the more likely for a non-matched pair with high similarity exists.

Results based on L2 datasets: Table 5 shows the results on the six L2 datasets (where only entities with at least two records are included). L2 datasets performed significantly better than original datasets since there is at least one match guaranteed for each record and they are also smaller.

We observed that the approach without a temporal model (*no model*) produced the best F1 scores, which is consistent with the observation from Li et al. (2012) where better performance was achieved only after using a temporal clustering technique. The *mixed* approach produced the highest Recall throughout the experiment, however at a significant cost with Precision. The *disprob* approach produced the best result among temporal models, however, generally inferior to the *no model* baseline.

Comparison between temporal and non-temporal approaches. We extracted a subset of the results in attempt to analyze the impact of the temporal models. As Figure 5 shows, we found the approach with a temporal model tends to produce higher similarity on record pairs than the approach without a temporal model. On a closer look we found that temporal models gave attribute *first name* a higher weight and attribute *address* and *last name* a lower weight, which is expected since it is more common for people to change address and last name. However, when a pair of non-match records share a popular first name such as 'Anna', the temporal model made them more likely to be matched by mistake, especially when their age and gender are the same too. A potential fix to this issue is to introduce a frequency based weighting strategy, such as weighting attribute according to a value's frequency in the context (TF-IDF) (Witten et al. 1999), or use the temporal clustering method as proposed by Li et al. (2012).

7 Conclusion and Future Works

In this paper we developed two attribute weighting approaches using a linear regression model to improve the quality of temporal record linkage. Our regression model uses multiple attribute values from a record pair as input, to predict the probability of an attribute value to change within a certain time period, and then adjust the weight of the attribute accordingly. We evaluated our approaches on twelve datasets derived from NCVR datasets. Experimental results show that one of our approaches performed better than the temporal baseline, and another approach achieved overall highest recall.

In the future, we intend to incorporate a frequency based weighting strategy into the framework, and to see if the undesired high similarities can be adjusted properly. Another possible direction is to test the temporal models with different clustering techniques, such as the temporal clustering techniques proposed by Li et al. (2012) and Chiang et al. (2014*b*).

Acknowledgements

This work was partially funded by the Australian Research Council (ARC) under Discovery Project DP160101934.

References

Benjelloun, O., Garcia-Molina, H., Menestrina, D., Su, Q., Whang, S. E. & Widom, J. (2009), 'Swoosh: a generic approach to entity resolution', *The VLDB Journal* **18**(1), 255–276.

Blum, A. L. & Langley, P. (1997), 'Selection of relevant features and examples in machine learning', *Artificial Intelligence* **97**(1-2), 245–271.

Chiang, Y.-H., Doan, A. & Naughton, J. F. (2014*a*), Modeling Entity Evolution for Temporal Record Matching, *in* 'ACM SIGMOD', New York, NY, USA, pp. 1175–1186.

Chiang, Y.-H., Doan, A. & Naughton, J. F. (2014*b*), 'Tracking Entities in the Dynamic World: A Fast Algorithm for Matching Temporal Records', *Proceedings of the VLDB Endowment* **7**(6), 469–480.

Christen, P. (2012*a*), *Data Matching – Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*, Springer, Berlin Heidelberg.

Christen, P. (2012*b*), 'A Survey of Indexing Techniques for Scalable Record Linkage and Deduplication', *IEEE Transactions on Knowledge and Data Engineering* **24**(9), 1537–1555.

Christen, P. & Gayler, R. W. (2013), Adaptive Temporal Entity Resolution on Dynamic Databases, *in* 'PAKDD, Springer LNAI', Gold Coast, Australia, pp. 558–569.

de Vries, T., Ke, H., Chawla, S. & Christen, P. (2011), 'Robust Record Linkage Blocking Using Suffix Arrays and Bloom Filters', *ACM TKDD* **5**(2), 1–27.

Indyk, P. & Motwani, R. (1998), Approximate nearest neighbors: towards removing the curse of dimensionality, *in* 'ACM Symposium on Theory of Computing', Dallas, TX, USA, pp. 604–613.

Kim, H.-S. & Lee, D. (2010), HARRA: Fast Iterative Hashed Record Linkage for Large-scale Data Collections, *in* 'International Conference on Extending Database Technology', ACM, Lausanne, Switzerland, pp. 525–536.

Krueger, D., Montgomery, D. C., Peck, E. A. & Vining, G. G. (2015), *Introduction to Linear Regression Analysis*, John Wiley & Sons., Hoboken, NJ.

Kum, H.-C., Krishnamurthy, A., Machanavajjhala, A. & Ahalt, S. C. (2014), 'Social Genome: Putting Big Data to Work for Population Informatics', *IEEE Computer* **47**(1), 56–63.

Li, F., Lee, M. L., Hsu, W. & Tan, W.-C. (2015), Linking Temporal Records for Profiling Entities, *in* 'ACM SIGMOD', New York, NY, USA, pp. 593–605.

Li, P., Dong, X. L., Maurino, A. & Srivastava, D. (2012), 'Linking temporal records', *Frontiers of Computer Science* .

Rajaraman, A. & Ullman, J. D. (2011), *Mining of Massive Datasets*, Cambridge University Press, Cambridge.

Ukkonen, E. (1992), 'Approximate string-matching with q-grams and maximal matches', *Theoretical Computer Science* **92**(1), 191–211.

Wang, Q., Cui, M. & Liang, H. (2016), 'Semantic-Aware Blocking for Entity Resolution', *IEEE Transactions on Knowledge and Data Engineering* **28**(1), 166–180.

Witten, I. H., Moffat, A. & Bell, T. C. (1999), *Managing Gigabytes: compressing and indexing documents and images*, Morgan Kaufmann Publishers, San Francisco, Calif.

Augmenting Classification with Support Vector Regression for Boosting Financial Forecasting Returns

Mojgan Ghanavati[1,2] **Raymond K. Wong**[1] **Fang Chen**[1,2] **Simon Fong**[3]

[1] School of Computer Science, Engineering
University of New South Wales, Sydney, Australia
[2] Data61, Sydney, Australia
[3] University of Macau, Macau, China
Email: {mojgang, wong}@cse.unsw.edu.au

Abstract

With the popularity of data analytics, effective prediction approaches on a large amount of data become increasingly important. On the other hand, classification is a research area in data mining that has a long history. While many efficient classification algorithms have been developed, it can be difficult to interpret and apply their results for prediction applications especially for stock market prediction purposes. This paper investigates the benefits of classification method combining with regression approach to enhance prediction efficiency. In particular, data are first classified by hierarchical beta process based method before being projected by a local metric learning based support vector regression method. Experiments based on real stock market datasets show the effectiveness of our proposal. We also show that the prediction returns are further enhanced by considering other data sources such as news and overseas financial markets using a local metric learning based support vector regression method.

Keywords: Stock market prediction, Hierarchical beta process, Regression, Metric learning

1 Introduction

Time series analysis is to find correlations in data. Typical challenges faced by time series analysts include determining shapes of time series, forecasting their future trends, and classifying them to different categories. Solving these tasks can have significant contribution in many financial, economic and social applications. In particular, these tasks are probably the most important in the stock market prediction. Malkiel and Fama believe that historical stock prices can be efficiently used to predict their future trends (Fama 1970). However, stock market analysis is a complicated task due to its large, high-dimensional, non-normally-distributed and non-stationary time series data. Non-stationary means that the statistical properties such as mean, variance and/or autocovariance, change over time. New theories and methods are needed to handle these settings. In particular, forecasting stock market price movement trends has been a big challenge for a long time. Lots of works have been done on mining / forecasting financial time series data (Montgomery et al. 2015, Moskowitz et al. 2012, Qian et al. 2015, Shumway & Stoffer 2013).

Prediction accuracy in stock market forecasting is vital, since a wrong investment may cost a fortune in practice. In addition, most of the developed models for stock market prediction are parametric or semi-parametric (e.g. (Ausín et al. 2014)). Parametric models usually have a fixed model structure based on apriori assumptions on data behaviour. Also, these models are mostly unable to adjust to the complexity of problems. Considering these limitations, we adopt a Bayesian non-parametric model to predict the stock future trends. non-parametric models provide the flexibility that requires fewer assumptions about model structure. In this paper, we consider beta process (BP) which provides a Bayesian non-parametric prior for statistical models that involve binary prediction results. In our case, the prediction results include up and down trends which are presented with 1 and -1, respectively. Therefore, the model is used to predict the next trend of stocks more accurately by capturing specific patterns of different stock clusters.

In the literature, there are relatively few works that specifically address the classification of financial time series with a non-parametric approach (Lin et al. 2012). In this paper, we try to fill this gap by providing an HBP-based method (a non-parametric approach) suitable to predict stock trends.

However, it can be difficult to interpret and compare the results of classification approaches (especially binary classifications). For example, considering a binary classification approach, if we receive the same class label (e.g. positive) using two methods, it's hard to conclude which of them has better performance. To address this issue, we adopt a regression model to predict the price returns on those with positive labels. However, when the data is large and heterogeneous, distance learning (which is a crucial step in regression) becomes very challenging. Although several efficient metric learning methods have been proposed to measure the similarities between instances for classification, there is not much research done on metric learning for regression. In one of the first attempts to address this problem, metric learning for kernel regression (MLKR)(Weinberger & Tesauro 2007) was proposed. Earlier metric learning methods learn a global distance metric that captures the important features and their correlations in a global manner. However, when the data is complex, such as the finance data, global metric learning may not be able to fit the distance over the data manifold well. Local metric learning methods address this problem by learning a metric on each neighbourhood, i.e., learning different metrics across the given space. Based on these observations and to further investigate metric learning for regression (rather than for classification), we propose a local metric learning method for kernel regression. In this paper, a method called fuzzy-based, local metric learning for non-linear support vector regression (SVRML) is proposed. Firstly, fuzzy c-means is used to partition the training data into different clusters. Then, a local metric is learned for each kernel in each cluster. In this paper, radial basis function (RBF) kernel is employed. Finally, selected

stocks picked from our HBP-based method are used as the input of SVRML so that they are ranked based on their returns.

To summarize, our primary contributions in this paper are:

- Propose an flexible financial return price prediction framework - that can utilize multiple data sources and machine learning methods including non-parametric approaches
- Augment an HBP-based method with a local metric learning based regression to predict stock trends
- Analyze the impact of classification on regression results for better investment

The remainder of this paper is organized as follows. Section 2 presents the related works. The proposed framework and our non-parametric approach are described in Section 3. Section 4 shows the experiment results, and finally Section 5 concludes the paper.

2 Related Works

A financial market that has been heavily studied and analysed by different methods is stock market. Many works have been done in mining the financial markets with having a common aim of predicting stock market trends. One of the difficult challenges faced by analysts is to model the behaviour of human traders. Constant changing of their behaviour patterns has made prediction difficult. To address this problem, researchers have used a variety of approaches. A large group of researchers put the problem into a machine learning framework, and believed that historical trading volume and pricing gave enough information to predict future trends (Huang et al. 2008, Lin et al. 2013). Another group of researchers thought that there were other sources which might have greater impact on the trading behaviour than historical price. They have done various research and evaluated different sources to prove their claims (Benthaus & Beck 2015, Schumaker et al. 2012). We review the most popular and the most recent works in this section. At first, various efforts on applying different machine learning techniques and models are reviewed. Then, recent works that considered different data sources are briefly presented.

(Huang et al. 2008) used a wrapper approach to choose the best feature subset of 23 technical indices and then combined different classification algorithms to predict the future trends of the Korea and Taiwan stock markets. (Yu et al. 2009) proposed a support vector machine (SVM) based model called least squares support vector machine (LSSVM) to predict stock market trends. They have used genetic algorithms, first to select the input features for LSSVM and then to optimize LSSVM parameters. To evaluate the efficiency of the proposed model, the S&P500 index, Dow Jones industrial average (DJIA) index, and New York stock exchange (NYSE) index, have all been used as testing targets. (Lin et al. 2013) proposed a two-part SVM-based approach to predict stock market trends. In the first part of the proposed method, a correlation-based SVM is applied to technical indicators to select the best subset of features. In the second part, a quasi-linear SVM is used to predict the direction of stock market movements, considering the selected subset of the indicators as an input. (Patel et al. 2015) compared the application of four models including the artificial neural network (ANN), SVM, random forest and Naive Bayes for stock market trend prediction. Two different approaches have been taken to provide the inputs for these models. In the first approach, ten technical parameters were computed based on stock price data while the second approach focused on representing these technical parameters as trend deterministic data. To evaluate the efficiency of these models, 10 years of historical data of two stocks, namely Reliance Industries and Infosys Ltd., and two stock price indices, CNX Nifty and S&P Bombay Stock Exchange (BSE) Sensex, have been used. (Li, Jiang, Li & Chen 2015) proposed a supervised regression approach to discover the patterns between stock movements and different information sources. (Sheta et al. 2015) presented a model using multi-gene symbolic regression genetic programming(GP) to predict the return of S&P500 index. Multi-gene GP evolves linear combinations of non-linear functions of the input variables. Their proposed model presents robust results in comparison with the traditional forecasting models. (Prema et al. 2016) investigated the effectiveness of a Neural Network model improvised by using Genetic Algorithm to initialize synaptic weights. Their proposed system was used to predict BSE Sensex closing price.

In one of the important attempts in using news articles, (Schumaker et al. 2012) implemented a machine learning approach called AZFinText to learn historical price movements by using the sentiment of the terms used in financial news articles and also the tone of the author. After that, several works were done to improve the modelling and analysing process of news articles. (Li, Xie, Chen, Wang & Deng 2014) first setup a generic framework that took market information and used different prediction models to analyse the input and predict the future trend. They then implemented various prediction models considering different sentiment analysis approaches including bag-of-words and sentiment polarity approaches, comparing the accuracy of their daily stock price return prediction with five years of Hong Kong Stock Exchange market data. They further improved the prediction accuracy in their next paper (Li, Xie, Song, Zhu, Li & Wang 2015) by applying a sentence-level summarization model on news articles to extract useful information and to filter noise for prediction enhancement. Afterwards, they fed their framework with both summarized and full length articles which showed that summarization can effectively improve the prediction accuracy. In their most recent attempt on stock market prediction (Li et al. 2016), they focused more on the speed of forecasting. Thee therefore adopted extreme learning machine (ELM) as an emergent supervised technique to integrate market news and prices in their proposed framework. (Benthaus & Beck 2015) analysed the influence of social broadcasting on the stock market. Their results showed that the message and discussion dimensions have a significant influence on stock markets whereas user dimensions did not to such an extent.

Not much work has been done to specifically address the classification of financial time series using non-parametric approaches (Lin et al. 2012). A new nonparametric test was introduced in (Lee & Mykland 2008) to detect jump arrival times and realized jump sizes in asset prices up to the intra-day level. They showed that using high-frequency returns, the likelihood of misclassification of jumps could be ignored. (Durante et al. 2015) proposed a non-parametric method to cluster the time series according to their tail dependence behaviour. The usefulness of their proposed approach was shown using financial data. Hierarchical beta process (HBP) was first used in (Li, Zhang, Wang, Chen, Taib, Whiffin & Wang 2014) to assess water pipe conditions. Recently, the combination of HBP with swarm clustering method has been applied for financial time series trend prediction in (Ghanavati et al. 2016).

3 Proposed framework and approach

In this section our proposed framework and non-parametric approach are described. Figure 1 shows the system architecture diagram. In the proposed framework, various sources are accepted as input, each source is preprocessed separately, the preprocessed sources are integrated, a model is learnt using various classiffication methods and the future movement direction of each stock is predicted. Top trend stocks are used as the input of various Regression methods and the return price of each stock is predicted. At the end top selected stocks can be ranked based on their actual return prices. The framework gives the opportunity to compare the results with other popular regression/classification methods. User can choose the methods to compare with and the way of presenting the results. Key notations used in this section are listed in Table 1.

Table 1: Notations

Symbol	Description
t	A trading day
r_t	Open-to-Close Return in a trading dat t
l_t	Class lable at trading dy t
c	Number of clusters
q_k, f_k	Mean and concentration parameteres for cluster k
$K()$	Kernel function
d	Attribute index
x_i	Instance i
z_ik	History of i^{th} stock in cluster k
w_{ik}	Membership degree of instance x_i to the cluster k
α	The fuzzifier
M_k	Local metric of cluster k
M_i	Local metric of instance i

3.1 Preprocessing

To reflect actual trend of the stock market, we use various financial indicators. In this work we use monthly prices as a default duration for the indicators (since we are more interested in the longer term trend). The selected indicators are introduced in the experiment section. For each time span, we consider the first open to last close price return as the ground truth label in the prediction.

News summaries do not need any preprocessing since they are manually summarized and labelled with stock IDs. To tag the news articles with stock IDs, we prepare a list of stock names and IDs. News containing more than one stock name or ID, is tagged with the stock ID with higher frequency. If all the stocks in one news article have equal frequencies, the news will be removed from the list.

3.2 News sentiment analysis

To declare the sentiment of both news articles and summaries, we carried out the following procedure. First the OpenNLP sentence detection tool was applied to extract the sentences. Then OpenNLP tokenization tools were used to segment each sequence into tokens. Tokens are usually words, punctuation, numbers, etc. After that the 2014 version of Loughran and McDonald Sentiment Word Lists was applied to tag the proper words and declare their sentiments in each sentence. The reason for choosing this dictionary was that the Loughran and McDonald word list is a financial domain specific dictionary which is manually constructed. Manually constructed dictionaries are usually more accurate than automatic ones because the words have chosen by linguistic experts (Loughran & McDonald 2011). A vector of sentences is created for each news article/summary. Each row of sentence vector has 2349 columns for negative words, 354 columns for positive words, 8 columns for negation words from (Prollochs et al. 2015) and one column to record the total number of words in each sentence. Each column is filled with the number of corresponding words in the sentence. We add one more column to this matrix at the end, for the sentiment of each sentence which is declared using the following formula:

$$Sentiment_s = \frac{\sum \text{positive words} - \sum \text{negative words}}{\text{Total no. of words}} \tag{1}$$

After that, the sum of all negation words in each sentence is calculated. If an odd number of negation words occur in a sentence, the sentence sentiment is inverted. An even number of negation words has no impact on the sentence sentiment. At the end, a news sentiment is declared as sum of sentiment of all the sentences divided by total number of sentences in the news.

3.3 Putting news sentiments and technical indicators together

After preparing the technical indicators and news sentiments, the inputs for prediction methods need to be prepared. The inputs are prepared in two different ways. In the first setup, the input is technical indicators alone. In the second setup, sentiments from news are combined with technical indicators and used as the input of prediction methods. As there are differences in the average number of published news each day for different stocks, we use n-day simple moving average (SMA) to adjust the sentiment values. Therefore the sentiment value for each day t for each stock is calculated as follows:

$$Sentiment_t = \text{SMA of positive news} - \text{SMA of negative news} \tag{2}$$

where we use 22-day SMA ($SMA22$) to calculate monthly moving averages respectively.

3.4 Hierarchical beta process (HBP)

Hierarchical beta process (HBP) is applied on the clustered stocks to predict the future stock trends. A beta process, $B \sim BP(f, B_0)$, is a positive random measure on a space Ω, where f, the concentration function, is a positive function over Ω, and B_0, the base measure, is a fixed measure on Ω. If B_0 is discrete, $B_0 = \sum_k q_k \delta_{w_k}$, then B has atoms at same locations $B = \sum_k p_k \delta_{w_k}$, where $p_k \sim Beta(f(w_k)q_k, f(w_k)(1-q_k))$, and each $q_k \in [0,1]$. An observation data X could be modelled by a Bernoulli process with the measure $B, X \sim BeP(B)$, where $X = \sum_k z_k \delta_{w_k}$, and each z_k is a Bernoulli variable, $z_k \sim Ber(p_k)$. Furthermore, when there exists a set of categories, and each data belongs to one of them, hierarchical beta process could be used to model the data. Within each category, the atoms and the associated atom usage are modelled by a beta process. Meanwhile a beta process prior is shared by all the categories. More details could be found in Thibaux and Jordan (Thibaux & Jordan 2007). For a stock market, denote π_{ki} as the probability of price going up for the i^{th} stock in the k^{th} group. Considering hierarchical construction for stock market assessment,

$$\begin{aligned} &q_k \sim Beta(f_0 q_0, f_0(1-q_0)), \quad where \quad k = 1, 2, ..., \\ &\pi_{ik} \sim Beta(f_k q_k, f_k(1-q_k)), \quad where \quad i = 1, 2, ..., \\ &z_{ijk} \sim Ber(\pi_{ik}), \quad where \quad t = 1, 2, ... \end{aligned} \tag{3}$$

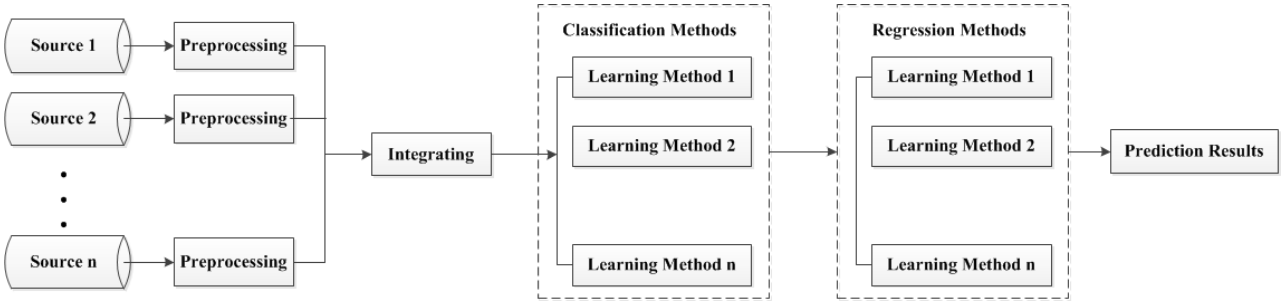


Figure 1: System Architecture

q_k and f_k are the mean and concentration parameters for the k^{th} group. q_0 and f_0 are hyper parameters for the hierarchical beta process. $z_{ki} = \{z_{kit} | t = 1, 2, ...\}$ is the history of i^{th} stock in the k^{th} group. $z_{ijk} = 1$ means the stock price has been increased in the j^{th} timestamp, otherwise $z_{ijk} = 0$. For hierarchical beta process, a set of q_k are used to describe increasing rates of stock prices in different groups. Following the work of (Li, Zhang, Wang, Chen, Taib, Whiffin & Wang 2014), we consider both mean and concentration parameters, which is important for more effective assessment. Therefore, for each group, given both f_k and q_k, we have:

$$\pi_{ki}|q_k, f_k, z_{ki} \sim Beta(f_k q_k + \sum_j z_{ijk}, f_k(1-q_k) + m_{ik} - \sum_j z_{ijk}) \tag{4}$$

The approximate inference algorithm is shown in Algorithm 1.

Algorithm 1 Hierarchical beta process for stock market trend prediction

Input: Stock up trend history $\{z_{kit}\}, k = 1, 2, ...; i = 1, 2, ...; j = 1, 2, ..., m_{ik}$; hyper parameters f_0 and q_0
The probability of price increasing π_{ki} for each stock
while $t \leq T$ **do**

$$q_k = \frac{f_0 q_0 + \sum_i s_{ik}}{f_0 + \sum_i s_{ik} + \sum_i \sum_{l=0}^{m_k - 1 - s_{ik}} \frac{f_k}{1+f_k}}$$

where $s_{ik} = \sum_j z_{ijk}$;

$$\pi_{(t)}^{ik} = \frac{f_k^{(t)} q_k^{(t)} + \sum_j z_{ijk}}{f_k^{(t)} + m_{ik}} \quad t = t + 1;$$

end while
calculate $\bar{\pi}_{ik} = \frac{1}{T} \sum_{t=1}^{T} \pi_{(t)}^{ik}$
Return $\bar{\pi}_{ik}$

3.5 Fuzzy-based local metric learning for non-linear SVR (SVRML)

In this subsection, SVRML is proposed as a metric learning based regression method. Given a set of training samples (x_i, y_i), support vector regression (SVR) solves the primal problem (as in (Smola & Schölkopf 2004)):

$$min_{w,\xi_i,\xi_i^*} \quad \frac{1}{2}\|w\|^2 + B\sum_{i=1}^{N}(\xi_i + \xi_i^*) \quad s.t. \begin{cases} y_i - wx_i - b \leq \epsilon + \xi_i \\ b + wx_i - y_i \leq \epsilon + \xi_i^* \\ \xi_i^* \xi_i \geq 0 \end{cases} \tag{5}$$

where $B > 0$ is constant and ξ_i and ξ_i^* are the corresponding positive and negative errors at the i^{th} point, respectively. Practically, the dual formulation of Eq. 5 is used to solve the primal problem.

$$min_{\alpha,\alpha^*} \quad \frac{1}{2}(\alpha - \alpha^*)^T K(x_i, x_j)(\alpha - \alpha^*) + \epsilon\sum_i(\alpha_i + \alpha_i^*) - \sum_i y_i(\alpha_i - \alpha_i^*) \quad s.t. \ \sum_i(\alpha_i^* - \alpha_i) = 0, \quad \alpha_i, \alpha_i^* \geq 0 \quad i = 1, 2, ..., \tag{6}$$

In non-linear SVR, a kernel function ($K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$) transforms the data into a higher dimensional feature space to make it possible to perform the linear separation. Incorporating the Mahalanobis distance (8) into the kernel function, we have:

$$K_L(x_i, x_j) = e^{-(x_i - x_j)^T L^T L (x_i - x_j)} \tag{7}$$

$$d_M(x_i, x_j) = \sqrt{(x_i - x_j)^T \cdot M \cdot (x_i - x_j)} \tag{8}$$

A linear metric learning method learns the positive semidefinite metric $M = L^T L$, based on some constraints. Although these settings are effective in common applications, they are designed in a global fashion and lack the flexibility to capture the local trend in some applications, in particular in stock markets data. In the context of financial time series prediction, the data are usually highly volatile and the associated variance of noise varies over time. In order to address this issue, we propose the local metric learning for support vector regression (SVRML) model. We will show that by taking the local data trend into consideration, our model provides a systematic and automatic scheme to adapt the margin locally and flexibly.

As a first step, we partition the training data into different clusters using fuzzy c-means (FCM). Each data point has membership degree to each cluster. All the membership degrees are stored in a weight matrix w with data points as the rows and columns as clusters. Our aim is to learn a metric for each kernel in each cluster. We denote different metrics by $M_1, M_2...$. The metric M_i is parameterized for instance x_i with the instance membership degree. We adjust the dual formulation of Eq. 5 to learn local metrics for local kernels with following optimization problem:

$$min_{\alpha,\alpha^*,L} \quad \frac{1}{2}\sum_k \sum_{i,j\in k}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K_{Mk}(x_i, x_j)$$
$$+\epsilon\sum_k\sum_{i\in k}(\alpha_{ik} + \alpha_{ik}^*) - \sum_k\sum_{i\in k} y_{ik}(\alpha_{ik} - \alpha_{ik}^*) + \lambda\sum_k \|M_k\|_2$$
$$s.t. \ \sum_i(\alpha_i^* - \alpha_i) = 0, \quad \alpha_i, \alpha_i^* = 0 \quad i = 1, 2, ..., \tag{9}$$

where $w = \sum_k \sum_{i\in k}(\alpha_i - \alpha_i^*)\phi_k(x_i)$, $M_k = L_k^T L_k$ and α, α^* are dual variables. Final prediction function is approximated by a linear combination of weighted distance based kernels:

$$f(x) = \sum_k (W_k^T \phi_k(x) + b_k) = \sum_k \sum_{i,j\in k}(\hat{\alpha} K_{L_k}(x_i, x_j) + b_k + \|M_k\|_2) \tag{10}$$

where $\hat{\alpha} = \alpha_i - \alpha_i^*$. Applying chain-rule to the derivative of $f(x)$ (e.g. equation 10) results in:

$$\frac{\partial f(x)}{\partial L} = \frac{\partial f(x)}{\partial k_{L_k}} \frac{\partial K_{L_k}}{\partial L_k} \tag{11}$$

where $K_{L_k} = K_{L_k}(x_i, x_j)$

$$\frac{\partial f(x)}{\partial L} = (\frac{1}{2}c(c - 2n - 1)\hat{\alpha}) * (c(c - 2n - 1). L_k x_{ij}^2 e^{-x_{ij}^2 L_k^2}) = c^2(c - 2n - 1)^2 \hat{\alpha} L_k x_{ij}^2 e^{-x_{ij}^2 L_k^2} \tag{12}$$

where $x_{ij} = x_i - x_j$, c is the number of clusters and n is the number of data points in each cluster. Algorithm 2 shows pseudo-code of the proposed SVRML.

Algorithm 2 SVRML pseudo-code

```
Grouping the data points using FCM
Initializing L_c for each cluster using PCA
while cost ≥ ϵ do
    Compute kernel matrix K_c from L_c
    Call SVR with K_c to obtain w_c and b_c
    Update metrics L_c using Eq. 12
end while
```

4 Experiments and Results

4.1 Settings

Three datasets (Hang Seng Index (HSI), Hong Kong shares and Standard & Poor's 500 (S&P500) Index) have used in the experiments. We also have access to a premium stock database of listed companies that contains, for example, market caps (the number of shares released to the market x its current share price) and also the market cap ranking of each stock. To find the most effective indicators on trend movement, five years data of the 53 largest stocks of Hong Kong is considered. Then five years monthly prices of total 1328 stocks in Hong Kong is used to evaluate the proposed HBP-based trend prediction method. The historical data is daily and covers the period between June 2010 and June 2015. The monthly prices between 2010 and 2014 are used to train the model and predict the movement trends of the stocks in the first month of 2015. After that, the stocks with up trend movement prediction are chosen to be used as the input of the SVRML method.

In addition, five years monthly prices of HSI between 2010 and 2015 are used to predict the returns of HSI in first month of 2015. In this experiment, our aim is to investigate the impacts of different sources including indicators, news sentiments, the number of published news and also US S&P500 on HSI, in addition to the performance evaluation of the proposed SVRML.

For each stock in HSI, we have access to a repository of financial news articles and news summaries of $1-2$ sentences. These summaries are prepared by human experts and manually tagged with their relevant stock IDs. We collected all news summaries of all the stocks in HSI for five years between 2010 and 2015.

All the experiments have been performed on a computer with Intel Core $i5$ (2.6GHz CPU) and 8GB RAM running Matlab $R2012b$, 64bit on Windows 7.

4.2 Selection of indicators

In this subsection, various feature selection techniques are applied to select the most effective subset of financial indicators. We collected 79 various financial indicators that affect the stock market trend prediction by reviewing the previous studies. However some indicators may have more contribution on the stock future trends (Xing et al. 2009). Therefore, we applied different feature selection methods to find an optimum subset of financial indicators for the training step. The average of the results of applying different feature selection and ranking techniques before using SVM on 50 large stocks of Hong Kong has been shown in Table 3. The number of selected top indicators in each run using various techniques has been shown by variable n in the Table 3. The first 5 rows show the average results of using PCA as a dimension. The best result is achieved when taking the top 20 dimensions. Note that PCA reduction technique and using SVM to identify the trend movement. Although the results are promising, PCA assumes independent observations so its use in a time series context is not reasonable. The next 5 rows of the table present average results of ranking the indicators based on their relationships with the trend variable using cross-correlation coefficient(CC). The correlation between indicators haven't been considered in the ranking process, therefore it is no wonder that it does not give the best results compared to the other methods. In the next 4 rows of the table, t-test and Wilcoxon signed-rank test have been used to assess the significance of each indicator respectively. The best results are achieved when cross-correlation coefficient weighting (CCW) is used to rank the indicators. In this technique, correlation information is used to outweigh the Z value of potential indicator using $Z * (1 - Alpha * (RHO))$ where RHO is the average of the absolute values of the cross-correlation coefficient between the candidate indicator and all the previously selected indicators. $Alpha$ sets the weighting factor which is a scalar value between 0 and 1. When $Alpha = 0$ potential indicators are not weighted. A large value of RHO (close to 1) outweighs the significance statistic; this means that indicators that are highly correlated with the indicators already picked are less likely to be included in the output list. Based on the results, selected indicators in the output list of CCW where $Alpha = 0.5$, have been chosen in all the following experiments. The selected indicators are lister in Table 2.

4.2.1 Technical indicators-Discrete representation

Following the work of (Patel et al. 2015), the effectiveness of coverting technical indicator values to discrete trend representation based values is invistigated. Therefore, each indicator value is converted to up and down categories based on their inherent property through which traders generally predict the stocks up or down movement. For example, in the conversion process, considering 30

Table 2: Selected technical indicators

1. Aroon	9. Signal Index
2. Rate of Change	10. Wildmer's DMI (ADX)
3. True Strength Index	11. Relative Strength Index (RSI)
4. Force Index	12. KDJ Indicator
5. Money Flow Index	13. Acceleration
6. William's %R	14. K/D Stochastic Oscillator
7.Commodity Channel Index (CCI)	15. Accumulation/distribution line
8. Moving Average Convergence Divergence (MACD)	16. Simple/Weighted Moving Average (SMA/WMA)

Table 3: Average results of applying various dimension reduction and feature ranking techniques to predict the movement trends of 53 large stocks in Hong Kong

	Error Rate	**FScore**	**Precision**	**Recall**	**AUC**
PCA(n=10)	0.15	0.85	0.84	0.86	0.84
PCA(n=20)	0.08	0.92	0.92	0.93	0.92
PCA(n=30)	0.12	0.89	0.87	0.92	0.87
PCA(n=40)	0.13	0.88	0.86	0.92	0.86
PCA(n=50)	0.14	0.88	0.85	0.90	0.85
CC (n=10)	0.13	0.88	0.86	0.91	0.86
CC (n=20)	0.17	0.83	0.84	0.86	0.82
CC (n=30)	0.16	0.84	0.85	0.85	0.84
CC (n=40)	0.15	0.85	0.87	0.86	0.85
CC (n=50)	0.13	0.88	0.89	0.89	0.87
ttest(n=50)	0.11	0.89	0.89	0.90	0.88
ttest(n=40)	0.11	0.90	0.90	0.91	0.89
entropy(n=50)	0.10	0.90	0.91	0.90	0.89
entropy(n=40)	0.10	0.90	0.91	0.91	0.90
wilcoxon(n=40)	0.10	0.91	0.91	0.91	0.90
CCW(Alpha=0)	0.072	0.93	0.93	0.93	0.92
CCW(Alpha=0.5)	**0.06**	**0.93**	**0.94**	**0.94**	**0.93**
CCW(Alpha=0.7)	0.07	0.93	0.93	0.93	0.92
CCW(Alpha=1)	0.07	0.93	0.93	0.93	0.92

days mobing average, if the current price is higher than the moving average the trend is up, represented as 1. If the price is below a moving average the trend is down and represented as -1. William's %R, MACD and K/D stochastic oscillators follow the trend of the stock. If the value of William's %R, MACD and K/D stochastic oscillators at time t is greater than the value at time $t-1$ (goes up), the trend is up (represented as 1) and vice-a-versa. RSI ranges between 0 and 100. The value over 70 means the stock is overbought, so, it may go down in near future (down trend). Tthe value less than 30 means the stock is oversold. Therefore, the stock may go up in near future (up trend). For the values are between 30 and 70, RSI follows the trend of the stock. Therefore, if RSI value at time t is greater than its value at time $t-1$ (goes up), the trend is up and vice-a-versa. Momentum also follows the trend of the stock. Positive value of momentum indicates up trend, while negative value indicates down trend.

4.3 Performance of HBP

In this subsection, the proposed HBP method is evaluated and compared with popular trend prediction methods. Five years data of 1328 stocks in Hong Kong is used to evaluate the proposed trend prediction method. To predict the trend of stocks for a given month, all the available records before that month are used as training data that is a popular way of using HBP for prediction. The stock market trend prediction problem is defined as a binary classification problem where 1 and -1 represent the up and down trends respectively. Table 4 shows the performance diagnostics including sensitivity, specificity, f-measure and AUC by different methods. Similar indicators have been considered in all models for fair comparison. HBP-Discrete representation shows the results of applying HBP when the technical indicator values are converted to up and down categories. The results suggest that discritizing technical indicator values based on their inherent property improves the performance dramatically especially in terms of precision.

Table 4: Performance of different methods for stock market trend prediction

	Recall	**Precision**	**FScore**	**GMean**	**AUC**
HBP	0.53	0.72	0.51	0.52	0.52
HBP-Discrete representation	0.48	0.93	0.64	0.64	0.47
SVM(Linear)	0.51	0.73	0.49	0.43	0.44
SVM(Polynomial)	0.60	0.37	0.48	0.41	0.42
LMNN	0.33	0.61	0.28	0.30	0.22

To investigate the significance of the results shown in Table 4, Friedman and Nemenyi tests (Demšar 2006) have been chosen to further analyse the results. Friedman test is used to compare different classification results according to their similarity. Friedman test, as a non-parametric test, is more suitable than parametric tests when more than two classifiers are involved (Demšar 2006). Friedman test compares the average ranks of algorithms under a *null*-hypothesis. The *null*-hypothesis assumes that all the algorithms perform the same and hence their output ranks should be the same. Table 5(a) presents the results of Friedman test and hence we reject the hypothesis, i.e., the algorithms do not perform the same. Friedman test has been performed considering the results of 10 times performing all three methods on stock market data. The χ_F^2 values namely Chi-Square more than 0 with $p < 0.05$ for all the performance diagnostics suggest that there are significant differences between performance measurements values of these three methods (HBP, SVM(Linear) and LMNN). In other words, these results illustrate that even if we perform these methods several times on different data samples, the performance will not be similar and their performances can be ranked.

When the *null*-hypothesis is rejected, we can proceed with a post-hoc test. Nemenyi tests are used here to show the difference between each pair of the algorithms and rank the methods based on their performance. In other words, Nemenyi test is a post-hoc test for pairwise comparisons (Table 5(b)) that are used after Friedman test. Nemenyi test is to test the same hypothesis as Friedman's but just between two methods at a time. As shown in the comparison between HBP and SVM(Linear) in Table 5(b), critical value (q) more than 0 with $p < 0.05$ rejects the *null*-hypothesis. This means that HBP performs better than SVM with linear kernel on all measurements. Similarly, as shown in the comparison between HBP and LMNN in Table 5(b), q more than 0 with $p < 0.05$ implies that KDFuzzyML outperforms Weibull. The higher the critical value is, the bigger difference between their performances is. For instance, for AUC, the critical value from Nemenyi on HBP and LMNN is 6.53, which is larger than the critical value from Nemenyi on HBP and SVM(Linear) (i.e., 5.99). This implies that for AUC, HBP performs the best, followed by SVM(Linear) and then LMNN. This is consistent with the results shown in Table 4.

4.4 Performance of SVRML

Figure 2 compares the root-mean-square error (RMSE) of applying three regression methods including linear regression (LR), support vector regression (SVR) and the proposed SVRML on HIS index. Five years monthly prices of HSI between 2010 and 2014 has been used to train the model and predict the return price of HSI in first month of 2015. Left bar of each method is when only the selected indicators are used to train the model while the right bar is when the return price of S&P500 on the previous day is added to the HSI indicators for training the model.

4.5 Evaluation of the classification-regression strategy

In this subsection we show how financial return can be increased by integrating the classification and regression techniques. We run various expriments in different subsets of stock markets using different classification and regression combinations. The summary of the results applying SVM and MLKR on 73 large stocks of Hong Kong is shown in Table 6. By applying SVM to find the trend movement in first month of 2015, 38 stocks showed up trend. Considering buying all the 38 stocks at begining of the month and selling them at the end of the month, the average return will be 3.79% per stock (first column). If the selected stocks are ranked based on their total share in the market and top 10 stocks are chosen for investment, the average return is 3.02%. Third column is when the up trend stocks are ranked based on their ranking in the Hong Kong market and the top 10 are chosen for investment. Second row shows the results of applying MLKR as a regression method to predict the financial return of same 73 large stocks. Based on MLKR results, 35 stocks will have positive returns. Considering investing on all the 35 stocks, the average return will be 4.25% per stock. The last strategy is using the 38 up trend stocks getting from SVM (classification) as input of the regression technique. Out of 38 stocks, 32 stocks showed positive return which will give 4.65% average return.

Table 7 presents the average return of using HBP, SVRML and the combination of both on 3 years historical data of all the 1374 stocks in Hong Kong. HBP is used to predict the condition of all the stocks in the first

Table 5: Comparison between the proposed method and other methods using (a) Friedman test and then (b) Nemenyi test

(a) **Friedman test**		(b) **Nemenyi test**		
Metrics	**Critical values**	**Metrics**	**HBP vs. SVM**	**HBP vs.LMNN**
Sensitivity	χ_F^2=12.15, p<0.05	Sensitivity	q=3.53, p<0.05	q=4.48, p<0.05
Specificity	χ_F^2=13.14, p<0.05	Specificity	q=4.43, p<0.05	q=5.24, p<0.05
FScore	χ_F^2=23.22 , p<0.05	FScore	q=7.94, p<0.05	q=7.78 p<0.05
AUC	χ_F^2=16.37, p<0.05	AUC	q=5.99, p<0.05	q=6.53, p<0.05

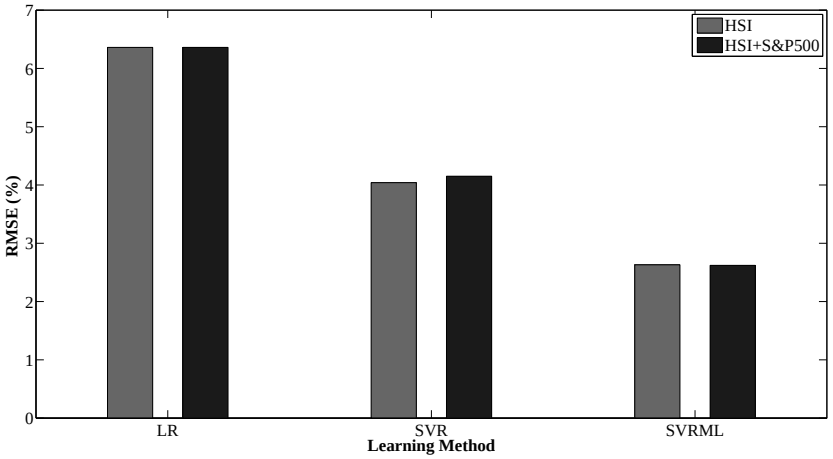


Figure 2: RMSE of applying SVRML on HSI index

Table 6: Average return (in %) of appying different methods on 73 large stocks in Hong Kong

	All Up Trends	**Top 10 Share**	**Top 10 Rank**
SVM	3.79	3.02	4.24
MLKR	4.25	2.63	4.83
SVM+MLKR	4.65	3.37	4.84

month of 2015. Out of 1374 stocks in Hong Kong market, 689 stocks will go up. Considering buying all the up trend stocks at begining of the month and selling them at the end of the month, the average return will be 12.38% per stock. If the selected stocks are ranked based on their total share in the market and top 10 stocks are chosen to for investment, the average return is 7.64%. The best investment senario is when the up trend stocks are ranked based on their ranking in the Hong Kong market and the top 10 are chosen for investment with 13.15% average return. The results of applying SVRML as a regression method to predict the financial return is shown in the second row of the table. After applying SVRML, 964 stocks show positive returns. Considering investing on all the 964 stocks, the average return is 18.36% per stock. If top 10 stocks are chosen, the average return increases dramatically to 41.99%. At the end, the 689 up trend stocks getting from HBP as the classification algorithm are used as input of the regression technique. Out of 689 stocks, 606 stocks showed positive return. Investing on all these positive return stocks results in 28.78% average return. The last columns for SVRML and HBP+SVRML is when we rank the stocks based on the predicted returns and invest on the top 10 stocks. It can be seen that by integrating classification with regression, the average return is increased by 12.47% (273.68 vs 261.21).

Table 7: Average return (in %) of appying different methods on all the stocks in Hong Kong

	All Up Trends	**Top 10 Share**	**Top 10 Rank**	**Top 10**
HBP	12.38	7.64	13.15	
SVRML	18.36	9.26	41.99	261.21
HBP+SVRML	28.78	21.91	31.38	**273.68**

Table 8 presents the average return of using HBP, SVRML and the combination of both on 3 years historical data of the stocks with different sizes in Hong Kong. The best return is achieved by applying classification-regression method on all the small size stocks in Hong Kong (34.41). Overall, classification-regression gives the best results on stocks with any sizes (e.g. average return of $(15.14, 24.28, 34.41)$ vs $(8.12, 14.57, 19.26)$ and $(15.14, 24.28, 34.41)$ vs $(6.06, 12.21, 16.84)$).

Table 8: Average return (in %) of appying different methods on the stocks with different sizes in Hong Kong

	Large	**Medium**	**Small**
HBP	6.06	12.21	16.84
SVRML	8.12	14.57	19.26
HBP+SVRML	15.14	24.28	**34.41**

Cosidering all the stocks in Hong Kong market, Figure 3 shows the overal return (axis y) by investing on top n (axis x) stocks ranked by different methods.

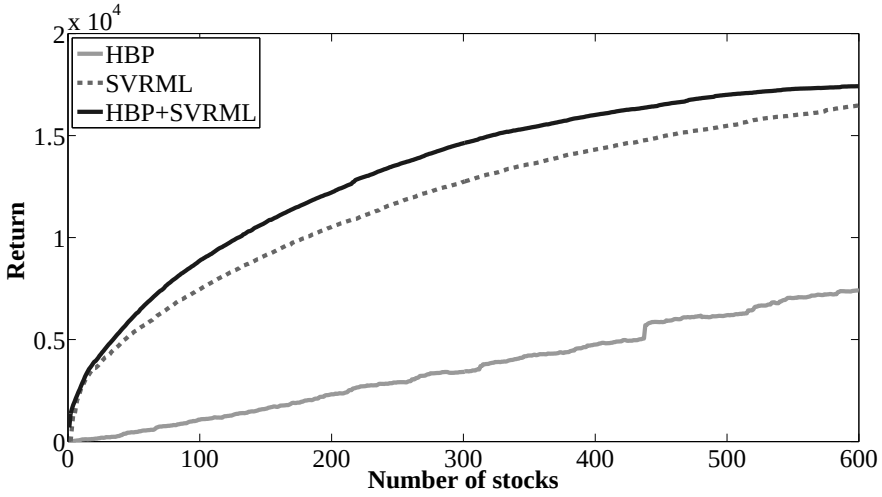


Figure 3: Sum of the stock returns using different methods

4.6 Different data sources

To investigate the impact of different sources including financial indicators, news summaries and S&P 500 index on HSI trend movement, we consider various scenarios of using these sources. The results are shown in Table 9. Error rate, FScore and AUC are used as evaluation measurements. The table is divided into three subsections. In the first subsection, the first row shows the results of applying SVMRML on HSI considering only selected indicators. In the next three rows, it can be seen that adding news sentiments, number of news and both news sentiments and numbers in each trading day may not help the prediction performance in general. In the 'After 4pm News' section, the impact of news is improved by considering the published news after 4pm in each trading day as the next day news. The results show that this small change improves the prediction performance in practice. In the last section, the impact of US S&P500 index trend movement in the previous day has been shown. The results show that integrating the SP500 index trend movement in the previous day and sentiment of the news on HSI stocks with HSI indicators is the best scenario.

Table 9: Comparison between the prediction results on HSI considering different data sources using SVRML

	Error Rate	**FScore**	**AUC**
Indicators Only	0.17	0.87	0.82
Ind.+NewsSentiment	0.19	0.86	0.77
Ind.+NewsNo	0.17	0.89	0.8
Ind.+NewsSen.+NewsNo	0.18	0.86	0.78
After 4pm News			
Ind.+NewsSentiment	0.11	0.93	0.79
Ind.+NewsNo	0.14	0.91	0.76
Ind.+NewsSen.+NewsNo	0.12	0.91	0.89
Adding SP trend-Day Before			
Indicators+SP	0.11	0.93	0.88
Ind.+NewsSen.+SP	0.07	0.95	0.95

4.7 Generality of proposed method

To show the generality of the proposed framework, it is also applied for horse race gambling prediction. We have access to the data of 2319 different races in 3 years between 2013 and 2016 from Hong Kong Jockey Club. The data consists of 1997 racehorses, 90 jockeys and 42 trainers. The data consist of information about each horse in each race and are of high dimension (247 features in total). Figure 4 shows the final return by betting in 1 to 50 races considering final odds, classification (SVM) results and winnets based on classification plus regression (SVM+MLKR) results. As Figure 4 shows, using regression to rank the winners after classification will produce more stable results.

4.8 Overall results

In summary, as shown in Table 4, higher recall, FScore, GMean AUC and lower specificity indicate that HBP performs better than traditional approaches such as SVM (e.g., recall, fscore, Gmean and AUC of $(0.54, 0.52, 0.55, 0.55)$ vs $(0.51, 0.49, 0.43, 0.44)$) and Specificity of $(0.32$ vs $0.33)$. As shown in Figure 2, SVRML produces lower RMSE (e.g. RMSE of 2.63 with SVRML vs 4.04 with SVR and 6.36 with LR) than the others. This means that using SVRML, the predicted returns are closer to the actual returns when compared with traditional regression methods. The experiment has also shown that the effect of considering S&P500 for HSI is better using SVRML when compared to the SVR (e.g. RMSE of 2.62 vs 4.15). The effect of considering different data sources has been investigated and shown in Table 9. As shown in the table, considering the published news in each day in addition to the indicators might not improve the performance in general. However, if the impact of published news after 4pm are considered for the next day, using both news sentiments and number of the published news improves the performance (e.g. error rate of 0.12 vs 0.17); and higher FScore and AUC (e.g FScore and AUC of $(0.91, 0.89)$ vs $(0.87, 0.82)$). As also shown Table 9, considering the S&P500 trend movement in the day before each trading day, in addition to the daily news sentiments, produces the most accurate trend prediction on HSI (e.g., the lowest error rate of 0.07 and highest FScore and AUC of $(0.95, 0.95)$).

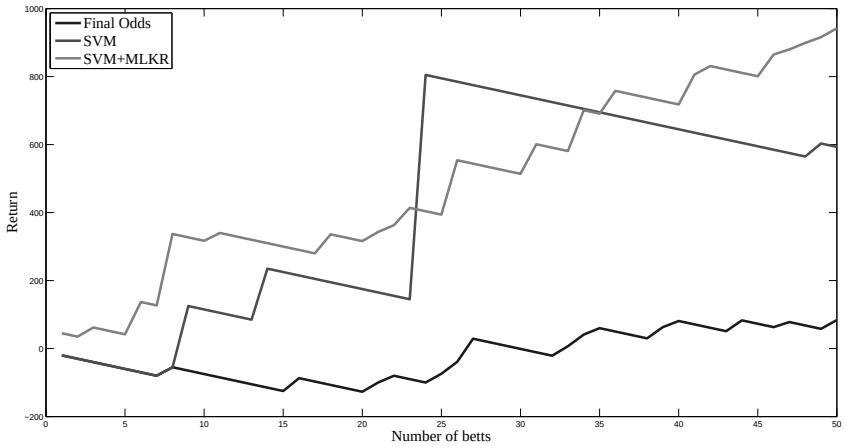


Figure 4: Total return by betting on 1 to 50 horses using different methods

5 Conclusion

We have proposed a flexible stock market prediction framework that allows users to choose different data sources and machine learning techniques to predict stocks. While most prediction approaches are based on Naive Bayes, neural networks or support vector machines, we have illustrated the flexibility of our framework by considering a more adaptive non-parametric learning approach for stock forecasting. In particular, a metric learning based regression method and an HBP-based approach have been proposed and implemented on our framework. As discussed in Section 4.8, our proposed approach (on the proposed framework) has produced better prediction than other approaches by utilizing more than one techniques and data sources in the prediction.

Although we have demonstrated a few techniques (sentiment analysis, feature selection, HBP, SVRML) that are implemented and integrated effectively on our proposed framework, our future work is to implement a large set of techniques and incorporate more data sources. As a result, we can report a more comprehensive study on a large set of learning techniques using multiple financial data sources.

References

Ausín, M. C., Galeano, P. & Ghosh, P. (2014), 'A semiparametric bayesian approach to the analysis of financial time series with applications to value at risk estimation', *European Journal of Operational Research* **232**(2), 350–358.

Benthaus, J. & Beck, R. (2015), It's more about the content than the users! the influence of social broadcasting on stock markets, *in* 'European Conference on Information Systems'.

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

Durante, F., Pappadà, R. & Torelli, N. (2015), 'Clustering of time series via non-parametric tail dependence estimation', *Statistical Papers* **56**(3), 701–721.

Fama, E. F. (1970), 'Efficient capital markets: A review of theory and empirical work*', *The journal of Finance* **25**(2), 383–417.

Ghanavati, M., Wong, R. K., Chen, F., Wang, Y. & Lee, J. (2016), A hierarchical beta process approach for financial time series trend prediction, *in* 'Pacific-Asia Conference on Knowledge Discovery and Data Mining', Springer, pp. 227–237.

Huang, C.-J., Yang, D.-X. & Chuang, Y.-T. (2008), 'Application of wrapper approach and composite classifier to the stock trend prediction', *Expert Systems with Applications* **34**(4), 2870–2878.

Lee, S. S. & Mykland, P. A. (2008), 'Jumps in financial markets: A new nonparametric test and jump dynamics', *Review of Financial studies* **21**(6), 2535–2563.

Li, Q., Jiang, L., Li, P. & Chen, H. (2015), Tensor-based learning for predicting stock movements, *in* 'Twenty-Ninth AAAI Conference on Artificial Intelligence'.

Li, X., Xie, H., Chen, L., Wang, J. & Deng, X. (2014), 'News impact on stock price return via sentiment analysis', *Knowledge-Based Systems* **69**, 14–23.

Li, X., Xie, H., Song, Y., Zhu, S., Li, Q. & Wang, F. L. (2015), 'Does summarization help stock prediction? a news impact analysis', *Intelligent Systems, IEEE* **30**(3), 26–34.

Li, X., Xie, H., Wang, R., Cai, Y., Cao, J., Wang, F., Min, H. & Deng, X. (2016), 'Empirical analysis: stock market prediction via extreme learning machine', *Neural Computing and Applications* **27**(1), 67–78.

Li, Z., Zhang, B., Wang, Y., Chen, F., Taib, R., Whiffin, V. & Wang, Y. (2014), 'Water pipe condition assessment: a hierarchical beta process approach for sparse incident data', *Machine learning* **95**(1), 11–26.

Lin, A., Shang, P., Feng, G. & Zhong, B. (2012), 'Application of empirical mode decomposition combined with k-nearest neighbors approach in financial time series forecasting', *Fluctuation and Noise Letters* **11**(02), 1250018.

Lin, Y., Guo, H. & Hu, J. (2013), An svm-based approach for stock market trend prediction, *in* 'Neural Networks (IJCNN), The 2013 International Joint Conference on', IEEE, pp. 1–7.

Loughran, T. & McDonald, B. (2011), 'When is a liability not a liability? textual analysis, dictionaries, and 10-ks', *The Journal of Finance* **66**(1), 35–65.

Montgomery, D. C., Jennings, C. L. & Kulahci, M. (2015), *Introduction to time series analysis and forecasting*, John Wiley & Sons.

Moskowitz, T. J., Ooi, Y. H. & Pedersen, L. H. (2012), 'Time series momentum', *Journal of Financial Economics* **104**(2), 228–250.

Patel, J., Shah, S., Thakkar, P. & Kotecha, K. (2015), 'Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques', *Expert Systems with Applications* **42**(1), 259–268.

Prema, K., Agarwal, N. M., Krishna, M. & Agarwal, V. (2016), 'Stock market prediction using neuro-genetic model', *Indian Journal of Science and Technology* **8**(35).

Prollochs, N., Feuerriegel, S. & Neumann, D. (2015), Enhancing sentiment analysis of financial news by detecting negation scopes, *in* 'System Sciences (HICSS), 2015 48th Hawaii International Conference on', IEEE, pp. 959–968.

Qian, X.-Y., Liu, Y.-M., Jiang, Z.-Q., Podobnik, B., Zhou, W.-X. & Stanley, H. E. (2015), Detrended partial cross-correlation analysis of two nonstationary time series influenced by common external forces, Technical report, arXiv. org.

Schumaker, R. P., Zhang, Y., Huang, C.-N. & Chen, H. (2012), 'Evaluating sentiment in financial news articles', *Decision Support Systems* **53**(3), 458–464.

Sheta, A. F., Ahmed, S. E. M. & Faris, H. (2015), 'Evolving stock market prediction models using multi-gene symbolic regression genetic programming', *Artificial Intelligence and Machine Learning AIML* **15**, 11–20.

Shumway, R. H. & Stoffer, D. S. (2013), *Time series analysis and its applications*, Springer Science & Business Media.

Smola, A. J. & Schölkopf, B. (2004), 'A tutorial on support vector regression', *Statistics and computing* **14**(3), 199–222.

Thibaux, R. & Jordan, M. I. (2007), Hierarchical beta processes and the indian buffet process, *in* 'International conference on artificial intelligence and statistics', pp. 564–571.

Weinberger, K. Q. & Tesauro, G. (2007), Metric learning for kernel regression, *in* 'International Conference on Artificial Intelligence and Statistics', pp. 612–619.

Xing, H.-j., Ha, M.-h., Hu, B.-g. & Tian, D.-z. (2009), 'Linear feature-weighted support vector machine', *Fuzzy Information and Engineering* **1**(3), 289–305.

Yu, L., Chen, H., Wang, S. & Lai, K. K. (2009), 'Evolving least squares support vector machines for stock market trend mining', *Evolutionary Computation, IEEE Transactions on* **13**(1), 87–102.

Revisiting and Extending the X-of-N Decision Tree Approach for Event Based Time Series Analysis

Chao Sun[1] **David Stirling**[2]

[1] Faculty of Arts and Social Sciences
University of Sydney
Camperdown 2006, Australia
Email: chao.sun@sydney.edu.au

[2] School of Electrical, Computer and Telecommunications Engineering
University of Wollongong
Wollongong NSW 2522, Australia,
Email: david_stirling@uow.edu.au

Abstract

Decision tree algorithms were not traditionally considered for sequential data classification, mostly because feature generation needs to be integrated with the modelling procedure in order to avoid a localisation problem. This paper presents an Event Group Based Classification (EGBC) framework that utilises an X-of-N (XoN) decision tree algorithm to avoid the feature generation issue during the classification on sequential data. In this method, features are generated independently based on the characteristics of the sequential data. Subsequently an XoN decision tree is utilised to select and aggregate useful features from various temporal and other dimensions (as event groups) for optimised classification. This leads the EGBC framework to be adaptive to sequential data of differing dimensions, robust to missing data and accommodating to either numeric or nominal data types. The comparatively improved outcomes from applying this method are demonstrated on two distinct areas – a text based language identification task, as well as a honeybee dance behaviour classification problem.

1 Introduction

Time series data mining (TSDM) is a challenging task which has attracted enormous attention in the recent years. Much of the assessed research focuses on producing appropriate time series (TS) representations in order to retain the order of the data. However, these representations, considered as TS features, are further analysed by a limited number of simpler methods. Some TSDM approaches rely on the indexibility of their features (Shieh & Keogh 2008, Agrawal et al. 1993), and others focus on similarity based features (Möller-Levet et al. 2003, Morrill 1998). As traditional data mining algorithms, such as the intention of decision trees, are generally designed to address static datasets, thus they are rarely employed for analysing TS.

Beyond the level of representations, orders among nominal features can be represented by Allen's Interval Algebra (Allen 1983), in which 13 basic relations between two intervals are defined, or various extensions of Allen's definition (Freksa 1992, Roddick & Mooney 2005). However, with numerous potential ordering types between intervals, the problem becomes significantly more complex in a real time series dataset with tens or hundreds of representative features.

Episode mining (Mannila et al. 1995) is a specialised approach for analysing temporal event data with numerous event combinations that occur within a given window. This technique can be extended to general time series data if nominal TS features are defined as events. The concept of an episode emphasises together with the importance of combinations of various events, that this conforms to the manner in which humans often perceive temporal events (Batyrshin & Sheremetov 2008). Despite this extensive research on episode mining considers the discovery of frequent serial episodes as the fundamental problem (Mannila et al. 1997, Laxman et al. 2007), as these frequent episodes are believed to have a higher importance than infrequent episodes. However, in the real world, this is not always true. For example, a combination that repeatedly occurs may not be interesting to the observer.

A similar situation arises with decision tree modelling. Typically, every node in a tree represents a conditional test out of many others which divides the data into segments with a purer mixture of classes at each descending level. Therefore, the meaningfulness of a condition node is measured by the information gain obtained in dividing the dataset. If all event combinations in an event sequence are labelled, then the meaningfulness of these can be measured in a similar way. In this paper, a new sequential data classification approach is presented where the meaningful event combinations are the key factors. The meaningfulness of a combination is established by an XoN (X-of-N) decision tree (Zheng 2000) rather than by its frequency, therefore this approach is based on information theory. Multiple event groups are constructed and selected in order to form compact XoN features that are used in the tree model.

An XoN representation contains special features that cover multiple possible combinations of given conditions. It is formed by two parts, X and N, and each is a non-empty unsorted set. The N component consists of several traditional conditions (TS events in this case), and the X component is a list of non-negative integers, denoting the exact numbers of conditions in N to be satisfied. A typical XoN feature

can be decomposed into a number of combinations of various unordered events, as seen in Figure 1. The simple XoN representation covers seven different situations, where the A, B, C and D denote conditions that may occur concurrently in the dataset. Although the XoN algorithm was not developed for time series analysis, we understand there are similarities between the decomposed event combinations and episodes. As an integrated feature, an XoN node practically gathers a series of episodes and combines their classifying capacities in order to obtain a better performance.

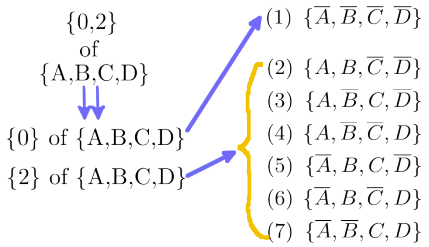


Figure 1: A full expansion of an XoN feature.

In the remainder of this paper, Section 2 focuses on the methodology and experimental settings. Section 3 describes the experimental results by comparing outcomes with other existing techniques. Lastly, in Section 4 we discuss the advantages of this approach and the possibility to extend it as a universal time series data mining frame work for more complex datasets.

2 Methodology

The method is designed to process sequential data with either numeric or nominal values, and is based on an idea that any real-world event can be observed and identified as a combination of other observable "events" over various temporal attributes. The word "event" here is defined as any distinguishable feature derived locally from a sequential data, such that, it may refer to temporal shapes, different value zones, transformations of the real data, symbol segments or any other features that reflect the characteristics of any temporal segments.

Whilst not all contextual features are tied to targeted real-world events, some feature combinations across various attributes however, may be exclusively associated with such an event. This is similar to the manner in which a doctor may diagnose patients: one symptom alone may not be sufficient evidence of a certain disease, however a set of relevant symptoms could align the diagnosis to a specific disease with considerable confidence. In this paper an EGBC framework, the "events" hold an equivalent meaning to the "symptoms" in a medical diagnosing system.

The basic assumption can be described as follows: If a (detectable) process occurs at some time point T, its causal factors or resultant responses can be represented as time series features and in turn transformed into nominal events that occur around the time point T. The novelty of this work is that all casual and consequent events within a given time frame are indiscriminately analysed as unsorted combinations, and the type of the target process can be classified based on these combinations.

Therefore, in situations where specific classes are clearly labelled on each stage of such sequences, the goal is to build a model that classifies these segments based on certain important event combinations they contain. Most time series data are not originally presented in the form of nominal events, thus an appropriate transformation is required to represent the time series in a nominal event form. Event groups are extracted with a sliding window over the sequence, and meaningful combinations of these are automatically constructed and selected by an adapted XoN algorithm.

The flow chart in Figure 2 briefly explains the overall procedure with a simplified example. Assuming there are three different working stages in a process, and the task is to identify these stages (classes) from two associated time series TS-A and TS-B. The numeric time series are then transformed into two event sequences respectively in which synchronisation is not compulsory, named Seq-A and Seq-B, and events are labelled as An and Bn accordingly (n being an event numbers). Event groups are then formed across both sequences by selecting all events that fall within a sliding window, and labelled by the classes as the training dataset. The XoN decision tree model is then trained in order to find the optimised event combinations for classifying the various process stages. During this procedure, an understanding of the expected event combinations and their correct labelling are essential for verifying whether this approach is effectively finding useful combinations and classifying the data accurately.

In order to evaluate the feasibility of our approach, we employed our event-group based approach for two different sequential data classification tasks in the following experiment section, described as follows:

1. Language identification. The goal of this task is to recognise the languages from three articles written in different languages. Our approach is used to identify what language a word is written in without any dictionary. This task demonstrates how our approach performs on real event based sequences (non-time series) rather than artificially generated data. The outcomes are compared with the text mining algorithm – "TextCat" (Hornik et al. 2013).

2. Honeybee Dance Behaviour Recognition. In order to identify the dancing behaviours of a honey bee, genuine honeybee motion data are studied by employing our event group based approach. The sequential real-valued bee motion data are transformed and clustered into events before being analysed by the XoN decision trees. This task extends the online classification work to a real world scenario, involving multi-dimensional time series data.

3 Experiment and Results

In this section, the method proposed in Section 2 will be applied on two sequential classification tasks. A language identification (LID task) is conducted in order to model and distinguish three different languages (English, Italian and Dutch) without the benefit of any prior linguistic or dictionary based knowledge. With naturally occurring events and groups within the text, i.e. letters and words, the aim of this task is to verify the feasibility of our event group based methodology. In the second, the honey bee dancing task, we evaluate the method on real-valued time series. In both tasks, online classification along a continuous sequences is accomplished, i.e. different classes in different states, and in a progressive manner in terms of the complexity.

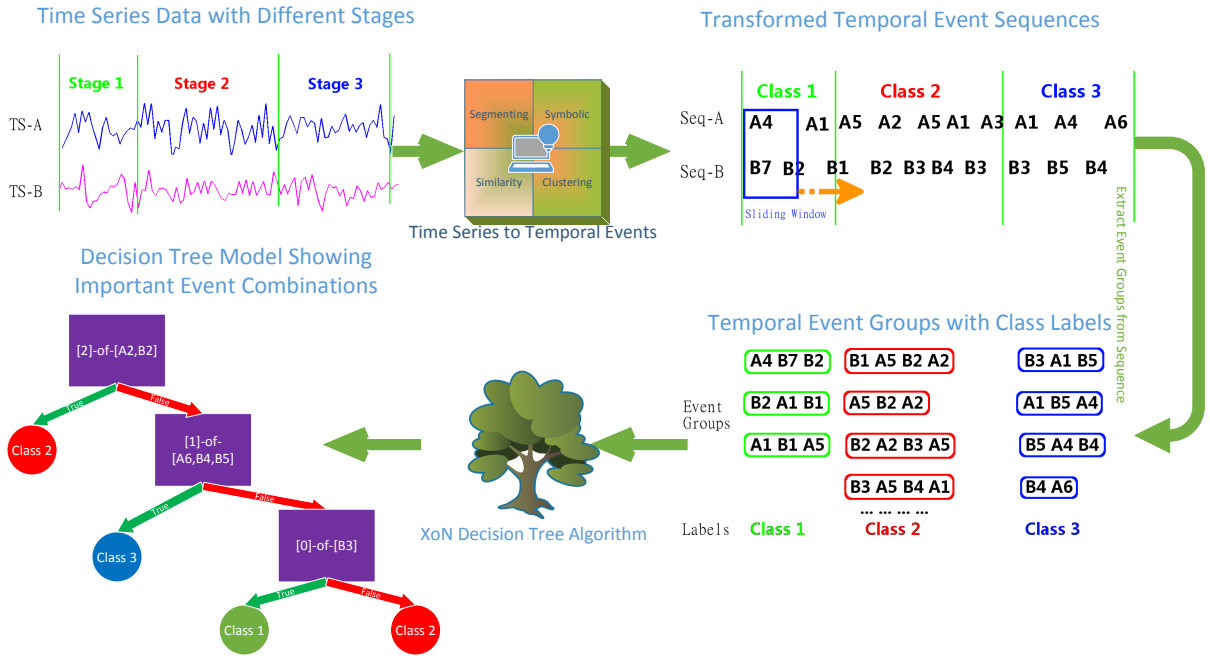


Figure 2: The Procedure of Event Group Based TS Classification Approach.

3.1 Text Mining – Language Identification

The automatic identification of a language from text is an important and well studied problem, which was considered as a solved problem (McNamee 2005). LID is a typical multi-class classification task, however, in the past, decision tree techniques were not considered in this domain. The general LID methods are based on either statistical language modelling or the frequency of common word usage. Both the statistical and frequency of common words usage methods work better on sentences that consist of more than 15 words (McNamee 2005). The accuracy of various traditional LID methods approaches some 98% if the decision is based on sentences or short paragraphs (Takcı & Soğukpınar 2004).

For the general LID task, identifying the language of single words is not practically necessary, because it is not common to mix different languages within individual sentences and paragraphs. However, in this work, in order to evaluate the proposed event group based classification approach, all articles are analysed and classified word by word.

3.1.1 Dataset

Six public domain e-books in the three specific languages are obtained from the Project Gutenberg website (*Project Gutenberg.* n.d.), two in each language[1]. In order to ensure that only letter combinations are used as a valid basis for the classification process, all symbols and non-English characters are removed. Words with three or fewer letters are ignored and the texts are pre-processed with the Porter Stemming algorithm (Porter 1980) to reduce the word form variances for better information retrieval. The sizes of the training and testing datasets are 118,192 words (Eng: 31,387; Ita: 50,282; Dut: 36,523) and 76,143 words (Eng: 25,644; Ita: 34,068; Dut: 16,431) respectively.

Because words are the basic meaningful elements in languages, the event groups are naturally defined as single words. As the spelling is sensitive to the order of characters, in order to partially retain ordering information, every pair of adjacent characters in a word are defined as events rather than individual letters. Preliminary experiments indicate that the XoN models using 2-letter events have a noticeable improvement (5%) on classification accuracy compared to models using 1-letter events, if the same modelling parameters are used.

3.1.2 Experiment Setting and Results

Eight XoN models were generated based on a random 60% of the training text, which all performed similarly in terms of the amount of pruning used and their accuracy. In the following experiments, the model with the highest classification accuracy and moderate pruning was chosen for further analysis.

In this experiment, the XoN tree model is not forced to make a decision on every event group (word), as some words may exist in different languages, and these cross-class groups are not classifiable. Also, the unordered event groups increase the difficulty for language identification on individual words. For the aim of either, classifying languages or obtaining linguistic knowledge, the classification of every word is not necessary. In fact, ignoring some common words is seen to help the model to focus on the more important linguistic characteristics.

In order to avoid vague decisions, in this experiment, a minimum confidence parameter is used to control the output behaviour of the XoN model. When the confidence from a decision is lower than a given threshold "minConf", the classifier output is altered to "unknown" rather than the most likely class. Introducing this "unknown" class and a minimum confidence reduces the number of overall misclassifications and increases the overall classification

[1] Training Texts:
English: http://www.gutenberg.org/files/1999/1999.txt
Italian: http://www.gutenberg.org/cache/epub/1012/pg1012.txt
Dutch: http://www.gutenberg.org/files/18066/18066-8.txt
Testing Texts:
English: http://www.gutenberg.org/ebooks/43230
Italian: http://www.gutenberg.org/ebooks/43226
Dutch: http://www.gutenberg.org/ebooks/11500

accuracy. Figure 3 illustrates how the accuracies and unknown ratios (the number of unknown cases divided by the size of the test dataset) are affected when the model is being pruned with various minCase and minConf.

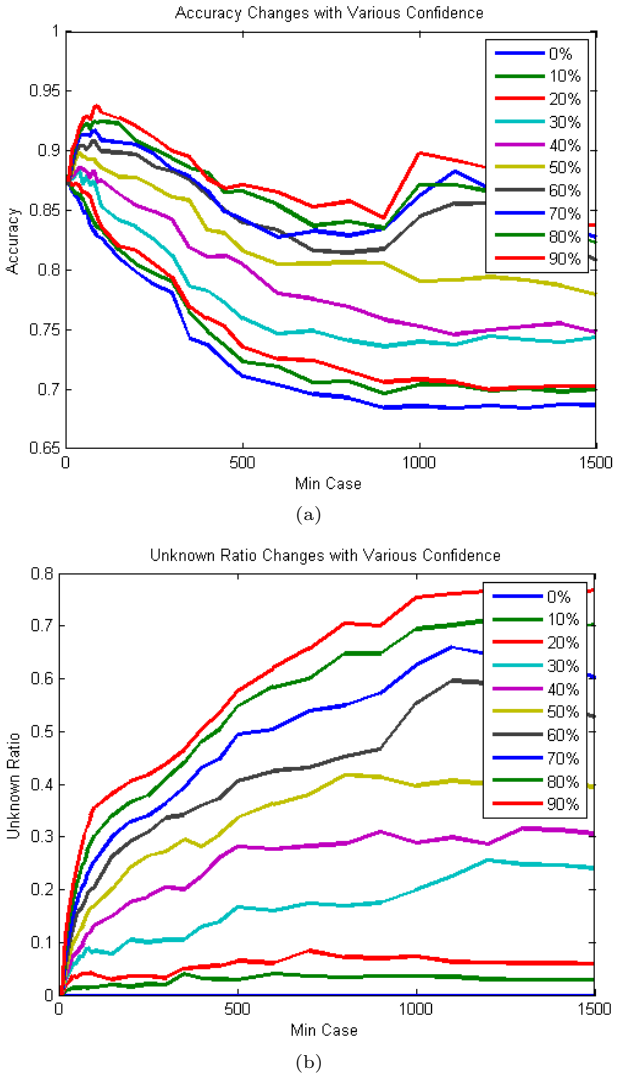


Figure 3: Classification Performance on Models with Different minCase and minConf. a) Accuracy; b) Unknown Ratio.

In Figure 3, as the minimum confidence varies from 0% to 90%, the classification accuracy on the testing text increases. However, a high confidence requirement also renders the model as ineffective when the tree is heavily pruned, and the unknown ratio becomes too high. It is reasonable to assume that an optimal model could be expected to maintain a relatively high accuracy whilst keeping the unknown ratio below an acceptable level.

The yellow line (minConf = 50%) in Figure 3(a) reaches its peak (accuracy = 89.6%) when the value of minCase is 40, and the unknown ratio for the same model is 9.8% when classifying the testing dataset. Considering the highest possible accuracy model has 93.79% accuracy and a 34.09% unknown ratio, the yellow line model is considered to be an optimised tree model for the task. Note that the training and testing texts are sourced from different e-books with different authors, styles and expressions. It is also expected that this approach would also improve if the training and testing data were sourced from the same article or author context.

3.1.3 Performance Comparison

Text-Cat (van Noord n.d.), a LID implementation of the N-gram-based text categorization (Cavnar et al. 1994), is applied on the testing dataset as a comparison. The Text-Cat is limited to the selected three languages in this experiment although it supports up to 69 different languages. Text-Cat may make multiple decisions on a word, and these are post-processed to either "unknown" or a false decision. For example, if an English word is identified as "English or Italian", then the result is converted to an "unknown"; and if an English word is identified as "Italian or Dutch", because both are incorrect, the first incorrect decision will be kept.

The Text-Cat models used in this work are publicly available (van Noord n.d.), and each language model file contains 400 high frequency terms. A comparison between Text-Cat and the XoN tree models (with three levels of pruning) can be viewed in Table 1. The first XoN decision tree model is lightly pruned, containing some 1222 nodes and achieves 89.6% word-by-word classification accuracy on the testing texts (9.8% unknown words of the testing texts). As a comparison, the Text-Cat has 81.7% in accuracy on the same testing text (32.4% unknown words of the testing texts). Confusion matrices expressed in percentages, are also included in Table 1 for easy comparison between all models.

One advantage of the XoN decision tree model is that it can be easily pruned to suit different requirements. For instance, if the accuracy is of higher priority than the unknown ratio, the model can be pruned as Model 2 in Table 1, which provides a similar unknown ratio to what Text-Cat provides, but with a significantly higher accuracy at 91.9%. However, if a smaller, less complex form is preferred, the XoN model can also be further pruned for less nodes. With a minCase=460 and minConf=50%, such as Model 3 which contains only 292 nodes yet also produces a similar unknown ratio (32.6%) compared to Text-Cat. This model still provides a marginally higher identification accuracy at 83% on the test dataset.

The actual XoN tree models in Table 1 are too large to be included in this paper. Figure 4 illustrates a number of the classification results where three sections of texts and their classification results are visually presented. In Figure 4, the English, Italian and Dutch are printed in blue, red and green fonts. If a word is mis-classified, the background colour of that word changes to the colour of the wrong decision accordingly. Further, when an unknown identification is made, the word is presented in black.

3.2 Honeybee Dancing Behaviour Classification

In this task, classification on genuine, real-valued time series data is involved. The honey bee dancing dataset (Oh et al. 2008) includes six videos of honeybee dancing, where the trajectory of a signalling bee is automatically tracked and converted into quantitative sequential motion data, including sequences of position (X and Y) and the head angle of the bee. This dataset contains ground truth labels of all data records according to which behaviour of three possible behavioural patterns it contains: waggle, right-turn or left-turn. A number of researchers have previously attempted to classify the type of dancing behaviour based on these motion features, using extended HMM, segmentation (Fox et al. 2008) and

Table 1: Identification on Testing Texts with XoN and Text-Cat Models

Model	*XoN Model 1* minCase=40 minConf=50%	*XoN Model 2* minCase=40 minConf=98%	*XoN Model 3* minCase=460 minConf=50%	*Cat Model* Original
Size	1222 Nodes	1342 Nodes	292 Nodes	1200 Terms
Accuracy	89.6%	91.9%	83%	81.7%
Unknown	9.9%	25%	32.6%	32.4%
Confusion Matrix (%)	*Pred* → / *Eng* 88.9 7.3 3.8 / *Ita* 6.5 90.9 2.6 / *Dut* 6.6 5.0 88.4	*Pred* → / *Eng* 90.1 6.5 3.4 / *Ita* 5.0 93.5 1.5 / *Dut* 5.5 3.8 90.7	*Pred* → / *Eng* 70.4 23.4 6.2 / *Ita* 6.6 89.6 3.8 / *Dut* 9.3 5.9 84.8	*Pred* → / *Eng* 60.1 19.4 20.5 / *Ita* 6.9 89.1 4.0 / *Dut* 6.1 1.8 92.1

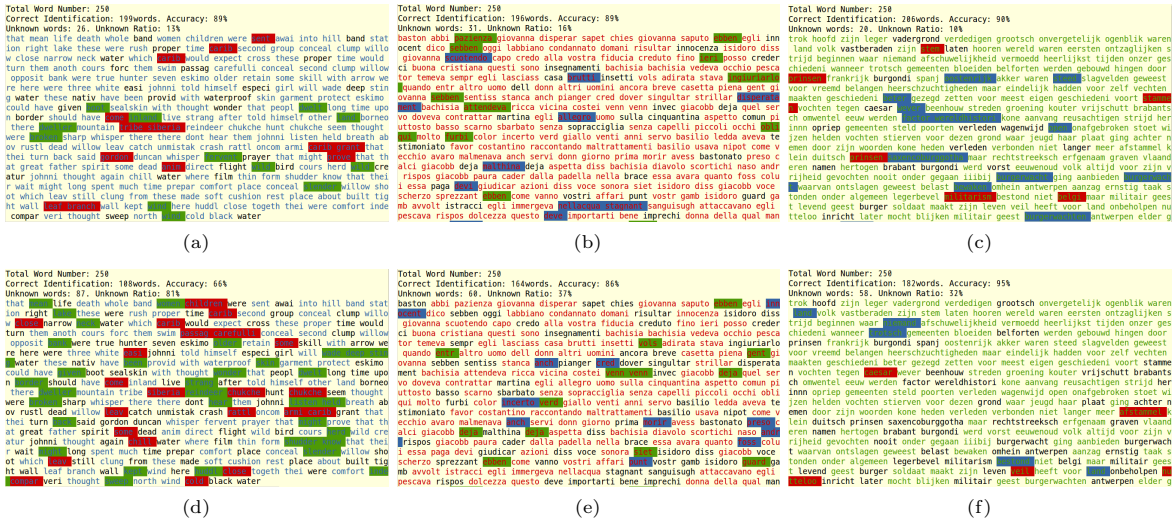


Figure 4: Word-by-Word Language Identification by XoN and Text-Cat Models. Font colours indicate True Class, Background colours indicate Identified Class. Meaning of Colours: Blue-English, Red-Italian, Green-Dutch, Black-Unknown. (a,b,c) Results from XoN Model; (d,e,f) Results from Text-Cat Model.

clustering methods (Zhou et al. 2013). In this section, the same classification task is performed to show how our event based decision tree method performs on this same dataset.

3.2.1 Data Pre-processing

In order to transform the real-valued motion data into nominal events, we define patterns of movements and consequential movements at any given time point T as the event for describing the dancing behaviours. For example, it is safe to assume when the behaviour is left-turn, the bee's position moves towards the left relative to the previous trajectory. For the waggle behaviour, the bee may move both left and right in an alternating manner. Therefore, the basic movement patterns for this task are closely associated with the directions of movements. The sequences of raw positions and head angles are transformed into the following six features:

- *X-Off*: The X offset of current position relative to the previous frame.
- *Y-Off*: The Y offset of current position relative to the previous frame.
- *CosAng*: Cosine of the head angle.
- *SinAng*: Sine of the head angle.
- *AngDiff*: The difference between head angle and previous trajectory.
- *Dist*: The distance the bee travels since last frame.

In the above list, the *CosAng* and *SinAng* features can be considered as an estimation of the *X-Off* and *Y-Off* for the following moment, if the bee maintains its current heading direction and speed. The *AngDiff* indicates the difference between the current heading and that of the previous movement. From every frame, a vector of all six features is viewed as an unique motion status of the dancing bee, and every status contains information derived from both the current and previous frame.

If every such motion status is treated as a nominal event, there will be a massive number of events. The curse of dimensionality will affect our decision tree algorithm that would seek to find the optimised combinations. The feature vectors are therefore pre-clustered using a Minimum Message Length (MML) algorithm(Wallace & Boulton 1968). Subsequently, each cluster is treated as an event representing a series of similar motion states. The MML clustering also reduces the six dimensional feature sequences into a single dimensional event sequence with a limited number of events.

3.2.2 Parameter details

Unlike the textual LID event sequences, the honeybee dancing events have uniform time intervals, therefore a fixed number of events are selected with a fixed-width sliding window. The behavioural label of an event group is determined by the majority of the ground truth within that window. In order to obtain an optimised model reflecting the inherent truth within the honey bee dancing motion data, a series of training processes were executed with various

parameters, such as different initialisations, training datasets, numbers of MML clusters, sliding window sizes and the minimum case numbers required before splitting a node.

The parameters used in this experiment are listed below:

- Number of MML clusters: This parameter was pre-set to be 8, 10 or 12 clusters.
- Sliding window size (WinSize): 5,7,9,11. These are also the sizes of event groups, i.e. the number of continuous video frames used for classifying a behaviour. Only an odd number of events are used for simple majority labelling.
- XoN seed: 100, 200, 300, 400. This seed controls all random functions in the XoN training procedure, such as initialisation, dataset division etc.
- Training ratio: 70% of the sequential data are used for training process and the rest are for testing purpose. Because the continuity is important in the time series data, the testing data is always sourced from a continuous section of the sequence.
- Maximum X (Max-X): Varies from the number four (4) up to the number (WinSize-1). Because the event group size is WinSize, any X greater than WinSize is meaningless, therefore the maximum integer in the X part is limited by the number (WinSize -1).
- Minimum case number for splitting a node (MinCase): 2,5,8,10,15,20,40,70. These are traditional parameter values used in many decision trees, and indicates the minimum size of data before a node can be further split, to extend the model.

3.2.3 Model Selection

The training process of the XoN algorithm exhaustively uses all parameters listed in previous section, and the models are evaluated on all three datasets (training, testing and whole) for error rates.

The classification error rates are calculated through two techniques: a raw prediction error rate and a sliding prediction error rate. The raw error rate is the case by case error rate showing the accuracy of event group classification. However, because each frame of data is contained in multiple event groups, the classification on a single frame should also be further determined using the majority of classifications it receives as the sliding window passes. The sliding error rate in general is about 5-10% better than the raw prediction error rate, thus all error rates or accuracies in the rest of this section are based on the sliding method.

In most cases, a fully grown decision tree model would have higher classification accuracy on the training data than on the testing data, this is of course a symptom of over training as the model performs worse on previously unseen data. Pruning with larger values of MinCase normally reduces this performance difference, however it also lowers the overall accuracy. In order to select an optimised tree model from all the trained models, the models are ascendantly sorted based on a $score = AllErr + abs(TrainErr - TestErr)$. A good model is expected to have a low score indicating that both the overall error rate and the performance difference between testing and training datasets are low.

Table 2: Selected models and sliding prediction accuracy

ID	cNo	Seed	winSize	Max-X	MinCase	slideAccu	ErrDiff
1	12	400	11	5	15	85.3%	0.4%
2	8	300	11	8	5	93.0%	0.7%
3	8	100	9	7	20	87.04%	9.5%
4	14	400	11	5	10	90.62%	7.7%
5	16	200	11	8	15	88.92%	11.5%
6	14	200	9	5	15	88.17%	12.1%

3.2.4 Experimental Results

The exhaustive training process selects the following model parameters for classifying each honeybee dancing data as shown in Table 2.

Table 2 illustrates that the best fully grown XoN tree models discovered during the exhaustive training process. Due to the limited number of models that were generated, these are not reflective of the best performance our approach could achieve. Even though, the performance of these models are comparable and even exceed several of other techniques which were employed for the same task.

Zhou et al. (Zhou et al. 2013) summarised a table to compare some state of the art techniques for classifying the honeybee dancing dataset, and the accuracies are further compared with our method in Figure 5.

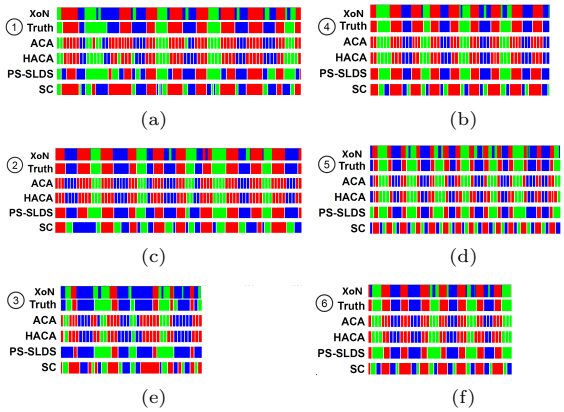


Figure 5: Comparison with classification result from other researches; where the motions are waggle (green), right-turn (red), left-turn (blue).

3.2.5 Discussion

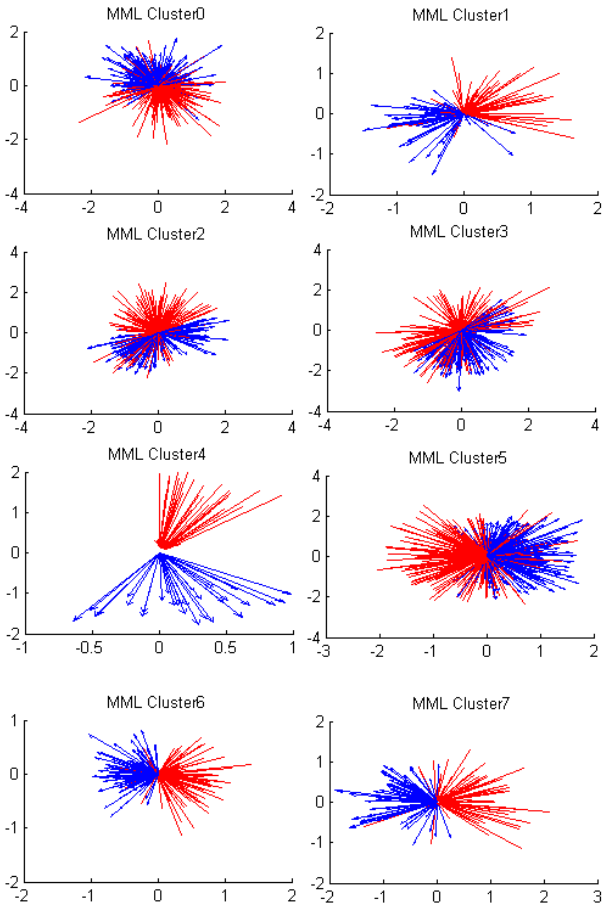
The accuracy comparison in Table 3 indicates that selected event group based decision tree models have comparable classification performance to other existing techniques. Our method outperforms all other techniques in three of the six sequences, and it also has the highest average accuracy in all of the compared methods.

Figure 6(a) illustrates the MML cluster preprocessing utilised in the No.2 honeybee dancing dataset model as an example. Figure 6(a) illustrates the common bee motion states as eight MML clusters, where the red arrows represent the movements from the previous frame, contrastingly, the blue arrows represent the current head direction. Figure 6(b) also includes histograms that indicate how the signatures or profiles of these eight MML cluster (Events) associate with the three behaviour categories.

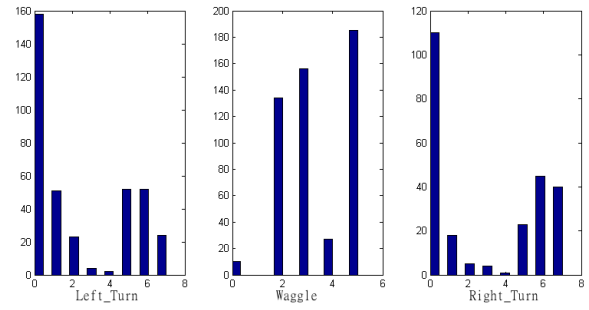
A number of key facets can be understood in Figure 6(b): Firstly, the MML cluster events are a biased distribution across the three behaviours, which

Table 3: Classification Accuracy Comparison with Other Techniques, all numbers in percentage. Bold fonts stand for the best accuracy among methods. The last row includes the accuracies of the XoN-MML approach.

Category	Algorithm	Seq1	Seq2	Seq3	Seq4	Seq5	Seq6	Average
Unsupervised	ACA	84.5	92.5	60.0	92.2	87.8	**92.8**	85.0
	HACA	85.3	90.0	62.9	91.7	84.5	87.8	83.7
	SC	69.8	63.1	50.9	67.1	57.7	64.9	62.3
	HDP-VAR(I)	46.5	44.1	45.6	83.2	**93.2**	88.6	66.9
Weakly Supervised	HDP-VAR(II)	65.9	88.5	79.2	86.9	92.3	89.1	83.7
Supervised	HDP-SLDS	74.0	86.1	81.3	**93.4**	90.2	90.4	85.9
	PS-SLDS	75.9	92.4	83.1	**93.4**	90.4	91.0	87.7
	XoN-MML	**85.3**	**93.0**	**87.0**	90.6	88.9	88.2	**88.8**



(a)



(b)

Figure 6: (a) Presentation of MML clusters and (b) the Histogram of clusters on each behaviour

indicates the possibility to identify the honeybee behaviours based on these MML events. Secondly, the distribution histograms of "left-turn" and "right-turn" are similar to each other, therefore judging the type of behaviour with a single cluster/event is not feasible. For example, when new motion data is identified as cluster 0, it has little chance of it being a "Waggle", however the chance for being either a "Left-turn" or "Right-turn" are both very high. Considering that the final XoN model provides an accuracy of some 93% on classifying Sequence-2, this provides evidence of the effectiveness of this extended decision tree algorithm.

4 Conclusion and Future Work

This paper presents a novel method to utilise an XoN decision tree technique for classifying generic streams of data sequences. Provided that numeric time series data can always be transformed into a sequence of nominal events with an appropriate abstraction method, the XoN approach can be readily expanded to other time series data mining areas. The language identification test indicates that the XoN based approach has similar or better performance than the traditional LID algorithm, especially when classification is made on isolated words. The success on the honeybee behaviour classification demonstrates that our approach has the capacity to model and classify genuine real-valued time series data, with appropriate transformation of temporal events, and its stable performance which is significant when compared with other techniques.

There are a number of advantages for the methodology presented in this paper. Firstly, the method looks for causally related signs for the classification, which naturally allows time variances between various attributes of the input data. Secondly, the method is based on an abstracted layer (beyond numeric values), the differences in attributes' physical meanings are ignored on this layer. Thirdly, temporal information and local ordering of a single attribute can be embedded into individual events, depending on the event transformation, and the sliding window ensures decision are made over a controllable period. However most importantly, unordered event combination allows temporal variance within the period, which simplifies the feature space and provides tolerance to temporal mis-alignment among attributes.

It has been illustrated in this work that the event group based decision tree approach demonstrates significant capability for classifying the sequential data. This event group based method now has the potential to be used as a generic methodology for modelling more complex time series data with missing or mis-aligned variates. However, users need to obtain

considerable knowledge of the target problem in order to appropriately select the techniques for event generation, and the parameters for the EGBC modelling procedure, e.g. the size of sliding window, duration of events and size of the XoN nodes.

We also find the event groups generated by XoN have certain similarities to the concept of topic modelling (Steyvers & Griffiths 2007), however these are derived from supervised learning based on the information gain rather than probabilities. Future research will include utilising topic modelling for minimising or scaling down the feature space, in order to extend this method to more challenging real world time series data mining with multi-dimensional forms and potentially complicated intra-sequential relationships.

References

Agrawal, R., Faloutsos, C. & Swami, A. N. (1993), Efficient similarity search in sequence databases, *in* 'FODO', pp. 69–84.

Allen, J. F. (1983), 'Maintaining knowledge about temporal intervals', *Commun. ACM* **26**(11), 832–843.

Batyrshin, I. Z. & Sheremetov, L. (2008), 'Perception-based approach to time series data mining', *Appl. Soft Comput.* **8**(3), 1211–1221.

Cavnar, W. B., Trenkle, J. M. et al. (1994), 'N-gram-based text categorization', *Ann Arbor MI* **48113**(2), 161–175.

Fox, E. B., Sudderth, E. B., Jordan, M. I. & Willsky, A. S. (2008), Nonparametric bayesian learning of switching linear dynamical systems, *in* 'NIPS', pp. 457–464.

Freksa, C. (1992), 'Temporal reasoning based on semi-intervals', *Artif. Intell.* **54**(1), 199–227.

Hornik, K., Mair, P., Rauch, J., Geiger, W., Buchta, C. & Feinerer, I. (2013), 'The textcat package for n-gram based text categorization in R', *Journal of Statistical Software* **52**(6), 1–17.

Laxman, S., Sastry, P. & Unnikrishnan, K. (2007), A fast algorithm for finding frequent episodes in event streams, *in* 'Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 410–419.

Mannila, H., Toivonen, H. & Verkamo, A. I. (1995), Discovering frequent episodes in sequences extended abstract, *in* '1st Conference on Knowledge Discovery and Data Mining, Montreal, CA'.

Mannila, H., Toivonen, H. & Verkamo, A. I. (1997), 'Discovery of frequent episodes in event sequences', *Data Mining and Knowledge Discovery* **1**(3), 259–289.

McNamee, P. (2005), 'Language identification: a solved problem suitable for undergraduate instruction', *Journal of Computing Sciences in Colleges* **20**(3), 94–101.

Möller-Levet, C. S., Klawonn, F., Cho, K.-H. & Wolkenhauer, O. (2003), Fuzzy clustering of short time-series and unevenly distributed sampling points, *in* 'IDA', pp. 330–340.

Morrill, J. P. (1998), 'Distributed recognition of patterns in time series data', *Commun. ACM* **41**(5), 45–51.

Oh, S. M., Rehg, J. M., Balch, T. R. & Dellaert, F. (2008), 'Learning and inferring motion patterns using parametric segmental switching linear dynamic systems', *International Journal of Computer Vision* **77**(1-3), 103–124.

Porter, M. F. (1980), 'An algorithm for suffix stripping', *Program: electronic library and information systems* **14**(3), 130–137.

Project Gutenberg. (n.d.), http://www.gutenberg.org. (Accessed on 15/Aug/2013).

Roddick, J. F. & Mooney, C. (2005), 'Linear temporal sequences and their interpretation using midpoint relationships', *IEEE Trans. Knowl. Data Eng.* **17**(1), 133–135.

Shieh, J. & Keogh, E. J. (2008), *i*sax: indexing and mining terabyte sized time series, *in* 'KDD', pp. 623–631.

Steyvers, M. & Griffiths, T. (2007), 'Probabilistic topic models', *Handbook of latent semantic analysis* **427**(7), 424–440.

Takcı, H. & Soğukpınar, İ. (2004), Centroid-based language identification using letter feature set, *in* 'Computational Linguistics and Intelligent Text Processing', Springer, pp. 640–648.

van Noord, G. (n.d.), 'Textcat', http://odur.let.rug.nl/ vannoord/TextCat/index.html. (accessed on 10/Aug/2015).

Wallace, C. S. & Boulton, D. M. (1968), 'An information measure for classification', *The Computer Journal* **11**(2), 185–194.

Zheng, Z. (2000), 'Constructing x-of-n attributes for decision tree learning', *Machine Learning* **40**(1), 35–75.

Zhou, F., la Torre, F. D. & Hodgins, J. K. (2013), 'Hierarchical aligned cluster analysis for temporal clustering of human motion', *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(3), 582–596.

Generating Synthetic Datasets for Experimental Validation of Fraud Detection

Ikram Ul Haq, Iqbal Gondal, Peter Vamplew, Robert Layton
ICSL, Faculty of Science and Technology, Federation University, Australia
PO Box 663, Ballarat 3353, Victoria
ikramulhaq@students.federation.edu.au, {iqbal.gondal, p.vamplew}@federation.edu.au,
robertlayton@gmail.com

Abstract
Frauds are dramatically increasing every year, resulting in billions of dollars in losses around the globe mainly to banks. One of the key limitations in advancing research in the area of fraud detection is the unwillingness of banks to share statistics and datasets about this fraud to the public due to privacy concerns. To overcome these shortcomings, in this paper an innovative technique to generate highly correlated rule based uniformly distributed synthetic data (HCRUD) has been presented. This technique allows the generation of synthetic datasets of any size replicating the characteristics of the limited available actual fraud data, thereby supporting further research in fraud detection. The technique uses reference data to produce its characteristic measures in terms of Ripple Down Rules (RDR) ruleset, classification and probability distribution to generate synthetic data having same characteristics as reference data. In the generated data, we have ensured that the distribution of individual and the combination of correlated attributes is maintained as per reference data. Further, the similarity of generated data with reference data is validated in terms of classification accuracy using four well-known classification techniques (C4.5, RDR, Naïve Bayes and RandomForest). Instance-based learning classification techniques were used to validate the classification accuracy further as instance-based learners classify the instances to the nearest neighbour instances using similarity functions. The empirical evaluation shows that the generated data preserves a high level of accuracy and data distributions of single attributes as well as the combination of attributes and is very similar to the original reference data.

Keywords: Synthetic Data Generation, Fraud Analysis, Classification, Rule based, Uniformly distributed, Ripple Down Rules

1 Introduction
Online banking frauds are resulting in billions of dollars losses to the banks around the world. In 2008, Phishing related Internet banking frauds costed banks more than US$3 billion globally (McCombie, 2008). Microsoft Computing Safety Index (MCSI) survey (2014) has highlighted that the annual worldwide impact of phishing and various forms of identity theft could be as high as US$5 billion and the cost of repairing damage to peoples' online reputation is much higher at around US$6 billion, or an estimated average of US$632 per loss (Marican & Lim, 2014). Fraud detection for online banking is a very important research area but there are a number of challenges facing research on this topic. In particular knowledge on banks' fraud detection mechanism is very limited and banks do not publish statistics of the fraud detection systems (Maruatona, 2013). Most of the security is provided by third party IT-companies who also protect their intellectual property from their competitors. So both banks and IT security companies do not publish information on their security systems. Bolton & Hand (Bolton & Hand, 2002) also highlight that development of new fraud detection methods is difficult because the exchange of ideas in fraud detection is very limited but authors also support the notion that fraud detection techniques should not be described with details publically; otherwise criminals may also access that information.

To conduct innovative research in fraud analysis, a large amount of data is required. Banks do provide data in some cases, but the data is normally either in small volume or may not provide specific features which are needed to verify new research techniques and algorithms. With the consideration of these limitations, a viable alternative is to generate synthetic data. This paper presents an innovative technique for generating simulated online banking transaction data and evaluates how well this simulated data matches the original, small set of reference data. Further, paper presents fraud detection study on the synthetic data.

Synthetic data can be used in several areas and benefits of synthetic data is well presented by (Bergmann, n.d.):

- It allows controlling the data distributions used for testing. So the behaviour of the algorithms under different conditions can be studied.
- It can help in performance comparison among the different algorithms regarding the scalability of the algorithms.
- It creates instances having the finest level of granularity in each attribute, whereas in publicly

available datasets anonymization procedures are applied due to privacy constraints.

2 Related Work

The idea of synthetic data generation is not new, as in 1993, Donald B. Rubin generated data to synthesize the Decennial Census long form responses for the short form households (Rubin, 1993). However, it has not been applied to the area of online banking fraud.

Various attempts have been made to generate synthetic datasets. One technique uses uni-modal cluster interpolation e.g. Singular value decomposition interpolation (SVDI) (Coyle, et al., 2013). This technique presents a method that uses data clusters at certain operating conditions where data is collected to estimate the data clusters at other operating conditions, thus enabling classification. This approach is applied to the empirical data involving vibration-based terrain classification for an autonomous robot using a feature vector having 300 dimensions, to show that these estimated data clusters are more effective for classification purposes than known data clusters that correspond to different operating conditions. SVDI's main shortcoming is that the estimates of data clusters and known data clusters have the same number of samples.

Different frameworks to synthesise the data (Anon., 2015), (Bergmann, n.d.), (Anon., 2015), (Maj, 2015) have been studied but all of these frameworks neither classify the data nor are based on any existing datasets. One attempt was to generate synthetic census based micro-data with customization and using extensibility of an open-source Java based system (Ayala-Rivera, et al., 2013). In data generation process, they used probability weights by capturing frequency distributions of multiple attributes. Due to attribute interdependency, they also applied attributes constraints, but they have not applied the weightage on the combination of attributes. It might be possible that distribution of individual attributes is same in the generated data as in the reference, but this distribution cannot be guaranteed for the combination of the attributes. The generated data, cannot be used in the domain of classification problems, as this is not the classified data. Another attempt was made to generate constraint-based automatic test data. The technique is based on mutation analysis and creates test data that approximates relative adequacy (DeMilli & Offutt, 1991). This technique is used to generate test data for unit and module testing. This paper does not mention whether this technique is also applicable to produce data for classification.

Chawla et al present synthetic minority over-sampling technique, which is based on the construction of classifiers from imbalanced datasets. Under-sampling of the majority (normal) class has been proposed as a good means of increasing the sensitivity of a classifier to the minority class. Their method of over-sampling the minority class involves creating synthetic minority class examples (Chawla, et al., 2002). This approach is ideal for imbalanced data where the requirement is to reduce majority class and increase the minority class. This technique is not ideal to increase overall data size.

In another paper (Yoo & Harman, 2012) have proposed a technique to generate additional test data from existing reference data. Their paper highlights that mostly existing automated test data generation techniques tend to start from scratch, implicitly assuming that no pre-existing test data is available. However, this assumption may not always hold, and where it does not, there may be a missed opportunity; perhaps the pre-existing test cases could be used to assist the automated generation of additional test cases. They have used search-based test data regeneration technique; that can generate additional test data from existing test data using a meta-heuristic search algorithm (Yoo & Harman, 2012). But the generated data, cannot be utilized in the domain of classification problems, as it does not have classification labels.

Another synthetic data generation and corruption technique is by Christen & Vatsalan (Christen & Vatsalan, 2013), which generates data based on real data having the capability to produce data for Unicode character sets as well. This technique also caters attribute distribution and dependency. Besides these features, this technique is lacking labelled data and attribute distribution multiple attributes. One novel technique is to generate synthetic data for electronic medical records proposed by Buczak et al (Buczak, et al., 2010). However, this technique can generate data mainly for medical domain having the laboratory, radiology orders, results, clinical activity and prescription orders data elements.

In this paper, an innovative technique has been presented which generates highly correlated rule based uniformly distributed synthetic data for fraud analysis. Empirical results are presented by comparing the generated data and original reference data. We have compared the distribution of individual attributes and combinations of correlated attributes. Classification accuracy results for fraud detection are also observed with four well-known classification techniques. The empirical results show that the synthetic data preserves similar characteristics as the original reference data and have similar fraud detection accuracy.

Knowledge-based systems can represent knowledge with tools and rules rather than via code. Mostly current used fraud detection systems, use knowledge base in their architecture with rule-based as commonly used approach. Ripple Down Rules (RDR) was suggested by Compton & Jansen (Compton & Jansen, 1990) as a solution of maintenance and knowledge acquisition (KA) issues in knowledge-based systems(KBS). Ripple Down Rules (RDR) is an approach to knowledge acquisition. RDR has notable advantages over conventional rule-bases; including, better, faster and less costly rule addition and maintenance approaches. Another benefit is the addition of prudence analysis of RDR systems which allows the system to detect when a current case is beyond the system's expertise by issuing a warning for the case to be investigated by the human. Prudence Analysis (PA) was introduced by Compton et al (Compton, et al., 1996).

The synthetic data generation approach can be used to generate data for any classification domain, but in this

paper, test data has been generated to simulate bank transactions to study fraud analysis in banking domain.

In the remainder of the paper, section 3 presents our methodology in detail, while section 4 presents empirical results to show the working of the proposed technique. Finally, paper is concluded in section 5.

3 Synthetic Data Generation Using Highly Correlated Rule Based Uniformly Distribution

Synthetic data is generated with following desired characteristics:

- In some attributes, the generated values should have constraints due to the attribute interdependency on those attributes.
- The continuous attributes values should be within predefined ranges set in the constraints.
- Single attributes should have similar attribute distributions.
- Paired attributes should have similar attribute distribution as the reference data.
- Data should have classification labels.

A high-level flowchart is given in Figure 1. The process is explained in Algorithm 1.

Step 1	Load Reference data in a two-dimensional matrix using (1)
Step 2	Check attribute interdependency. Calculate attributes and class distributions from Reference Data using (2).
Step 3	Generate the Ruleset
Step 4	Start New Instance
Step 5	Generate attributes values from 1 to n with discrete probability distributions using (4).
Step 6	Validate generated attributes values with the ruleset expressions. (if all expressions are not validated then ignore the instance) GOTO STEP 4
Step 7	If generated attributes are validated in STEP 6 then assign the classification label to these attributes (if not classified ignore the instance) GOTO STEP 4
Step 8	Validate class distribution (if not within range ignore the instance) GOTO STEP 4
Step 9	Finalize the Instance.
Step 10	Repeat from STEP 4 to 9 till required instance count matches.
Step 11	Store Generated Data.
Step 12	END

Algorithm 1

A true representation of a generated synthetic data can be ensured by generating Ripple Down Rules (RDR) from reference data and then generating data samples ensuring the distribution of both individual attributes and combinations of attributes remain the same as in the sample reference dataset. Uniform distribution is applied on the attributes to keep data similarity. An innovative HCRUD technique is proposed in this paper to generate synthetic data with desired characteristics.

Reference data is a two-dimensional matrix as given in (1)

$$D_R = [\, d_{ij} \,] \quad (1)$$

where D_R is reference data and *i* are the attributes from 1 to *n* and *j* are rows from 1 to *m*.

Due to attribute interdependency in some attributes, constraints are applied to those attributes. The probability distribution of attributes is calculated with the ratio of the instances having a particular attribute value over the total instances in the reference dataset.

$$P_i = | D^i_R | / | D_R | \quad (2)$$

Where P*i* is the proportional value of the attribute *i* and $| D_R |$ is the cardinality of D_R i.e reference data and D^i_R are the instances having attribute *i*.

In the first step, reference data is loaded from the source in the form of a matrix. Attribute interdependency, attributes and class distributions are calculated in 2nd step. In the 3rd step, rules are generated from the reference data. Instance creation is initiated in 4th step. In step five attributes values are generated by applying attribute interdependency and the discrete probability distribution on single as well as the combination of attributes, which is calculated in step 2. In sixth step an instance is formed with the generated attribute values which are then validated based on the established rules, ensuring single and multiple attributes distributions resemble with reference data. The instance is ignored if the instance is not validated from all the expressions from the ruleset. In step seven, after validating the instance, a classification label is generated for the instance. In step eight, it is ensured that class distribution in the generated datasets is also maintained by ensuring the class distribution is within the threshold values. The instance is also ignored if a particular class distribution exceeds the threshold, calculated in step 2. Instance is finalised in step 9 and the steps 1 to 9 are repeated till the desired instance count is reached. Figure 1 is showing a high-level flowchart of the data generation process.

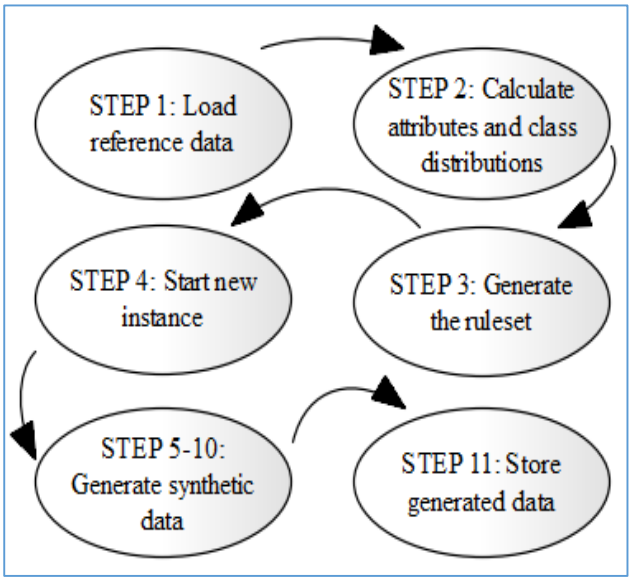


Figure 1: Synthetic Data Generation

The process of generating synthetic data is explained in detail in Figure 2.

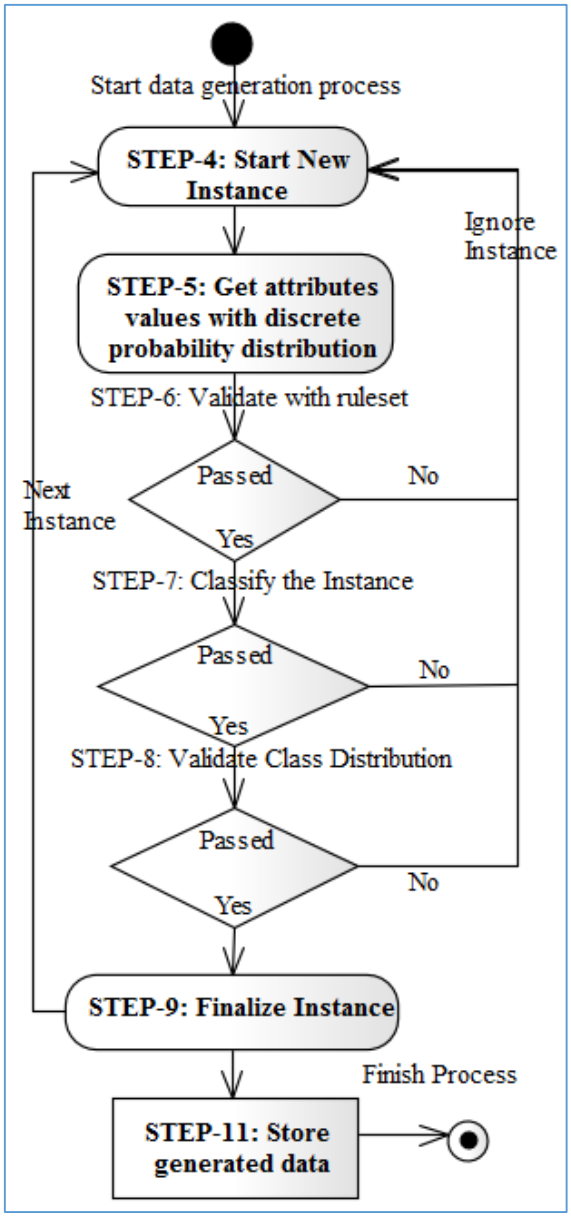


Figure 2: Detailed process to generate data

3.1 Applying HCRUD to Generate a Synthetic Fraud Dataset

To evaluate fraud detection algorithms in the banking data logs, a synthetic data emulating bank transactions has been generated, which is a mix of numerical and alphabetical attributes. An obfuscated dataset of 1775 internet banking transactions from a commercial bank was used to generate synthetic data. Although the dataset is small, the HCRUD technique presented in this paper demonstrates that a synthetic dataset can be generated of any desired size from small reference data. Format and structure of a typical online bank transaction dataset is given in (Maruatona, 2013). The attributes of sample dataset are shown in Table 1. Different banks and fraud detection systems adopt different nomenclatures for transactions.

Name	Description	Type
Transaction ID	Unique ID for transaction	Label
Transaction Type	Type of transaction	Discrete
Account From	Source account	Label
Account To	Destination account	Label
Account Type	Type of account in use	Discrete
Event time	Time of transaction	Time
Session ID	Unique session ID	Label
Browser String	String describing browser	Label
IP Address	IP address for machine	Label
Country	Host country for given IP	Label
Trans Amount	Transfer amount (if Transfer)	Continuou
Biller Code	Unique biller code	Label
Biller Name	BPay Biller business name	Label
Log in ID	User's log in ID	Label
Log in Time	Time of log in	Time
Log in Count	Logins count for the day	Continuou
Password change	Password changes count	Continuou s

Table 1: A sample bank transaction attributes

Discrete probability distribution has been applied on the combination of attributes, i.e. transaction type and class to ensure close resemblance with the sample data:

$$F(x) = P(a \leq x \leq b) = \sum_{k=a}^{b} f(k) \quad (3)$$

where x takes value k between a and b. For combination of the attributes, x is representing the combined value of the paired attributes Transaction Type and Class. Table 2 shows the distribution detail for the combination of attributes.

Transaction Type	Class	Probability
BPAY	Anon	0.022
BPAY	Fraud	0.083
BPAY	Non	0.208
PA	Anon	0.076
PA	Fraud	0.226
PA	Non	0.386

Table 2: Distribution of the attributes for the combination of attributes

Where PA is Pay Anyone and BPAY is a transaction type through which utility bills and other service providers can be directly paid. The class attribute represents the classification of Anon as anonymous and Non as not a fraud. Only one combination of paired attributes is shown as an example here. More paired attributes, even more than two attributes can also be taken, but the more attributes we add, the more would be the ignored instances as mentioned in step 6 in Algorithm 1; hence it will take more time to generate the synthetic dataset. Experimental evaluation has shown that there are about 0.1% to 0.12% ignore cases by taking one combination of paired attributes.

Similarly, discrete probability distribution is applied on individual attributes i.e. transaction type and class separately as shown in (4)

$$\sum_{k=a}^{b} f(k)=1 \quad (4)$$

Table 3 and 4 show the distribution details for single attributes.

Transaction Type	Probability
BPAY	0.313
PA	0.688

Table 3: Single attribute distribution for Transaction Type

Account Type	Probability
Business	0.227
Other	0.001
Personal	0.773

Table 4: Single attribute distribution for Account Type

Sum of the probabilities for both individual attributes is 1.0 Transaction Type and Account Type are the most significant attributes, so distributions detail of these two attributes is discussed above as an example.

3.2 Classification Techniques used for data validation

The system is trained with generated datasets and tested on bank dataset. Datasets of different sizes were generated ranging from 5,000 to 1 million; detail is given in Table 8. Classification accuracy of the generated dataset is observed and compared with four well-known classification techniques, which are Decision Tree (Quinlan, 1993), Ripple Down Rules (Compton & Jansen, 1990) (Richards, 2009)**,** Naïve Bayes (Swain & Sarangi, 2013) and RandomForest (Breiman, 2001).

3.2.1 Instance-Based Learning (IBL)

Aha et al have presented an instance-based learning (IBL) framework which generates classification predictions using only specific instances by applying similarity functions (Aha, et al., 1991). IB1 and IBk are instance-based learners (IBL) (Chilo, et al., 2009) which are also used for testing the classification accuracy in this paper. IB1 is the simplest instance-based learning, nearest neighbour algorithm where similarity function is used. It classifies the instance according to the nearest neighbour identified by Euclidean distance approach (Chilo, et al., 2009) (Aha, et al., 1991). IBk is similar to IB1, but the difference is that in IBk, the K-nearest neighbours are used instead of only one. Three different distance approaches are employed in IBk, including Euclidean, Chebyshev and Manhattan Distance (Chilo, et al., 2009).

3.3 HCRUD Implementation for Data Generation

Weka is well-known data mining tool having a collection of machine learning algorithms and Ridor is a Ripple Down Rules(RDR) implementation in Weka. In this paper, RDR ruleset is generated by using RDR classification from Ridor.

$$R_R = C(D_R) \quad (5)$$

where R_R is set of RDR format ruleset obtained by RDR classification function C. When reference data D_R is classified with Ridor in Weka, it not only classifies the data but also generates a ruleset in RDR format.

A sample format of Ripple Down Rule Learner ruleset is given in Figure 3 that is used in this technique to produce rules from reference data.

```
Except (Browser = Alt) => Class = Fraud  (546.0/0.0) [252.0/0.0]

Except (Network_Count <= 6.5) and (Transfer_Amt > 277.75) => Class = Non  (37.0/0.0) [14.0/0.0]

Except (Network_Count <= 11) and (Login_Count > 11.5) and (Login_Count <= 16.5) => Class = Non  (41.0/1.0) [31.0/1.0]

Except (Source_Acc = Business) and (Network_Count > 8) => Class = Non  (4.0/0.0) [1.0/0.0]

Except (Network_Count <= 2.5) and (Acc_Type = PA) and (LogTime = PM) and (Network_Count > 1.5) => Class = Fraud (28.0/4.0) [7.0/1.0]
```

Figure 3: A sample of an RDR Ruleset

JEXL name stands for Java Expression Language, an implementation of Unified EL(Expression Language) (Foundation, 2015), JEXL is used to get advantage of extra operators which is used in the rules compactness and to facilitate the implementation of dynamic and scripting features in this technique. The ruleset is transformed from RDR format to JEXL format, attributes-distributions and weightage calculated from reference data is fed to the proposed technique to generate the synthetic data. Figure 1 shows the abstract representation of the technique, while Figure 2 shows the detailed working of the synthetic data generation process. For compactness and efficiency, the generated rules are transformed to (JEXL) format:

$$R_J = T(R_R) \quad (6)$$

where R_J is JEXL format ruleset and R_R is set of RDR rules and *T* is transformation function of RDR ruleset.

A typical sample of JEXL expressions is shown in Figure 4.

```
Network_Count > 10 & Network_Count <= 12

Transfer_Amt > 2990 & Browser = Moz_4 & Country = AU

Login_Count <= 3 & Country = UK

BPay_Amt > 4750 & Browser = Moz_5Win & Country = AU

Transfer_Amt > 1005.5 & Browser = Opera

Acc_Type = BPAY & Source_Acc = Credit & Browser = Moz_4

PwdChange > 1 & Browser = Moz_5Win
```

Figure 4: JEXL expressions sample

Single classification, JEXL based implementation of RDR is developed and used in this technique to generate class labels to each generated instance. HCRUD generates dataset in variety of formats including Comma separated values (CSV), LibSVM and Attribute-Relation File Format (ARFF), which are widely used data formats in any data mining and machine learning tools. A comma separated values (CSV) format is shown in Table 5 as an example.

Transaction Type	Amount	Account type	Login Count	Network Count	Pwd Changes	Login Time	Browser String	Country	Class
PA	4,000	Other	1	1	1	AM	Alt	Other	Non
BPAY	1,200	Personal	6	3	0	AM	Alt	Other	Non
PA	3,000	Business	1	1	0	AM	Moz_4	AU	Fraud
PA	4,000	Personal	7	1	0	AM	Alt	Other	Fraud
BPAY	860	Personal	3	3	0	AM	Opera	AU	Non
PA	1,500	Personal	14	3	3	AM	Moz_4	AU	Fraud
PA	1,422	Personal	13	2	0	AM	Alt	Other	Non

Table 5: Example Dataset

4 Results

After generating the datasets, the next step was to compare it with original reference data as a benchmark using two different measures. One of the measures was to check the attribute distributions in the reference and generated datasets. Distributions of individual as well as the combination of correlated attributes were also verified, including class association. The second measure was to check the classification accuracy in terms of fraud detection by loading the generated data as training data and reference data as test data. Classification accuracy is verified in Weka with four well-known classification techniques including C4.5/J48, RDR/RIDOR, RandomForest and Naïve Bayes. Instance-based learning classification algorithms (IB1 and IBk) were also used to further verify the classification accuracy outcomes.

4.1 Quality Metric for Attribute Distribution

Root mean squared error (RMSE) is used as a quality measurement indicator, by taking the square root of the mean of the square of all of the errors for data distributions for individual and the combination of attributes. It is represented in (7).

$$RMSE = \sqrt{\frac{1}{N} \sum (D_R - D_G)^2} \qquad (7)$$

Where D_R is reference data and D_G is generated data.

4.1.1 RMSE for Combination of Attributes

RMSE for the distribution of individual attributes as well as combination of attributes were calculated and the experimental evaluation has shown that there is a minor difference in the attribute distribution of reference data and generated data.

The difference in data distribution for the combination of attributes in reference and generated datasets is shown in Table 6.

Transaction Type & Class	Error
BPAY/Anon	0.80
BPAY/Fraud	1.18
BPAY/Non	1.81
PA/Anon	0.80
PA/Fraud	1.22
PA/Non	1.85

Table 6: Error in distribution for the combination of attributes

4.1.2 ..RMSE for Individual Attributes

The difference in data distribution for individual attributes is shown below in Table 7.

Attribute	Value	Error
Class	Anon	0.11
Class	Fraud	0.11
Class	Non	0.00
Transaction Type	BPAY	0.16
Transaction Type	PA	0.22
Account Type	Business	0.03
Account Type	Other	0.03
Account Type	Personal	0.12
Country	AU	0.05
Country	Other	0.11
Browser String	Alt	0.78
Browser String	Mozilla	0.78

Table 7: Error in distribution for single attributes

4.2. Class and Attribute Distributions

Comparisons of the class distribution and distribution of individual as well as the combination of correlated attributes are excellent measures to check how close the generated data is to the original reference data. Fifty datasets were generated and classification and distribution results were averaged and compared with the original reference data.

Figure 5 shows the comparison of distribution by class in generated dataset and in reference dataset; which is very similar.

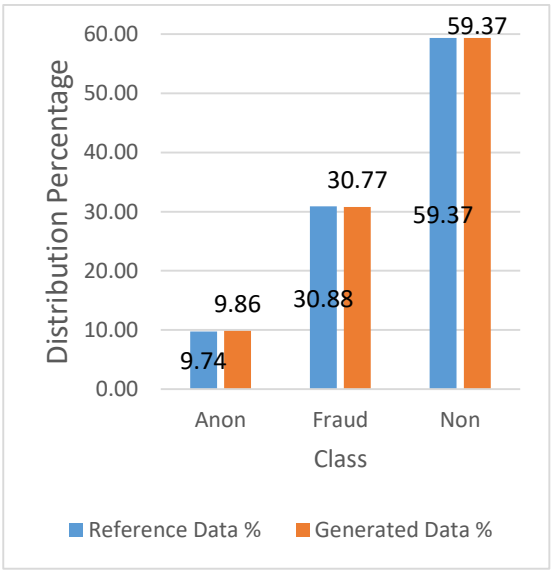


Figure 5: Distribution by class

Figure 6 shows the comparison of the distribution of the combination of attributes (Transaction Type and Class) in generated dataset and in the reference dataset. The results show that the percentages of values from both datasets are very close to each other.

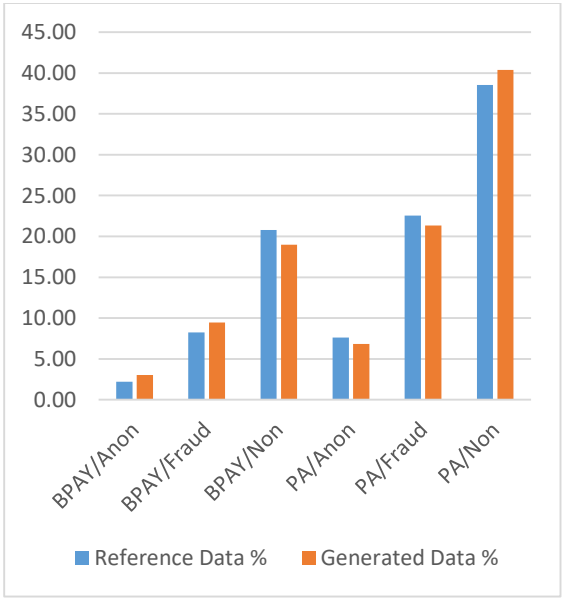


Figure 6: Distribution by Transaction Type and Class

Average time taken to generate instances is also calculated for the individual datasets. Results show that average time taken to generate 1,000 instances is 2.67 seconds. Maintaining attribute and class distributions and assigning class labels to the instance are the few factors, due to which more time is being taken to generate the synthetic datasets. Figure 7 shows the time taken to generate each dataset. It also shows the trend line of time and data size.

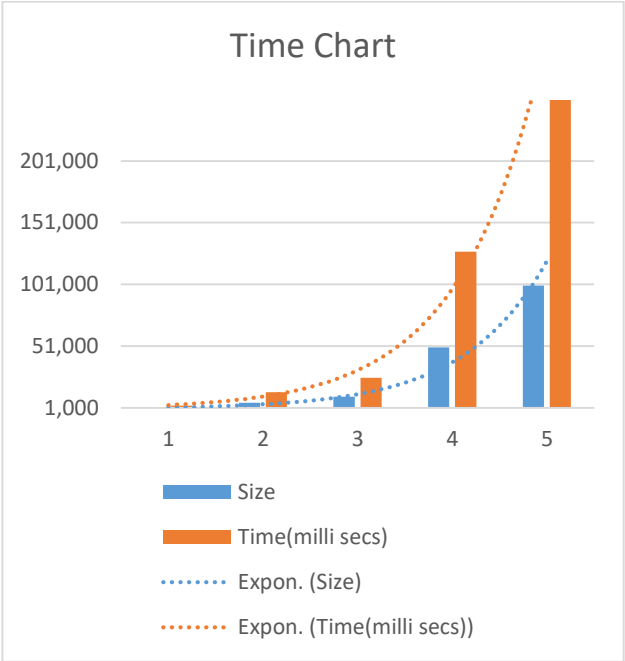


Figure 7: Time taken to generate datasets

4.3 Comparing Classification Accuracy for Fraud Detection

Classification accuracy of the generated dataset is tested with four well-known classification techniques. Table 8, 9 and 10 contain the classification accuracy results; where generated data is used as training data while reference data as test data using C4.5/J48, RDR/RIDOR, Naïve Bayes, and RandomForest classification techniques and instance-based learning (nearest neighbour, similarity based) algorithms as well. The mean classification accuracy for all generated datasets as well as the individual dataset is calculated and is very close to the individual accuracy percentage values.

Dataset	RDR	C4.5	Naïve Bayes	Random Forest	Class. Mean
5k	72.19	75.27	65.70	71.34	71.13
10k	73.96	75.50	65.62	71.74	71.70
25k	76.58	76.24	65.19	75.24	73.31
50k	76.98	76.64	65.41	75.73	73.69
100k	76.98	76.81	65.36	77.09	74.01
500k	77.04	76.98	65.19	77.44	74.10
1mil	76.98	76.92	65.13	77.98	74.22
Dataset Mean	76.03	76.34	65.37	74.93	73.17

Table 8: Fraud Detection classification accuracy results

Classification accuracy results are showing that with the increase of training data (generated data), there is an increase in the accuracy percentage in RDR, C4.5, RandomForest and Classification mean column as well.

Another testing is also performed using cross validation with fold=1755 for both reference and generated data. Fold value of 1755 was taken, due to the reference data size of 1755 instances. Table 9 shows the classification result with four classification techniques with both Reference data and generated data.

Classification	Reference Data	Generated Data	Difference
RDR	77.83	94.02	16.18
C4.5	87.41	96.70	9.29
Naïve Bayes	70.09	89.23	19.15
Random Forest	89.40	94.81	5.41

Table 9: Classification accuracy results with Cross validation

The results are showing that classification accuracy is higher when the system is trained on generated data.

To further verify classification accuracy with instance-based learning (nearest neighbour, similarity based) algorithms, we have performed the evaluation with IB1 and IBk algorithms. Classification accuracy results with instance-based learning are presented in Table 10.

	IBk	IBk	IBk	IB1
Dataset	Euclidean Distance	Chebyshev Distance	Manhattan Distance	
5k	65.64	64.50	66.84	66.95
10k	68.03	67.01	67.12	68.09
25k	71.19	69.18	72.42	72.29
50k	71.69	69.95	73.08	72.89
100k	72.59	70.71	73.73	73.11
500k	73.33	71.28	73.05	73.22
1mil	74.30	73.11	75.44	75.10

Table 10: Classification accuracy results with Instance-Based Learning algorithms

Classification accuracy results shown in Table 8,9 and 10 depict that with the increase of training data (generated data), there is an upward trend of the classification accuracy percentage.

5 Conclusion

To overcome a challenge of limited availability of datasets for fraud analysis studies for financial institutions, an innovative technique: highly correlated rule based uniformly distributed synthetic data has been presented to generate synthetic data. In this paper, we have presented the comparison of the distributions of the original and the synthetic data and the comparison of fraud detection classification accuracy with well-known classification techniques. A single classification, JEXL based Java implementation of RDR is developed and used to generate class labels to each generated instance. In classification accuracy testing, we used generated data as training and original data as test data. Empirical results show that synthetic dataset preserves a high level of accuracy and hence, the correlation with original reference data. Finally, we used an RMSE as a quality metrics for root mean square error to determine the difference of data distribution for individual and the combination of attributes in generated datasets as compared to original reference datasets. Studies have shown very similar distributions of the attributes of generated datasets.

Currently, we are generating the dataset with only 13 attributes of an obfuscated dataset. It needs to be more efficient, otherwise, for high dimensional data, it will take more time. One of the recommended future work is to test this technique on high dimensional data, while another work is to handle missing values from the reference data.

6 References

Aha, D. W., Kibler, D. & Albert, M. K., 1991. Instance-based learning algorithms. *Machine Learning*, 01, 6(1), pp. 37-66.

Anon., 2015. *Generatedata.* [Online] Available at: http://www.generatedata.com/ [Accessed 13 9 2015].

Anon., 2015. *Open Jail - The Jailer Project.* [Online] Available at: http://jailer.sourceforge.net/

Ayala-Rivera, V., McDonagh, P., Cerqueus, T. & Murphy, L., 2013. Synthetic Data Generation using Benerator Tool.

Bergmann, V., n.d. *Databene Benerator.* [Online] Available at: http://databene.org/databene-benerator [Accessed 16 9 2015].

Bolton, R. J. & Hand, D. J., 2002. Statistical Fraud Detection: A Review. *Statistical Science,* 17(3), pp. 235-254.

Breiman, L., 2001. Random Forests. *Machine Learning,* pp. 5-32.

Buczak, A. L., Babin, S. & Moniz, L., 2010. Data-driven approach for creating synthetic electronic medical records. *BMC Medical Informatics and Decision Making*, 10(1), pp. 59-59.

Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P., 2002. SMOTE: Synthetic Minority Over-sampling Technique. pp. 321-357.

Chilo, J., Horvath, G., Lindblad, T. & Olsson, R., 2009. *Electronic Nose Ovarian Carcinoma Diagnosis Based on Machine Learning Algorithms.* s.l., Springer, pp. 13-23.

Christen, P. & Vatsalan, D., 2013. *Flexible and extensible generation and corruption of personal data.* s.l., ACM, pp. 1165-1168.

Compton, P. & Jansen, R., 1990. *Knowledge in context: a strategy for expert system maintenance.* Berlin Heidelberg, s.n.

Compton, P., Preston, P., Edwards, G. & Kang, B., 1996. *Knowledge based systems that have some idea of their limits.* Sydney, s.n.

Coyle, E. J., Roberts, R. G., Collins, E. G. & Barbu, A., 2013. Synthetic data generation for classification via uni-modal cluster interpolation. *Autonomous Robots,* pp. 27-45.

DeMilli, R. A. & Offutt, A. J., 1991. Constraint-based automatic test data generation. *Software Engineering, IEEE Transactions on,* pp. 900-910.

Foundation, T. A. S., 2015. *Java Expression Language (JEXL).* [Online]
Available at:
http://commons.apache.org/proper/commons-jexl/

Maj, P., 2015. *DBMonster Core.* [Online]
Available at: http://dbmonster.sourceforge.net/

Marican, L. & Lim, S., 2014. *Microsoft Consumer Safety Index reveals impact of poor online safety behaviours in Singapore.* [Online]
Available at: https://news.microsoft.com/en-sg/2014/02/11/microsoft-consumer-safety-index-reveals-impact-of-poor-online-safety-behaviours-in-singapore/

Maruatona, O., 2013. *Internet banking fraud detection using prudent analysis,* Ballarat: University of Ballarat.

McCombie, S., 2008. *Trouble in Florida, The Genesis of Phishing attacks on Australian Banks.* Perth, s.n.

Quinlan, J. R., 1993. *C4.5 : programs for machine learning.* San Mateo, Calif.: Morgan Kaufmann Publishers.

Richards, D., 2009. Two decades of ripple down rules research. *The Knowledge Engineering Review,* 24(02), pp. 159-184.

Rubin, D. B., 1993. Statistical Disclosure Limitation. *Journal of Official Statistics,* 9(2), pp. 461-468.

Swain, S. & Sarangi, S. S., 2013. Study of Various Classification Algorithms using Data Mining. *International Journal of Advanced Research in,* 2(2), pp. 110-114.

Yoo, S. & Harman, M., 2012. Test data regeneration: generating new test data from existing test data. *Software Testing, Verification and Reliability,* pp. 171-201.

Running Boolean Matrix Factorization in Parallel

Jan Outrata **Martin Trnecka**

Department of Computer Science
Palacký University Olomouc, Czech Republic
Email: jan.outrata@upol.cz, martin.trnecka@gmail.com

Abstract

Boolean matrix factorization (also known as Boolean matrix decomposition) is a well established method for analysis and preprocessing of data. There is a number of various algorithms for Boolean matrix factorization, but none of them uses benefits of parallelization. This is mainly due to the fact that the algorithms utilize greedy heuristics that are inherently sequential. In this work, we propose a general parallelization scheme—and an algorithm which uses it—for Boolean matrix factorization. Our approach computes several possible locally most optimal (from heuristic perspective) partial decompositions and constructs several most optimal final decompositions in more processes running simultaneously in parallel. As a result of the computation, either the single most optimal decomposition or several top-k of them can be returned. The approach could be applied to any sequential heuristic Boolean matrix factorization algorithm. Moreover, we present results of various experiments involving this new algorithm on synthetic and real datasets.

Keywords: Boolean matrix decomposition; parallel algorithm

1 Introduction

Boolean Matrix Factorization (BMF) is a problem of decomposing a Boolean matrix into two Boolean matrices such that the (Boolean) matrix product of the two matrices exactly or approximately equals the given matrix. In a variant of the problem called Approximate Factorization Problem (AFP) (Belohlavek et al. 2010), a (non-trivial) solution with the inner matrix product dimension as low as possible, for a given maximal, usually zero, difference (error) of the product from the input matrix, is requested. Another variant, called Discrete Basis Problem (DBP) (Miettinen et al. 2008), demands minimal error for a given maximal inner dimension. We will focus on the AFP in this paper. The least dimension for which an exact decomposition of a Boolean matrix exists is called the *Boolean rank* (or Schein rank) of the matrix. As the problem of finding the Boolean rank for a given Boolean matrix as well as the AFP and the DBP is NP-hard (due to the NP-hardness of the set basis problem (Stockmeyer 1975)), existing BMF algorithms seek for a sub-optimal decomposition with the dimension as close to the Boolean rank as possible, utilizing some heuristic approach.

Well-recognized efficient algorithms are GreConD (Belohlavek et al. 2010) and GreEss (Belohlavek et al. 2015), both designed for the AFP, and Asso (Miettinen et al. 2008) for the DBP. Other competitive algorithms for either AFP or DBP include e.g. PaNDa (Lucchese et al. 2010) or Hyper (Xiang et al. 2011). Being heuristic, the algorithms construct the final decomposition from partial (approximate) decompositions which are only locally optimal among all possible partial decompositions. The choice of optimal partial decomposition is usually hardcoded in the algorithm design and for performance reasons one cannot afford in the algorithm to explore several most optimal decompositions (or even all of them) and then choose among them the most optimal one discarding the others. This is true for sequential algorithms, as all the above mentioned algorithms are sequential, and this is where a parallel computation approach could be used. Moreover, with the development and growing affordability of multicore processors and other hardware allowing parallel computations, interest in parallel computing increases and parallel algorithms are preferred to better utilize the hardware.

Interestingly, to our knowledge, there is no parallel algorithm for *Boolean* matrix factorization today introduced in the literature. The adjective 'Boolean' needs to be emphasized here. There are parallel algorithms for some of the many existing factorization methods designed originally for real-valued matrices, see for instance (Berry et al. 2006) or (Kannan et al. 2016), and those algorithms obviously can be applied also to Boolean matrices (and they are, though). However, as (Tatti et al. 2006) and other authors conclude, a problem with applying to Boolean matrices the methods designed originally for real-valued matrices is the lack of interpretability. And because of interpretability, which is crucial from the knowledge discovery point of view, BMF is considerably more appropriate to use with Boolean matrices than the methods designed originally for real-valued matrices. One of the reasons for the absence of a parallel BMF algorithm, however, may be that the most commonly used greedy heuristic approach, utilized in GreConD, GreEss and also in Asso, is inherently sequential. Other reason could be that factorization

The research was supported by grant No. GA15-17899S of the Czech Science Foundation. M. Trnecka also acknowledges partial support by grant No. PrF_2016_027 of IGA of Palacký University Olomouc.

of Boolean matrices is as a standalone research area relatively young within the data mining research and not so elaborated as the factorization of real-valued matrices.

Our contribution in this paper does not lie in a parallel algorithm for BMF which would compute a decomposition in a parallel manner either. Instead, we show a general parallelization scheme consisting in a viable way to compute in parallel several locally optimal decompositions and then select the most optimal one(s) hoping to find the globally optimal. In essence, as suggested above, the approach consists in following several possible choices of locally most optimal partial decompositions in the heuristic approach and construct several most optimal final decompositions in more processes running simultaneously in parallel. As a result of computation, either the single most optimal decomposition or several top-k of them can be returned—a distinctive feature of our approach. The approach could be applied, with more or less effort, to any sequential heuristic BMF algorithm. We chose as our base algorithm to demonstrate the approach the GreConD algorithm, due to its relative simplicity and high efficiency (for the AFP).

In the rest of the paper, the following Section 2 provides basic notions of Boolean matrix factorization, the main Section 3 contains first a brief description of the base GreConD algorithm and then a presentation of our approach of computing several matrix decompositions in parallel demonstrated on GreConD, including full pseudocodes of both original GreConD and our modification of it, Section 4 then presents results from basic experiments evaluating the approach, and finally Section 5 concludes the paper.

2 Boolean Matrix Factorization

Boolean matrix factorization (BMF), called also Boolean matrix decomposition, comprises various methods for analysis and processing of Boolean data, mostly for factorization or decomposition of the data. The data is in the form of Boolean matrices, i.e. matrices with entries either 1 or 0. We interpret such matrices primarily as object-attribute incidence relations, that is, the entry I_{ij} of a Boolean matrix I corresponding to the row i and the column j indicates that the object i does (value 1) or does not have (value 0) the attribute j. The ith row and jth column vector of I is denoted by $I_{i\_}$ and $I_{\_j}$, respectively. The set of all $n \times m$ Boolean matrices is denoted $\{0,1\}^{n\times m}$.

Generally speaking, the basic problem in BMF is to find for a given Boolean matrix $I \in \{0,1\}^{n\times m}$ Boolean matrices $A \in \{0,1\}^{n\times k}$ and $B \in \{0,1\}^{k\times m}$ for which

$$I \text{ (approximately) equals } A \circ B, \tag{1}$$

where $\circ$ is the Boolean matrix product, i.e.

$$(A \circ B)_{ij} = \max_{l=1}^{k} \min(A_{il}, B_{lj}).$$

Interpreting the matrices A and B as object-factor and factor-attribute incidence relations, respectively, such a decomposition of I into A and B may be interpreted as a discovery of k factors (k is the inner dimension of the product) exactly or approximately explaining I. In the factor model given by (1) matrices A and B explain I as follows: the object i has the attribute j ($I_{ij} = 1$) if and only if there exists factor l such that l applies to i ($A_{il} = 1$) and j is one of the particular manifestations of l ($B_{lj} = 1$). Thus, a factor in the model is naturally interpreted as an abstract property (or attribute), generally distinct from the m original attributes, which applies to some of the n objects and which is characterized by some of the m original attributes.

This easy interpretation of factors is further supported by the so-called geometric view on factors and on BMF, which is unfortunately not always recognized in the literature. We will use the view in the description of the GreConD algorithm below. Briefly, in the view each factor is identified with a rectangular matrix, or *rectangle* for short, a Boolean matrix whose entries with 1 form, upon a suitable permutation of rows and columns, a rectangular area (full of 1s). A decomposition of a Boolean matrix I using k factors then corresponds to a *coverage* of the entries of I containing 1s by k such rectangles (a Boolean matrix product from (1) is looked at as a max-superposition $\max_{l=1}^{k}(J_l)_{ij}$ of rectangles $J_l = A_{\_l} \circ B_{l\_}$).

In optimization versions of the basic BMF problem, the number k of factors is requested to be as small as possible (see the AFP below) and, as already mentioned in the beginning of the introduction section, the least k for which an exact decomposition $I = A \circ B$ exists is called the *Boolean rank* (or Schein rank) of I. The deviation from an exact decomposition, i.e. the approximate equality in (1), is assessed by means of the well-known L_1-norm $\|I\| = \sum_{i,j=1}^{m,n} |I_{ij}|$ and the difference of $A \circ B$ from I is measured by a distance (error) function $E(I, A \circ B)$ defined as

$$E(I, A \circ B) = \|I - A \circ B\| = \sum_{i,j=1}^{m,n} |I_{ij} - (A \circ B)_{ij}|.$$

Using E, quality of decompositions delivered by BMF algorithms is commonly assessed (Belohlavek et al. 2010, 2015, Geerts et al. 2004, Miettinen et al. 2008) by the following function representing the *coverage quality* of the first l factors delivered by the particular algorithm and measuring how well the data is explained by the l factors:

$$c(l) = 1 - E(I, A \circ B)/\|I\|.$$

We will use this function in Section 4 devoted to experiments where we also state what $c(l)$ should satisfy for a good factorization algorithm.

A (optimization) variant of the basic BMF problem of our concern is the approximate factorization problem (AFP):

Definition 1 (Approximate Factorization Problem, AFP (Belohlavek et al. 2015)). *Given $I \in \{0,1\}^{n\times m}$ and prescribed error $\varepsilon \geq 0$, find $A \in \{0,1\}^{n\times k}$ and $B \in \{0,1\}^{k\times m}$ with k as small as possible such that $\|I - A \circ B\| \leq \varepsilon$.*

AFP emphasizes the need to account for (and thus to explain) a prescribed (presumably reasonably large) portion of data, which is specified by ε.

For a more throughout study of the factor model and the geometric view on it described above, of the AFP and of the BMF in general, we refer the reader to (Belohlavek et al. 2010) or (Belohlavek et al. 2015).

3 BMF in Parallel Runs

In this section we present our approach of computing several Boolean matrix decompositions simultaneously in parallel. First, however, we need to recall the GreConD algorithm which we use as a base for our parallelization scheme.

3.1 GreConD Algorithm

The algorithm, proposed in (Belohlavek et al. 2010) and called Algorithm 2 there, solves the AFP by implementing greedy search for factors. I.e., each factor is sought to explain as much of the input Boolean matrix being decomposed as possible. The factors satisfying this property are, however, not selected among all candidate factors (as in the "classical" greedy approach), rather they are incrementally, or "on demand", computed with the aim to fulfill the property. And the computation is again in a greedy manner, see the description below. In the description we will use the geometric view on factors (as (Belohlavek et al. 2010) does too), identifying each factor with a rectangular matrix (rectangle) full of 1 as introduced in Section 2. Finding a decomposition of the input Boolean matrix I then means finding a coverage of 1s in I by such rectangles. GreConD finds factors as *maximal rectangles*. This is not accidental, maximal rectangles make factors better interpretable. Informally, maximal rectangles are rectangles which cannot be enlarged by adding another row or another column so that it remains a rectangle—hence factors, in this sense, apply to a maximal number of objects and are characterized by a maximal number of attributes. This rationale actually stems *Formal concept analysis* (FCA) (Ganter 1999) in which maximal rectangles correspond to so-called formal concepts (basic data units studied in FCA) and which is used as a description platform of the algorithm in (Belohlavek et al. 2010) (now, GreConD means "Greedy Concepts on Demand"). We do not use FCA in this paper, readers interested in (a fruitful) connection between BMF and FCA are referred to (Belohlavek et al. 2010) or (Belohlavek et al. 2015). We will briefly describe the GreConD algorithm now.

The algorithm, in its seek for a factor, starts with the empty set of attributes (characterized as the empty Boolean vector of size m or the empty $1 \times m$ Boolean matrix) which is repeatedly grown by a selected attribute. Together with the selected attribute other attributes may be possibly added to the set due to the construction of each factor as a maximal rectangle. Namely, the set of attributes after each addition is completed, or *closed*, to contain all attributes which are shared by all objects having the attributes. Such a set of attributes is called closed. The closed set of attributes together with the corresponding set of all objects having all the attributes determine a maximal rectangle. The selected attribute is such that the rectangle grown by the attribute covers as many still uncovered 1s in the input Boolean matrix I as possible. Within the aim of computing factors as rectangles which cover as many still uncovered 1s in matrix I as possible, the rectangle is grown repeatedly as long as the number of still uncovered 1s in I covered by the rectangle increases. The final maximal rectangle then represents a computed factor. Note the greedy and "on demand" computation of the factor. Further factors as maximal rectangles are, within the greedy factor search, sought the same way until the prescribed number of 1s in I is covered by the rectangles (the prescribed maximal error E is reached, recall that the algorithm is designed for the AFP). Finally, characteristic vectors of object sets determining found maximal rectangles/factors constitute columns of the object-factor matrix A and characteristic vectors of attribute sets of the rectangles/factors constitute rows of the factor-attribute matrix B. Matrices A and B, which determine the decomposition of the matrix I, form an output of the algorithm.

The above description results in a pseudocode of the algorithm depicted in Algorithm 1. D denotes the set of attributes determining a maximal rectangle (after closing) and is repeatedly grown by a selected attribute h between lines 4 and 16. Under the vector/matrix notation we use in the pseudocode D_j denotes the jth item of $D \in \{0,1\}^{1\times m}$ as a Boolean vector which effectively corresponds to the presence/absence of attribute j in D. The addition of h and the closure of D with h added are performed at line 14. Here, $[h] \in \{0,1\}^{1\times m}$ denotes the Boolean vector with hth item equal to 1 and all other items equal to 0 (i.e. the set with attribute h only) and the closure operator $^{\downarrow\uparrow}$ is a composition of the (so-called formal concept-forming (Ganter 1999)) operators $^{\uparrow}$ and $^{\downarrow}$. The operators are defined for a (column) Boolean vector $C \in \{0,1\}^{n\times 1}$ and a (row) Boolean vector D, respectively, as

$$C^{\uparrow} = \bigwedge [j] \in \{0,1\}^{1\times m}; I_{ij} = 1 \text{ for all } i \text{ s.t. } C_i = 1,$$
$$D^{\downarrow} = \bigwedge [i] \in \{0,1\}^{n\times 1}; I_{ij} = 1 \text{ for all } j \text{ s.t. } D_j = 1.$$

Note the meaning of the operators (in the set notation): $C^{\uparrow}$ contains all attributes shared by all objects in C and $D^{\downarrow}$ contains all objects having all attributes in D. The selection of attribute h is done between lines 7 and 13. The number of still uncovered 1s in the input Boolean matrix I covered by the rectangle determined by D, on which the selection is based, is computed by a function $cover(D, I, A, B)$ defined as

$$\|(D^{\downarrow} \times D^{\downarrow\uparrow}) \cdot (I - A \circ B)\|.$$

The $\times$ and $\cdot$ (dot) operations in the function definition denote the usual Cartesian and scalar matrix products, respectively. By line 6, D is repeatedly grown as long as the number computed by the function *cover* increases. I is then covered by the final maximal rectangles determined by D after growing which represent factors and the factors are "stored" in matrices A and

Algorithm 1: Original GreConD algorithm

Input: A Boolean matrix $I \in \{0,1\}^{n\times m}$ and a prescribed error $\varepsilon \geq 0$

Output: Boolean matrices $A \in \{0,1\}^{n\times k}$ and $B \in \{0,1\}^{k\times m}$

```
1  A ← empty n × 0 Boolean matrix
2  B ← empty 0 × m Boolean matrix
3  while ||I − A ∘ B|| > ε do
4  |  D ← empty 1 × m Boolean matrix
5  |  V ← 0
6  |  while there is j such that D_j = 0 and
   |   cover(D + [j], I, A, B) > V do
7  |  |  W ← 0
8  |  |  forall j do
9  |  |  |  if cover(D + [j], I, A, B) > W then
10 |  |  |  |  h ← j
11 |  |  |  |  W ← cover(D + [j], I, A, B)
12 |  |  |  end
13 |  |  end
14 |  |  D ← (D + [h])^{↓↑}
15 |  |  V ← W
16 |  end
17 |  A ← [A D^{↓}], B ← [B; D]
18 end
19 return A and B
```

B at line 17. Finally, finding the factors until they cover the prescribed number of 1s in I given by ε is wrapped between lines 1 and 18.

We can now proceed to the description of our approach to running the algorithm in parallel.

3.2 Parallel Runs of GreConD

As we saw above the GreConD algorithm implements a greedy search for factors in which each factor is computed to explain as much of the input Boolean matrix being decomposed as possible. While the alone factor computed this way may be optimal within the aim to explain by factors as much of the input matrix as possible, several (or all) factors together, forming a (final) decomposition, may be not. Hence, the partial decomposition formed by the factor and all factors computed previously is only locally optimal. Moreover, there can be more equally optimal factors instead of just one, forming more partial decomposition to choose from and generally leading to different final decompositions. Likely, the computation of a factor is also greedy, by growing a maximal rectangle by attributes selected so that the rectangle covers as many still uncovered 1s in the input matrix as possible. Similarly, while the alone attribute selected in such a way may be optimal within the aim to cover by the (final) maximal rectangle after growing as many still uncovered 1s in the input matrix as possible, more attributes together, determining a factor, may be not. Hence, here the partial factor (formed by the attribute and all attributes added to the corresponding maximal rectangle previously) is also only locally optimal. And finally, there can also be more equally optimal attributes to select, forming more partial factors to choose from and leading to different factors.

In our approach to parallel runs of a BMF algorithm, as aforementioned in the introduction section, we construct, simultaneously in parallel, several (locally) most optimal partial decompositions and select among them several most optimal final decompositions in hope to find the globally optimal one. For the GreConD algorithm it means that in the search for factors in each iteration several factors explaining most of the input Boolean matrix being decomposed are computed instead of just one, and in the factor computation in each iteration several attributes are selected so that the corresponding maximal rectangle covers most still uncovered 1s in the input matrix, instead of just one. Each of the factors computed, together with the factors computed previously, forms a (locally optimal) partial decomposition and each of the attributes (possibly with other attributes due to the closure), together with the attributes selected previously, forms a (locally optimal) partial factor. After each iteration, several most optimal partial decompositions or factors are selected for the next iteration. And while the computation of factors from the selected partial factors remains serial, the construction of the final decompositions from the selected partial ones is done in parallel.

In terms of a pseudocode the idea is included in the algorithm depicted in Algorithms 2 and 3. Algorithm 2 depicts an algorithm which, loosely speaking, represents several instances of the (modified) GreConD algorithm from Algorithm 1 where each instance is represented by the procedure depicted in Algorithm 3. All instances are running simultaneously in parallel—hence the name GreConDP as "GreConD in Parallel runs"—and *jointly* construct several most optimal decompositions of input Boolean matrix I. The number of instances equal to the number

Algorithm 2: GreConDP – GreConD in parallel runs

Input: A Boolean matrix $I \in \{0,1\}^{n\times m}$ and a prescribed error $\varepsilon \geq 0$
Output: Boolean matrices $A_1 \in \{0,1\}^{n\times k}$ and $B_1 \in \{0,1\}^{k\times m}$
Uses: Boolean matrices $A_r \in \{0,1\}^{n\times k}$ and $B_r \in \{0,1\}^{k\times m}$, values U_r, $r \in \{1,\ldots,P\}$, and a number $P \geq 1$ of processes

```
1  A_r ← empty n × 0 Boolean matrix
2  B_r ← empty 0 × m Boolean matrix
3  U_r ← 0 (r ∈ {1,...,P})
4  GreConDP-I(I, ε, A_1, B_1, true)
5  for r = 1,...,P do
6  |  with process r
7  |  |  GreConDP-I(I, ε, A_r, B_r, false)
8  |  end
9  end
10 wait for all processes
11 return A_1 and B_1
```

of decompositions is given by (presently equal to) the number P of processes in which the instances are run, see lines 5 to 9 of Algorithm 2. The decompositions of matrix I are determined by Boolean matrices A_r and B_r, $r \in \{1,\ldots,P\}$, sorted in the descending order from the most optimal one (by increasing r). The first one only, i.e. the most optimal one, determined by A_1 and B_1, is output.

Let us now focus on Algorithm 3 depicting the core of the modified GreConD algorithm and compare it to the original version of GreConD depicted in Algorithm 1. The procedure GreConDP-I constructs the ith decomposition (ith in the time of calling from Algorithm 2, the order of decompositions may change, see below) determined by matrices A_i and B_i. Several, actually P, sets of attributes determining P maximal rectangles (after closing) representing partial factors are denoted by D_r. By lines 4 and 5, all those sets are repeatedly (and serially) grown by several selected attributes between lines 2 and 33. Just in the first iteration of growing, when all D_rs are empty Boolean matrices, only D_1 is grown, see line 30. The selection of the attributes for a particular D_r is done between lines 6 and 18. Compare it to lines 7 to 13 in Algorithm 1. Instead of just one attribute h, P attributes h_s such that the maximal rectangle determined by D_r with h_s added covers most still uncovered 1s in the input matrix I are selected and the attributes are, as a bonus here, sorted (by increasing s, using the Insertsort sorting algorithm) in the descending order from the greatest number of 1s covered by the rectangle (computed by function *cover*). In addition, we need to check that the rectangles are distinct, at line 10 (adding different attributes to the same maximal rectangle can result in the same maximal rectangle).

Among all the sets grown from a D_r by all P selected attributes h_p, P of the sets only determining maximal rectangles which cover most still uncovered 1s in I, over all D_rs, are stored as E_s, between lines 19 and 29 (extending lines 14 and 15 in Algorithm 1). The sets E_s are sorted (by increasing s) in the descending order from the greatest number of 1s covered by the rectangle and here the sorting helps to choose the P sets. After an iteration of growing of all D_rs each E_s is renamed to D_s for another iteration of growing, line 32. When the repeated growing is finished (when the number computed by the func-

Algorithm 3: Procedure GreConDP-i

Input: A Boolean matrix $I \in \{0,1\}^{n\times m}$, a prescribed error $\varepsilon \geq 0$, Boolean matrices $A_i \in \{0,1\}^{n\times k}$ and $B_i \in \{0,1\}^{k\times m}$ and a Boolean flag $first$

Uses: Boolean matrices $A_r \in \{0,1\}^{n\times k}$ and $B_r \in \{0,1\}^{k\times m}$, values U_r, $r \in \{1,\ldots,P\}$, and a number $P \geq 1$ of processes

```
1  while ||I − A_i ∘ B_i|| > ε do
2      D_r ← empty 1 × m Boolean matrix
3      V_r ← 0 (r ∈ {1,...,P})
4      while there is ⟨r,j⟩ such that (D_r)_j = 0 and
        cover(D_r + [j], I, A_i, B_i) > V_r do
5          forall r from the ⟨r,j⟩ do
6              W_s ← 0 (s ∈ {1,...,P})
7              forall j from the ⟨r,j⟩ do
8                  s ← P
9                  while s > 0 and
                    cover(D_r + [j], I, A_i, B_i) > W_s
                    do
10                     if s > 1 and
                        (D_r + [j])^{↓↑} = (D_r + [h_{s−1}])^{↓↑}
                        then break
11                     if s < P then
12                         h_{s+1} ← h_s, W_{s+1} ← W_s
13                     end
14                     h_s ← j
15                     W_s ← cover(D_r + [j], I, A_i, B_i)
16                     s ← s − 1
17                 end
18             end
19             for p = 1,...,P do
20                 s ← P
21                 while s > 0 and W_p > V_s do
22                     if s < P then
23                         E_{s+1} ← E_s, V_{s+1} ← V_s
24                     end
25                     E_s ← (D_r + [h_p])^{↓↑}
26                     V_s ← W_p
27                     s ← s − 1
28                 end
29             end
30             if D_1 = empty 1 × m Boolean matrix
                then break
31         end
32         for r = 1,...,P do D_r ← E_r
33     end
34     U_i ← 0
35     synchronization barrier
36     for r = 1,...,P do
37         s ← P
38         begin critical section
39             while s > 0 and V_r > U_s do
40                 if s < P then
41                     A_{s+1} ← A_s, B_{s+1} ← B_s
42                     U_{s+1} ← U_s
43                 end
44                 A_s ← [A_i D_r^↓], B_s ← [B_i; D_r]
45                 U_s ← V_r
46                 s ← s − 1
47             end
48         end
49     end
50     synchronization barrier
51     if first = true then break
52 end
```

tion *cover* at line 4 does not increase) we have P final maximal rectangles (determined by D_r) representing factors which have to be "stored" in matrices A_i and B_i.

This is done between lines 34 and 50, over all the P decompositions (determined by A_is and B_is) constructed in the P processes running the procedure GreConDP-i in parallel. Therefore, due to data consistency reasons when storing the factors, we need to wait until all processes compute the factors and before they start to compute further factors. Hence the synchronization barriers at lines 35 and 50. Among all the factors computed by the processes, P of the factors only which explain most of the input Boolean matrix I are stored in matrices A_i and B_i (line 44) and the matrices are sorted (by increasing s) in the descending order from the greatest number of 1s covered by their factors. Again, due to the data consistency reasons, the storing of factors and sorting need to be done in a critical section, i.e. with inter-process switching disabled between lines 38 and 48. Note also that due to the sorting of matrices A_i and B_i determining the ith decomposition, the order of decompositions may change. But each process constructs still the same decomposition (ith in the time of calling the procedure GreConDP-i in Algorithm 2), until the prescribed number of 1s in I given by ε is covered.

The last comment is on line 51 of Algorithm 3 and line 4 of Algorithm 2. Since calling the procedure GreConDP-i in Algorithm 2 with the empty decomposition as input in P processes does not make sense (because all would compute the same P factors from which the same first one explaining most of the input Boolean matrix would just be stored to P copies the same decomposition), we first call the procedure once to construct the first P non-empty decompositions (in the first iteration) only and after that we can continue the construction of the decompositions in the P processes.

4 Experimental Evaluation

Now we provide an experimental evaluation of the GreConDP algorithm described in the previous section and present a comparison with the base algorithm GreConD. We do not include a comparison with other algorithms and approaches to the general BMF. A comparison of GreConD with other BMF algorithms can be found e.g. in (Belohlavek et al. 2015).

4.1 Datasets

As in the typical experiment scenario, which occurs in various BMF papers, we use both synthetic and real datasets which are described below. Experiments on synthetic datasets enable us to analyze performance of the algorithms on data with the same and known characteristics—we can analyze results in the average case. On the other hand such data are fully artificial while real data are influenced by real factors.

4.1.1 Synthetic Data

We created 1000 of randomly generated datasets. Every dataset X_i has 500 rows (objects) and 250 columns (attributes) and was obtained as a Boolean product $X_i = A_i \circ B_i$ of Boolean matrices A_i and B_i that were both generated randomly. Final densities (the ratio of 1s in the Boolean matrix) of datasets are 0.05, 0.1, 0.15, 0.2 and 0.3 and there is the same

number of datasets with each density. The inner dimension of matrices A_i and B_i in the Boolean matrix product was set to 40 for all datasets, i.e. the expected number of factors is 40 (but the Boolean rank can be lower). Data generated in this way are standard for evaluation of BMF algorithms (Belohlavek et al. 2015, Miettinen et al. 2008).

4.1.2 Real Data

We used the datasets Emea (Ene et al. 2008), DBLP (Miettinen et al. 2008), Firewall 1 (Ene et al. 2008), Mushroom (Bache at al. 2013), Paleo[1] and Zoo (Bache at al. 2013), see Table 1. All of them are well known and used in the literature on BMF. The characteristics of the datasets in the table are number of objects × number of attributes (column Size), ratio of 1s in the dataset Boolean matrix (Dens. 1) and the average number of equally locally optimal factors per factor in the GreConD algorithm (column Equal).

Dataset	Size	Dens. 1	Equal
Emea	3046×35	0.095	157.279
DBLP	19×6980	0.130	2.105
Firewall 1	365×709	0.124	31.168
Mushroom	8124×119	0.193	3.148
Paleo	501×139	0.051	5.868
Zoo	101×28	0.305	5.867

Table 1: Real datasets and their characteristics

The last characteristics is an important one. Recalling Section 3.2, factors computed by GreConD (and other heuristic BMF algorithms) are alone locally optimal within the aim to explain by factors as much of the input Boolean matrix being decomposed as possible. For a partial decomposition formed by a factor and all factors computed previously there can, however, be more than one equally optimal factors to add to the decomposition, generally leading to different final decompositions. Classical sequential heuristic BMF algorithms like GreConD then have to select one and this selection is algorithm or implementation dependent. This problem is reduced in our parallelization scheme—instead of selecting one of the equally optimal factors several (or potentially all) of them producing most optimal decompositions are considered. And since the choice of equally optimal factors influences results obtained by the evaluated algorithms, as we will see below, the last column in Table 1 includes the total number of the equally optimal factors computed during the whole computation divided by the final number of factors delivered by the GreConD algorithm. Let us also note that this characteristics is not mentioned in any previous work.

4.2 Quality of Decomposition

We provide results on the most important aspect of evaluation of the performance of BMF algorithms—the quality of decomposition delivered by an algorithm (recall Section 2). As common in the literature on BMF, we evaluate the obtained results from the viewpoints of the two main BMF optimization problems: DBP (Discrete Basis Problem) and AFP (Approximate Factorization Problem), cf. the introduction section and Section 2.

DBP emphasizes the importance of the first few (presumably most important) factors. In this perspective, the quality of factors obtained by a BMF algorithm may be assessed by observing the values of coverage c for small numbers of factors.

AFP emphasizes the need to account for (and thus to explain) a prescribed (presumably reasonably large) portion of data. In this perspective, the quality of factors obtained by a BMF algorithm may be assessed by observing the numbers of factors needed to attain a prescribed coverage c.

4.2.1 Comparsion with GreConD Algorithm

We observe the values of $c(l)$ (see Section2) for $l = 0 \ldots k$, where k is the number of factors delivered by a particular algorithm. Clearly, for $l = 0$ (no factors, A and B are "empty") we have $c(l) = 0$. In accordance with general requirements on BMF, for a good factorization algorithm $c(l)$ should be increasing in l, should have relatively large values for small l (i.e. should be steeply increasing in the beginning), and it is desirable that for $l = k$ we have $I = A \circ B$, i.e. the data is fully explained by all k computed factors (in which case $c(l) = 1$).

The values of $c(l)$ (average over 1000 iterations) for synthetic data are shown in Figures 1, 2 and 3. In case of the GreConDP algorithm we present the values for the best factorization. We study similarities of particular factorizations delivered by GreConDP later in Section 4.3.

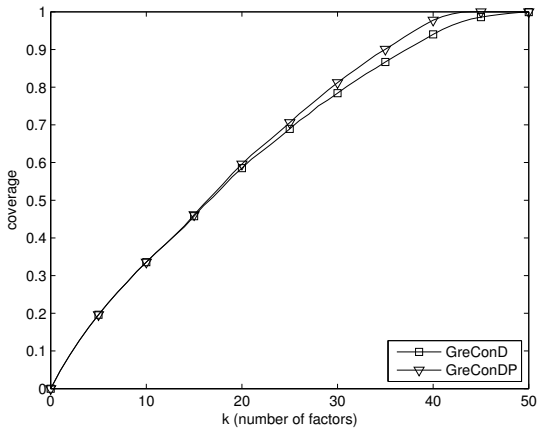


Figure 1: Comparison of GreConDP with GreConD on synthetic data, $P = 4$

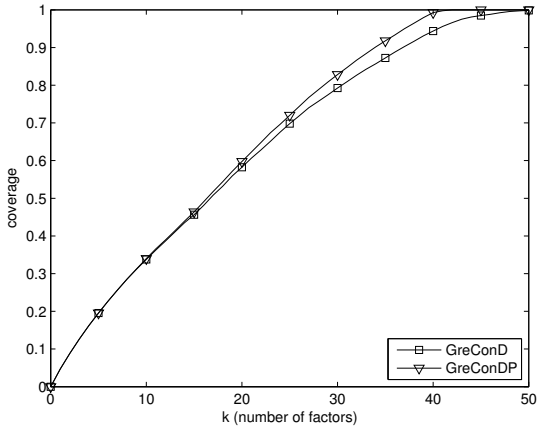


Figure 2: Comparison of GreConDP with GreConD on synthetic data, $P = 8$

[1] NOW public release 030717, available from http://www.helsinki.fi/science/now/

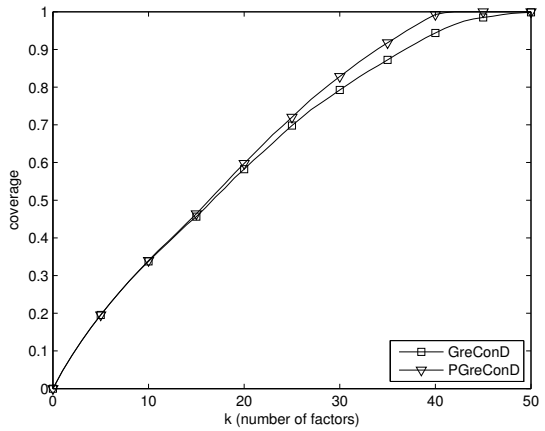


Figure 3: Comparison of GreConDP with GreConD on synthetic data, $P = 16$

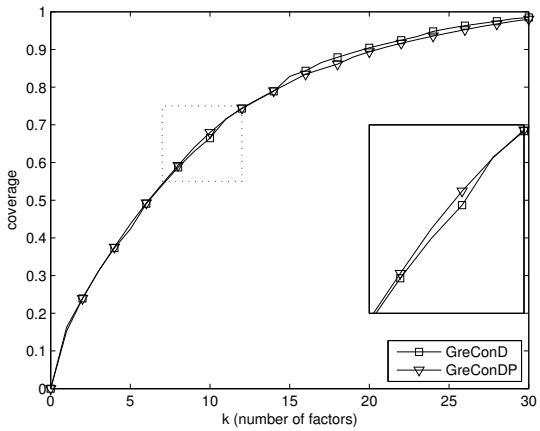


Figure 4: Comparison of GreConDP with GreConD on Emea dataset, $P = 4$

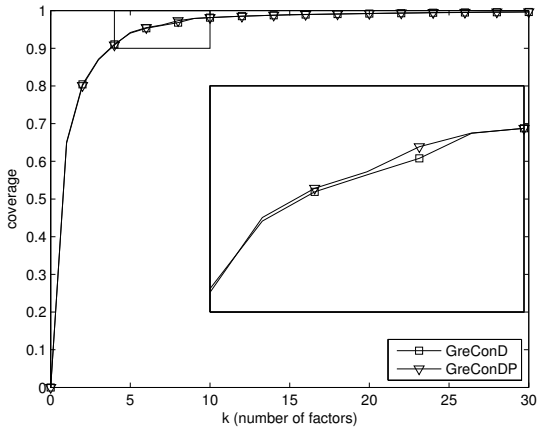


Figure 5: Comparison of GreConDP with GreConD on Firewall 1 dataset, $P = 4$

We can see that GreConDP considerably outperforms original GreConD algorithm, especially from the AFP point of view and slightly from the DBP point of view. Namely, the coverage values are higher for a given number of factors and the difference grows with the number of factors. Eventually, a full coverage of input data is obtained with less factors – let us note that the number of factors delivered by GreConDP is in most cases equal to the expected number of factors (40, see Section 4.1.1) and, moreover, the factors are the original factors used to generate the data. For small numbers of factors (values of $k < 10$), however, the difference between the two algorithms is slight. On the other hand, for increasing number P of processes this difference slowly increases (for $P = 1$, GreConDP produces the same results as GreConD).

For real datasets we obtain similar results, see Figures 4, 5, 6, 7 and 8. GreConDP outperforms GreConD on Mushroom, Paleo and Zoo datasets. Here, however, full coverage of data is obtained from both algorithms with the same number of factors but GreConDP gives higher coverage values for small values of k. In particular, this is quite notable on the Mushroom dataset, see the Figure 6. On Emea, GreConDP produces better results than GreConD from the DBP point of view but from the AFP point of view it is outperformed by GreConD. This is due to the fact that the average number of equally locally optimal factors per factor (see above) for this dataset is extremely high and the advantage of GreConDP over GreConD in utilizing more (but few compared to the number) of the equally optimal factors rather than just one vanishes. We also observed a similar behavior on Firewall 1 dataset. On the DBLP dataset, GreConDP produces exactly the same decompositions as GreConD so we do not include a graph for it. Let us also note that for this dataset we know the Boolean rank (19) and decompositions produced by both algorithms are optimal.

4.3 Similarity of Factorizations

As we saw in Section 3.2, the GreConDP algorithm produces P most optimal factorizations of the given input Boolean matrix (instead of just one) where P is the number of processes. Hence a natural question arises: How much are those factorizations similar to each other? For $P = 8$, we obtained for all datasets from Table 1 similar results like those shown in Table 2. The numbers in the table refer to the degree of similarity of two factorizations defined as follows: factorization F_i in a row of the table and factorization F_j in a column of the table are similar to degree $p \in [0, 1]$ if $p{\cdot}100$ percent of factors of F_i is also present (as the same factors) in F_j. Note that in this simple and easily interpretable (in general non-symmetric) similarity measure we do not consider indices of factors in the factorizations (i.e., in particular, if one of the factorizations is a permutation of the other one these are measured as equal).

We can see that the factorizations obtained by GreConDP are rather similar to each other (especially the best ones, denoted by F_i with index i close to 1). This is not surprising since the factor search strategy in the GreConD algorithm, which is included also in GreConDP, has its limits.

Moreover, in Figure 9 we can see the progress of the similarities of factorizations F_1, F_4 and F_7 of the Mushroom dataset for the number of the first factors going from 1 to 30. As we can see, the GreConDP algorithm starts with different factors (the factorizations are not similar at all) and with more factors the factorizations become more similar. That means that the algorithm finds the same factorizations in different ways. Similar results were obtained also for the others factorizations.

	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
F_1	1.000	0.991	0.991	0.991	0.983	0.983	0.991	0.983
F_2	0.991	1.000	0.991	0.983	0.991	0.983	0.983	0.991
F_3	0.991	0.991	1.000	0.983	0.983	0.991	0.983	0.983
F_4	0.991	0.983	0.983	1.000	0.991	0.991	0.991	0.983
F_5	0.983	0.991	0.983	0.991	1.000	0.991	0.983	0.991
F_6	0.983	0.983	0.991	0.991	0.991	1.000	0.983	0.983
F_7	0.991	0.983	0.983	0.991	0.983	0.983	1.000	0.991
F_8	0.983	0.991	0.983	0.983	0.991	0.983	0.991	1.000

Table 2: Similarity of the first eight factorizations of Mushroom dataset

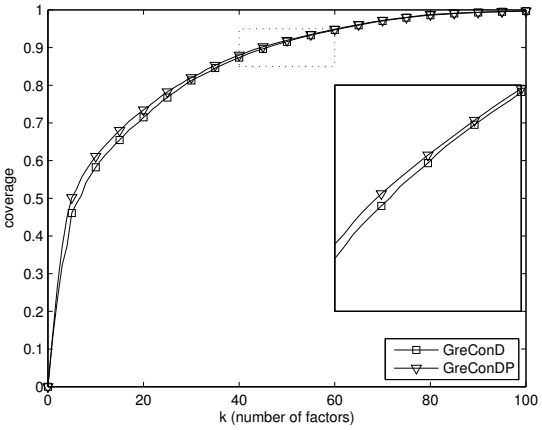


Figure 6: Comparison of GreConDP with GreConD on Mushroom dataset, $P = 4$

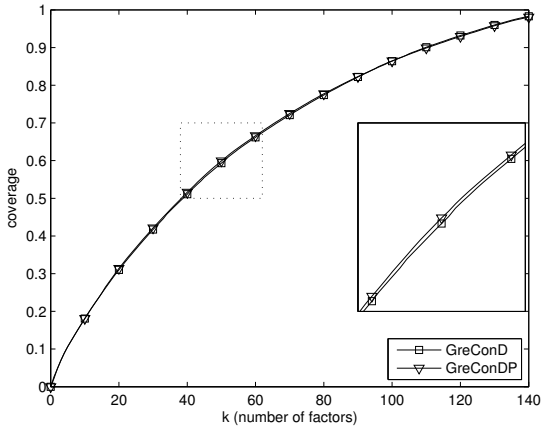


Figure 7: Comparison of GreConDP with GreConD on Paleo dataset, $P = 4$

4.4 Running Time

Usually, (theoretical asymptotic) time complexity of a BMF algorithm is not a primary concern in the BMF community, see e.g. (Belohlavek et al. 2015). Nevertheless, below we provide brief remarks on the observed running time of the GreConDP algorithm compared to the GreConD algorithm.

We implemented both GreConD and GreConDP in MATLAB. Critical parts (computing the operators $^{\uparrow}$ and $^{\downarrow}$, recall Section 3.1) were written in C and compiled to binary MEX files. For parallelization we used the Parallel Computing Toolbox MATLAB package. None of the algorithms was optimized for speed.

Despite that, each of the evaluated datasets (Table 1) was factorized by both algorithms, on an ordinary PC, in order of minutes[2]. The slowdown of GreConDP to GreConD depends on the number of processes run in GreConDP vs. the number of processor units. If we use less processes than we have processor units, GreConDP is only a slightly slower than GreConD, mainly due to the parallelization overhead (synchronization barriers and the critical section). If we use p times more processes than we have processor units, our observation is that GreConDP is approximately $p/2$ slower than GreConD.

5 Conclusions

In the paper we presented a general parallelization scheme for Boolean matrix factorization algorithms and also a new algorithm, called GreConDP, utilizing this scheme. The algorithm is based on one of the well established algorithms for Boolean matrix decomposition, the GreConD algorithm (Belohlavek et al. 2010). We chose GreConD as our base algorithm due to its relative simplicity and high efficiency but the proposed parallelization scheme can be applied to arbitrary sequential heuristic algorithm for Boolean matrix decomposition.

We evaluated properties and results delivered by GreConDP in various experiments involving synthetic (randomly generated) and real datasets. From presented results we can see that the algorithm outperforms in most cases the base algorithm, GreConD, most importantnly from the quality of decomposition standpoint, at almost none or moderate computing time expenses (depending the number of parallel runs/processes vs. the number of available processor units)—which were the objectives of our parallelization scheme. Namely, for the synthetic data, coverage produced by GreConDP is higher than coverage produced by GreConD for the same number of factors and in the end the data is fully explained by considerably less number of factors; for the real datasets, at least for those we used, the data is, however, fully explained by the same number of factors (and the final decompositions are very similar) but GreConDP produces higher coverage than GreConD for small numbers of factors in the beginnings of decomposition computation, provided the numbers of equally locally optimal factors in partial decomposition constructions are not very high (not significantly higher than the number of parallel runs)—only then the utilization of more equally optimal factors is beneficial. Moreover, as expected and intended, GreConDP tends to produce better results than GreConD with the increasing number of parallel runs. What was also expected, and has been observed, is that although producing rather similar

[2] An implementation of GreConD in C optimized for speed factorizes the datasets on an ordinary PC in order of seconds, see e.g. (Belohlavek et al. 2010).

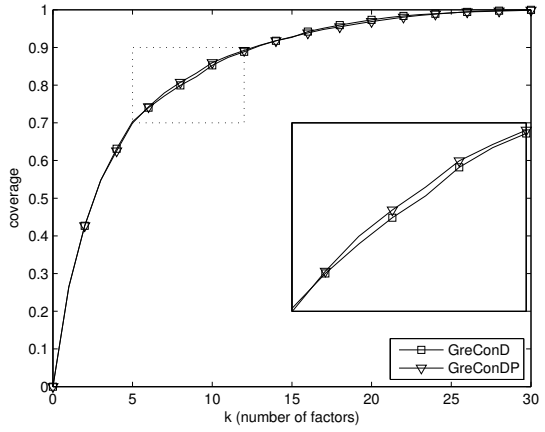


Figure 8: Comparison of GreConDP with GreConD on Zoo dataset, $P = 4$

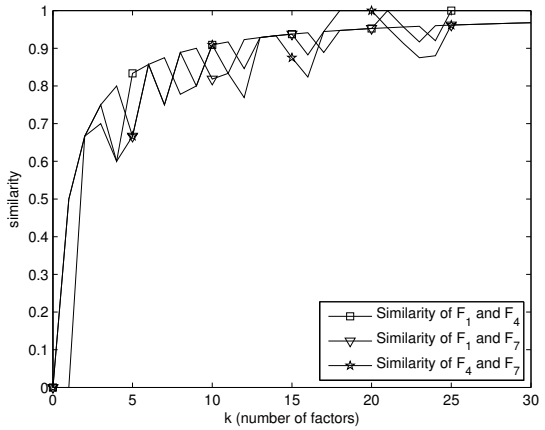


Figure 9: Progress of similarities of factorizations of Mushroom dataset for the first 30 factors

final decompositions (especially the best ones), the algorithm starts with different factors in the decompositions, i.e. the similar final decompositions are constructed in different ways.

The observed results encourage us to the following future research directions. First, apply the proposed general parallelization scheme to other BMF algorithms, especially to GreEss (Belohlavek et al. 2015) and Asso (Miettinen et al. 2008) which both involve a similar heuristic strategy like GreConD but in a different manner. Second, study the properties of the equally locally optimal factors—the problem which every heuristic algorithm faces—in order to find further ways leading to better factorizations.

References

Bache, K., Lichman, M. (2013), `http://archive.ics.uci.edu/ml`, University of California, School of Information and Computer Science, Irvine, CA.

Belohlavek, R. & Vychodil, V. (2010), Discovery of optimal factors in binary data via a novel method of matrix decomposition, *Journal of Computer and System Sciences* **76**(1), 3–20.

Belohlavek, R. & Trnecka, M. (2015), From-Below Approximations in Boolean Matrix Factorization: Geometry and New Algorithm, *Journal of Computer and System Sciences* **81**(8), pp 1678–1697.

Berry, M. W., Mezher, D., Philippe, B. & Sameh, A. (2006), Parallel Algorithms for the Singular Value Decomposition, *in* Kontoghiorghes, E. (ed.): 'Handbook on Parallel Computing and Statistics', Stat. Textb. Monogr., 184, Chapman & Hall/CRC, pp. 117–164.

Ene, A., Horne, W., Milosavljevic, N., Rao. P, Schreiber, R. & Tarjan, E. R. (2008), Fast exact and heuristic methods for role minimization problems, *in* 'ACM SACMAT Proceedings of the 13th ACM Symposium on Access Control Models and Technologies', pp. 1–10.

Ganter, B. & Wille, R. (1999), *Formal Concept Analysis: Mathematical Foundations*, Springer.

Geerts, F., Goethals, B., & Mielikäinen, T. (2004), Tiling databases, *in* 'Proc. Discovery Science', pp. 278–289.

Kannan, R., Ballard, G. & Park, H. (2016), A high-performance parallel algorithm for nonnegative matrix factorization, *in* 'Proc. 21st ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP'16)'.

Lucchese, L., Orlando, S., & Perego, R. (2010), Mining Top-K Patterns from Binary Datasets in presence of Noise, *in* 'SIAM ICDM International Conference on Data Mining', pp. 165–176.

Miettinen, P., Mielikinen, T., Gionis, A., Das, G. & Mannila, H. (2008), The Discrete Basis Problem, *IEEE Transactions on Knowledge and Data Engineering* **20**(10), pp 1348–1362.

Stockmeyer, L. (1975), The set basis problem is NP-complete, Tech. Rep. RC5431, IBM, Yorktown Heights, NY, USA.

Tatti, N., Mielikainen, T., Gionis, A. & Mannila, H. (2006), What is the Dimension of Your Binary Data?, *in* 'IEEE ICDM International Conference on Data Mining', pp. 603–612.

Xiang, Y., Jin, R., Fuhry, D. & Dragan, F. F. (2011), Summarizing transactional databases with overlapped hyperrectangles, *Data Mining and Knowledge Discovery* **23**(2), 215–251.

Factors influencing Australian teachers' intent to leave the teaching profession

Bo Cui[a] Alice Richardson[b]

[a]Department of Education and Training,
Queensland Government, 30 Mary Street, Brisbane 4001, Australia
[b]National Centre for Epidemiology & Population Health,
Australian National University, 62 Mills Rd, Acton 2601, Australia
bo.cui@dete.qld.gov.au.

Abstract

Teachers play a vital role in shaping the lives of our children. In Australia, the teaching work force is experiencing a teacher shortage especially in particular subject areas of science, technology, engineering and mathematics. Teacher retention rate is decided by teachers' outflow i.e. teachers permanently leaving the profession prior to retirement. This research uses the Staff in Australia's Schools 2010 data set as the data base to formulate a logistic regression model of teacher outflow, which enriches the quantitative research into Australian teaching work force planning. It addresses the teacher outflow issue by identifying what prominent factors would influence teachers' decision of leaving the profession. Factors that significantly affect Australian teachers' decision in terms of their intention to leave teaching profession are: teachers' satisfaction with student behaviour, salary, working relationship, and age. The analysis also has implications for the literature on school community and school effectiveness.

Keywords: logistic regression, teaching workforce, logit model, attrition, outflow

1 Introduction

Previous education research concluded that teaching is one of the primary drivers in improving student outcomes. From all over the world, it is challenging for governments to address the shortage of high quality teachers. Teacher attrition is highly correlated with teacher quality, because a high attrition rate indicates a high probability of skilled teachers leaving the teaching profession prior to their retirement (van Geffen & Poell 2014). It has been noted that current teacher shortages in Australia in mathematics and science potentially undermine student achievement in these key subjects. Also for at least a decade the Australian Government is concerned about the decreasing number of male teachers especially in primary schools (Ruddock, 2004). School staffing problems are primarily due to the excessively large number of qualified teachers who depart the sector permanently prior to their retirement (Ingersoll 2001). The responsibility for maintaining the day-to-day staffing requirements of schools, particularly in a climate of teacher shortages, lies with education authorities whose decision on particular initiatives to tackle the teacher shortage issue is built on evidence based policy formulation. Thus accurate understanding and prediction of teacher attrition will help to improve government strategies to deal with the teacher shortage problem.

The purpose of this research is to use logistic regression to estimate Australian teachers' intent to leave teaching profession, and describe the factors (or independent variables) that could precisely predict teachers' intention to permanently leaving teaching profession prior to retirement. Similar models have been applied to earlier waves of the SiAS (Pacific Analytics 2000), in predicting cigarette use (Adwere-Boamah 2010), the nursing workforce (Ujvarine 2011), and even to the task of predicting Academy Award winning movies (Pardoe 2005).

2 Literature Review and Data

Informed by the literature, three blocks of independent variables were identified as being potential contributors to teachers' attrition. They are teacher characteristics, school characteristics and organisational conditions (Dupriez, Delvaux, & Lothaire, 2016; Ingersoll, 2001; van Geffen & Poell, 2014). According to Hancock and Scherff (2010), working load, salary and student behaviour are major causes of teacher attrition. It is also worthwhile to note that teachers who have completed the most advanced studies are the most mobile (Dupriez et al., 2016), thus the logistic regression model will comprise the variables in Table 1 as predictors. It is worth noting that instead of using traditional models such as logistic regression, education related data mining could potentially apply to the modelling of teacher attrition, using techniques from learning analytics and psychometric analysis (Pearson and Moomaw 2005, Pistilli and Arnold 2012).

Staff in Australia's Schools (SiAS) 2010 is the second national survey with responses from 14,535 teachers and school leaders, funded by the Australian Government, and conducted by the Australian Council for Educational Research (ACER). The first survey was conducted in 2006-07. SiAS collected data on a wide range of teacher characteristics and workforce issues including: demographic items, professional learning, qualification, future career intention, and career path. One of the major

purposes of this national survey is to provide relevant data to inform teacher staffing issues and teacher workforce planning (McKenzie et al. 2011). Data on opinions and perceptions from SiAS are self-reported, so are subjective.

In this research 'Planning to leave teaching permanently prior to retirement' will be the response variable, recoded into a binary variable with 1 indicating the teacher will intend to leave versus 0 indicating the teacher will intend to stay. In the SiAS sample 24.7% teachers intend to leave. Predictor variables and their distribution are shown in Table 1.

Teacher Characteristics	
Age (years)	16.5% under 30, 51.6% 30-50, 31.8% 50+
ATSI origin	0.8% ATSI, 99.2% non-ATSI
Gender	69.1% female, 30.9% male
Highest qualification completed in Education	59.2% Undergraduate or vocational education/TAFE, 39.2% PG,1.6% others
School Characteristics	
ATSI Focus School	13.2% ATSI, 86.8% non-ATSI
Sector	16.1% Independent, 58.2% Government, 25.6% Catholic
Sector of first school	7.9% Independent, 74.9% Government, 17.2% Catholic
Years in first school	33.9% less than or equal to 1 year, 40.2% between 1 and 5 years, 25.9% 5 years and more
Organizational conditions	
Satisfaction with salary	8.9% Very Dissatisfied (VD), 28.7% Dissatisfied (D), 54.1% Satisfied (S), 8.4% Very Satisfied (VS)
Satisfaction with working relationships with colleagues	0.9% VD, 4.3% D, 51.6% S, 43.3% VS
Satisfaction with student behaviour	9.5% VD, 23.5% D, 52.5% S, 14.5% VS
Hours spent per week in face-to-face teaching	60.8% less than 20 hours, 39.2% 20 hours and more
Current employment as a teacher full or part-time	20.7% PT , 79.3% FT

Table 1. Descriptive Statistics for Predictor Variables

3 Method

A simple diagnostic check for multicollinearity was carried out using the correlation matrix. Due to the high correlation between Age and Years of experience (r = 0.82), Years of experience was omitted from consideration.

Pallant (2005) emphasizes paying close attention to the outliers. Confidence interval displacement (CBar) is a useful indicator to locate abnormal observations which are potential outliers and have an influential effect on the overall parameter estimates. In order to detect potential outliers, we adopted the suggestion from Peng & So (2002), which is to plot confidence interval displacement (CBar) against observations would reveal observations that exercise a large influence over parameter estimates. There are 173 absolute values of CBar greater than 1 in the SiAS data set, so the 173 observations corresponding to the outstanding CBar values were excluded before performing the final logistic model analysis.

Outliers were identified by checking if the confidence interval displacement exceeds 1 in absolute value, or if Pearson residuals or deviance residuals were greater than ±2 in magnitude (Zelterman, 2010). The magnitude of the regression coefficients change when individual observations are excluded while model is refitted was also used to identify influential observations (Zelterman, 2010).

SAS® software version 9.1 (SAS Institute 2011) was used to perform data analysis through logistic regression analysis. A logistic regression model was used to predict the chance of the binary outcome based on individual characteristics by obtaining the odds ratio (Sperandei, 2014). The stepwise method of variable selection was also used (Tabachnick, 1996, p.150).

4 Result

Under the stepwise selection procedure, odds ratio estimates and 95% confidence intervals for the variables which entered and stayed in the regression model are listed in Table 2.

Effect	Point Estimate	95% Wald Confidence limits	
Teacher characteristics			
Age (RC = 50+ years)			
less than 30 years	2.711*	2.518	2.918
30-50 years	1.222*	1.167	1.279
Indigenous status (RC = non-ATSI)			
ATSI	0.269 *	0.192	0.377
Gender (RC = male)			
Female	0.644*	0.615	0.675
Qualification (RC = Bachelors)			
Post-graduate	1.392*	1.348	1.438

School characteristics			
ATSI Focus school	1.304*	1.236	1.376
Sector (RC = independent)			
Government	0.743*	0.720	0.766
Catholic	1.245*	1.203	1.289
First school sector (RC = independent)			
Government	0.937*	0.882	0.995
Catholic	1.047	0.979	1.120
Years in first school (continuous variable)	0.979*	0.974	0.983
Organisational characteristics			
Salary (RC = very dissatisfied)			
very satisfied	0.398*	0.366	0.433
satisfied	0.563*	0.533	0.594
dissatisfied	0.786*	0.743	0.830
Colleagues (RC = very dissatisfied)			
very satisfied	0.297*	0.248	0.356
satisfied	0.398*	0.333	0.476
dissatisfied	0.760*	0.628	0.919
Student behaviour (RC = very dissatisfied)			
very satisfied	0.434*	0.406	0.464
satisfied	0.503*	0.478	0.530
dissatisfied	0.858*	0.812	0.905
Hours worked (continuous variable)	0.992*	0.990	0.994
Employment (RC = Full time)			
Part time	0.754	0.724	0.786
Interactions			
Age*Sector(RC = 50+years & Independent)			
<30 & Government	2.157*	2.059	2.258
<30 & Catholic	2.611*	2.477	2.751
30-50 & Government	0.858*	0.831	0.885
30-50 & Catholic	1.400*	1.347	1.449
Age*Sex (RC = 50+ years & Male)			
<30 & Female	2.400*	2.215	2.602
30-50 & Female	1.071*	1.017	1.128

Table 2. Odds Ratio Estimates (* denotes $p < 0.05$, RC refers to reference category)

For the overall model fitting, the Receiver Operating Characteristic (ROC) Curve (Figure 1), plots the sensitivity (the proportion of true positives) versus 1-specificity (the proportion of true negatives). It indicates the ability of the model to discriminate (Hosmer & Lemeshow 2000), and was chosen as the global logistic regression performance indicator. The area under ROC curve was 0.6829, which means the model correctly classified 68.29% of the teachers' intent to leave. Thus the logistic model can be regarded as an acceptable instrument to predict Australian teachers' Intent to leave teaching profession prediction.

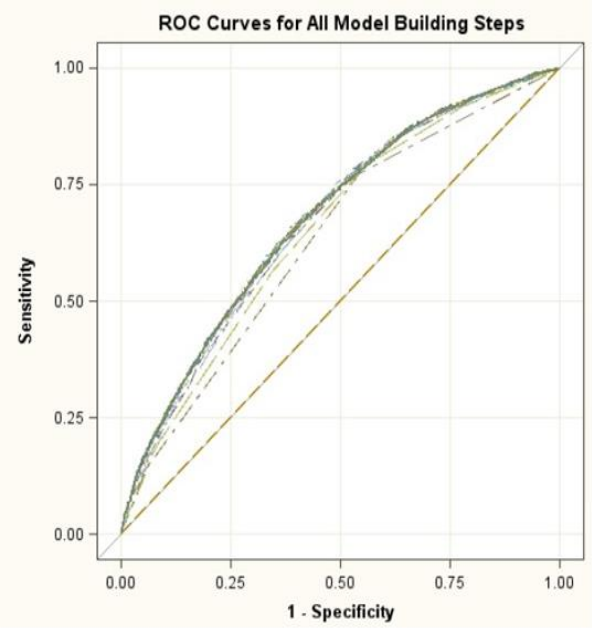


Figure 1. Receiver Operating Characteristic (ROC) curves for the logistic model of teachers' intent to leave.

Firstly, in terms of teacher characteristics, female teachers are 36% less likely to leave compared to male teachers, which is in line with the current serious male teacher shortage issue in Australian primary schools. Younger teachers (less than 30 years old) are nearly 3 times more likely to leave the teaching profession compared to more than 50+ years old teachers. Middle age teachers (30-50 years old) are about 20% more likely to leave compared to more than 50+ years old teachers. An Indigenous teacher has only about a quarter of the probability of leaving compared to non-Indigenous teachers. Teachers with postgraduate qualifications will be around 39.2% more likely to leave compared to those with bachelor degrees.

Secondly, in terms of school characteristics, a teacher from an Aboriginal and Torres Strait Islander (ATSI) focus school will be 30% more likely to leave teaching profession than a teacher from non-ATSI focus school. If the teacher's first school is in government sector, the teacher has slightly lower (around 7% less) chance to leave compared to the teacher's first school is in independent sector. If the teacher's first school is in the Catholic sector, the teacher has slightly higher, though not statistically significant, chance (4% more) to leave compared to if the teacher's first school is in independent sector.

Finally, in terms of organisational characteristics, full time teachers are about 25% less likely to leave the teaching profession than part time teachers. If a teacher is very satisfied with the remuneration, it is around 40% less likely the teacher will leave compared to the teacher who is very unsatisfied with the remuneration. If a teacher is very satisfied with the working relationships, about 70% less likely the teacher will leave compared to the teacher who is very unsatisfied with the working relationships. If a teacher is very satisfied with the students' behaviour, will be nearly 60% less likely to leave compared to the teacher who is very unsatisfied with his or her students' behaviour.

Hilbe (2001) stated 'interactions play an important role in modelling', so we used a likelihood ratio test to identify two significant interactions in the logistic regression model: age and gender, and age and sector.

In the age and sector interaction, holding other variables at the baseline, when teachers are less than 30 years old and working in government school, has about twice the chance of leaving the teaching profession over and above the effects of age and sector alone, compared to teachers who are more than 50 years old and working in independent schools. If teachers are between 30 to 50 years old and working in government schools, they are 15% less likely to leave the teaching profession below the effects of age and sector alone, compared to teachers who are more than 50 years old and working in independent schools. If teachers are between 30 to 50 years old and working in Catholic schools, they are 40% more likely to leave the teaching profession compared to teachers who are more than 50 years old and working in independent schools. Teachers who are working in Catholic schools and less than 30 years old, are 1.5 times more likely to leave the profession than independent schools' close to retiring age teachers.

The age-gender interaction indicates that female teachers who are less than 30 years old are 2.4 time more likely to leave the profession over and above the effects of age and gender alone, compared to more than 50+ years old male teachers. For 30 to 50 years old female teachers, they are 1.07 times more likely to leave over and above the effects of age and gender alone, compared to 50+ years old male teachers.

5 Discussion

According to the interaction of the predictors (Table 2), age-gender and age-sector interactions have a noticeable modification effect in regards to teachers' intention to leave.

In the big data era, it is not unusual that datasets contain thousands of observations, like the SiAS study, and it maybe not appropriate to use traditional modelling techniques due to the computational feasibility. In that case, we may need to consider more computationally intensive approaches such as random forests (Breiman, 2001). It should also be noted that research interest often lies in the estimation of population level relationship between independent and dependent variables in observational study. We based our analysis on unweighted survey data, so it is difficult to obtain population estimates without further information on the representativeness of the sample (Chen, 2015).

There could be a need to further explore confounding factors (with effects on both response and independent variables) which would influence teachers' intention to leave teaching profession prior to retirement, and include them in the model. Possible confounders could be government policy, such as strategies to reduce the current teacher shortage problem in regional areas of each state/territory; and strategies to reduce teacher shortage in particular subject areas, such as mathematics. Also a potential confounder could be the social environment, such as how respected the teaching profession is in the view of the general public.

6 Conclusion

We observe that teachers' intention to leave can be successfully predicted by the variables: age, Indigenous status, gender, qualification, whether in a ATSI focused school, school sector, first school sector, how long in first school, employment status, satisfaction with salary/student behaviour/colleagues, and workload. The multiple logistic regression model correctly classified around 70% of the cases, which is an effective approach to predict teachers' intention to leave teaching profession prior to retirement. The key findings are: that the teachers more likely to leave are younger, non-Indigenous, holders of a post-graduate qualification, male and part-time. In terms of perceptions of the workplace, teachers who are very unsatisfied with the salary, working relationships and student behaviour are also more likely to leave.

These findings are informative and beneficial for policy makers to formulate relevant evidence based teacher retention strategy and policy to address teaching staffing issues.

7 Reference

Adwere-Boamah, J. 2010. Multiple logistic regression analysis of cigarette use among high school students. *Journal of Case Studies in Education* **1**, 1 – 7.

Breiman, L. (2001). Random forests. *Machine Learning* **45**: 5 – 32.

Chen, B., Zhou, X.H., & Chan, G. (2015). Pseudoempirical-likelihood-based method using calibration for longitudinal data with dropout. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, **64**: 157-174.

Dupriez, V., Delvaux, B., & Lothaire, S. (2016). Teacher shortage and attrition: Why do they leave? *British Educational Research Journal*, **42**: 21-39.

Hancock, C. B., & Scherff, L. (2010). Who will stay and who will leave? Predicting secondary English teacher attrition risk. *Journal of Teacher Education*, **61**: 328-338.

Hastie, T. and Tibshirani, R. (1990) *Generalized Additive Models*. London: Chapman and Hall.

Hilbe, J.M. (2009). *Logistic Regression Models*. London: Chapman & Hall.

Hosmer D.W & Lemeshow, S. (2000). *Applied logistic regression* (2ed). New York: Wiley.

Ingersoll, R. M. (2001). Teacher turnover and teacher shortages: an organizational analysis. *American Educational Research Journal*, **38**: 499-534.

McKenzie, P., Rowley, G., Weldon, P. & Murphy, M. (2011). *Staff in Australia's School 2010: Main Report on The Survey*. http://research.acer.edu.au/cgi/viewcontent.cgi?article=1013&context=tll_misc (Accessed 17 June 2016).

Pacific Analytics Inc. (2000), *Teacher Supply and Demand in Queensland 1981-2009 Main Report*. Education Queensland, Government of Queensland, Brisbane, Australia.

Pallant, J.F. (2005) *SPSS survival manual: a step by step guide to data analysis using SPSS for Windows*. Sydney, NSW: Allen & Unwin.

Pardoe, I. (2005). Just how predictable are the Oscars? *Chance* 120.7% PT , 79.3% FT 8: 32 – 39.

Pearson, L. C. and W. Moomaw (2005). "The relationship between teacher autonomy and stress, work satisfaction, empowerment, and professionalism. *Educational Research Quarterly* **29**: 37.

Peng , C.J. & So, T.H. (2002) Logistic regression analysis and reporting: a primer. *Understanding Statistics* **1**: 31-70.

Pistilli, M. and K. Arnold (2012). Course signals at Purdue: Using learning analytics to increase student success. 2nd International Conference on Learning Analytics and Knowledge, Vancouver, Canada.

Ruddock, P. (2004). Government moves to address male teacher decline. Media release 186/2004. Canberra: Attorney General's Department.

SAS Institute (2011). *SAS Version 9.3*. Cary, NC: SAS Institute.

Sperandei, S. (2014). Understanding logistic regression analysis. *Biochemia Medica*, **24**: 12–18.

Tabachnick, B.G. & Fidell, L.S. (1996) *Using Multivariate Statistics*. 3rd ed. New York: HarperCollins.

Ujvarine, A.S. (2011). Intent to stay in nursing: internal and external migration in Hungary. *Journal of Clinical Nursing* **20**: 882-891.

van Geffen, R. E., & Poell, R. F. (2014). Responding to teacher shortages: relationships among mobility experiences, attitudes, and intentions of Dutch teachers. *Asia-Pacific Journal of Teacher Education*, **42**: 275-290.

Zelterman, Daniel (2010). *Applied Linear Models with SAS.* Cambridge: Cambridge University Press.

Acknowledgements

The authors are grateful for support from the Australian Commonwealth Department of Education, Employment and Workplace Relations (DEEWR, currently as Department of Education) Teacher Quality and Workforce Data Branch; and staff members Paul Hunt, Margaret Banks, Catherine Quinn and Jan Febey for permission to access to the SiAS data set, and relevant government educational policy guidance.

Residential Redevelopment of Greyfield Suburbs: Determinants of Medium Scale Redevelopment

Graham Webster, Denny Meyer, Peter Newton, Steven Glackin
Faculty of Health, Arts and Design
Swinburne University of Technology
PO Box 218, Hawthorn 3122, Victoria
gswebstr@gmail.com, dmeyer@swin.edu.au, pnewton@swin.edu.au, sglackin@swin.edu.au

Abstract
Urban redevelopment is an important issue facing Australian cities, and medium scale housing redevelopment has a key contribution to make in creating environmentally and economically sustainable cities for the future. However, this study of Melbourne greyfield redevelopment projects finds this type of redevelopment in only 16% of 6677 Melbourne greyfield statistical areas (effectively small 'suburbs' called SA1's) between 2004 and 2012. Medium scale redevelopment comprises projects delivering between 5-20 additional dwellings per lot, typically requiring some form of lot assembly among adjacent property owners. This paper examines five potential SA1 (suburb-scale) determinants of medium scale redevelopment (distance from the CBD; level of public transport accessibility; percentage of lots with separate/detached dwellings; mean lot size and median sales price per m^2), and one lot determinant of medium scale redevelopment (lot size). A cluster analysis is used to define six groups of small SA1 suburbs with significant differences in regard to the above potential determinants of medium scale redevelopment. Significant differences are also found between these clusters in terms of their redevelopment potential index and their level of medium scale redevelopment. A multi-level analysis confirms that the probability of a medium scale redevelopment for individual lots within any suburb also differs significantly between these clusters, with the effect of lot size significantly lower for the two most important medium scale redevelopment clusters. The results are confirmed using a second multi-level analysis based on the five SA1 variables used to define redevelopment potential and to create the clusters.

Keywords: multi-level model; sample imbalance; nested data; weighting of rare cases; agglomerative clustering.

1 Introduction
Population growth in the world's big cities is providing a challenge for increasing the supply of housing in an environmentally and economically sustainable manner, whilst at the same time creating "liveable" neighbourhoods. This is certainly true of Melbourne, Australia, which has a reputation to uphold having topped the *Economist's* liveability rankings in 2015 for a fifth consecutive year, despite a population growth rate of 2.2% in the year 2013-2014 resulting in a population of 4.44 million people in June 2014. The supply of additional housing in an urban environment comes from low density housing redevelopment in the outer suburbs (greenfields redevelopment), brownfields redevelopment of unused or under-utilized commercial and industrial sites, and from the greyfield redevelopment of inner and middle ring suburbs that are "physically, technologically and environmentally failing and which represent under-capitalized real estate assets" (Newton & Glackin 2015). Greyfields redevelopment is favoured when it results in infill housing. However, the Melbourne target of 70% infill housing for all new residential redevelopment was missed by about 20% in 2013 and 2014 (Bolleter 2013).

In order to better understand the factors associated with medium scale redevelopment this study was initiated using lot redevelopment data for the period 2004 to 2012. The objective in this study was to consider both lot and SA1 (equivalent to a small suburb) characteristics, as possible determinants of medium scale redevelopment, with the Property Developer Model (Rowley and Phibbs 2012, Rowley, Costello, Higgins and Phibbs 2014) and the Property Market Model (Charter Keck Cramer 2011a, 2011b) informing the choice of these characteristics. The first of these is a supply driven model and the second is a demand driven model.

The Property Developer Model stresses the financial attributes of redevelopments, suggesting that developers are only interested in projects that will produce a reasonable profit. The "property developer" model focuses on the individuals or organisations directly responsible for residential development projects, and proposes a model to explain the decision making process that occurs before a development project commences. Ultimately, the decision on whether or not a project proceeds is a financial decision. When there is stiff competition this means minimising costs. According to Charter Keck Cramer (2011a, 2011b), for a given type of dwelling construction, costs are generally the same throughout the Melbourne metropolitan area. This suggests that developers will favour lots with lower purchase prices. The "property developer" model

also suggests that the feasibility of a development project is assessed at the lot level. The first step in the process is lot identification. The physical and financial attributes of the property are an integral part of each assessment. Rowley observes that " developers assess many sites and and reject perhaps 95 percent of them" (Rowley & Phibbs, 2012, p.1). Purchase price needs to be low relative to surrounding areas and a square shape, a longer length of street access (frontage), corner lots, larger lots and level lots are all factors considered to be important by developers.

In contrast the Property Market Model recognises that the determinants of medium scale redevelopments may differ depending on whether the target purchaser is an investor or owner occupier, with investors more interested in the market values of their properties. Charter Keck Cramer (2011a, 2011b) estimates that investors purchase approximately 80% to 90% of new apartments in Melbourne, making the level of surrounding land and house prices the key determinant of medium scale redevelopment (MSR), because these support the sales prices of new medium scale redevelopments. Charter Keck Cramer (2011b, p10) state "while a range of factors will drive or inhibit development, from the developers' perspective, the sales rate per square metre (psm) will be the critical and fundamental driver for development in any location."

Another variable thought to influence redevelopment is the Redevelopment Potential Index (RPI), defined as the ratio of land value to property value. While land values appreciate over time, the value of dwellings generally depreciate. Charter Keck Cramer (2011a, 2011b) suggest that an RPI greater than 80% is indicative of a lot with redevelopment potential because the dwelling is no longer likely to be viable.

The data were sourced from various databases and then SA1 clustering and multi-level modelling were used to describe the probability of a medium as opposed to a small scale redevelopment project for any lot. Larger scale redevelopments and redevelopments involving no change in the number of dwellings or a reduction in the number of dwellings were excluded from the study.

It was expected that medium scale redevelopments would be found to be more common close to the central business district (CBD), and where the public transport access level (PTAL) was better. Medium scale redevelopments were expected to be less likely where the percentage of separate (detached) dwellings was higher or lot sizes were smaller. It was also expected that all these potential determinants of medium scale redevelopment would be strongly correlated. However, the effect of sales price/m^2 was unclear with the Property Developer Model suggesting that lower sales prices might promote MSR, while the Property Market Model suggested that higher sales prices/m^2 might promote MSR for the investor market. In addition, it was expected that the depreciated value of older housing would make the RPI a good predictor of MSR in place of a make-over.

However, before we start it is important to clarify why this paper belongs at a data mining conference rather than a statistical conference. With data for 46342 lots located in 6677 statistical areas and data files created using a fusion of multiple transactional data bases, the data challenges were considered sufficient for a data mining project. The use of a clustering algorithm to derive a classification of statistical areas using key redevelopment potential variables also speaks to data mining. However, the nested nature of the data, lots within SA1's, and the consequent use of multi-level modelling is unusual for a data mining conference, and our argument is that this must change. Big data is no excuse for an analysis that does not address the important data structures evident in the data.

In this study lots are nested within SA1's so one approach would be to model medium scale redevelopment within each SA1 using a different regression model for each SA1, with the coefficients for all the SA1's representing a sample of possible such values. This approach is incorporated in multi-level models, using a second level of equations to model the coefficients for each SA1 in terms of variables linked to medium scale redevelopment potential and captured for each of the SA1's.

Housing market segmentation together with multi-level modelling has been successfully applied in several studies in order to predict house prices, however, to the best of the authors' knowledge, this is the first study that has used this approach to address the issue of redevelopment potential. Most of the above house price studies rely on spatial definitions of submarkets, such as agent-based segments (Costello et al. 2012) or elementary school zones (Goodman and Thibodeau, 1998). However, Bourassa et al. (2003) used census data for population density and dwellings, homeownership rates, median household incomes, average rooms per house, ethnicity and percentage unemployed or receiving income support to define aspatial submarkets. They compared multi-level models for this aspatial definition of submarkets with a spatial definition used by government appraisers, finding that, in the case of house prices, the spatial definition produced better results.

This analysis compares two multi-level modelling approaches for modelling redevelopment potential. The first relies on a clustering of statistical areas using variables linked to medium scale redevelopment potential to define aspatial submarkets, while the second dispenses with the need for sub-markets, simply using the redevelopment potential variables themselves to describe the statistical areas. It is expected that the first of these methods will provide a superior multi-level model, due to the complex interplay between the variables being used to measure redevelopment potential, which should be better captured in a clustering of statistical areas than linear equations.

2 Methodology

2.1 Data and Data Sources

The only data collected for individual lots that is used in this study consists of the lot area prior to redevelopment and the classification of each redevelopment project as a small scale redevelopment (1-4 additional dwellings) or a medium scale redevelopment (5-20 additional dwellings). Both these variables were sourced from the Redevelopment Data Set, produced by Spatial Economics Pty Ltd as part of the Housing Development Data (HDD) database. Of the 81424 redevelopment projects undertaken between 2004 and 2012 only those in greyfield classified SA1's, resulting in an increase of between 1 and 20 dwellings, were

considered. A total of 46342 lots were considered within 6677 SA1 statistical areas.

SA1's are the smallest area units at which census data is released, with population numbers of between 200 and 800 persons and a density of over 200 persons per square kilometre. Several databases provided data at the SA1 level including the following. 1) The Victorian Property Transactions database provided data for the median sales price ($/m^2) for each SA1. These prices were adjusted to 2011 values and the median was chosen in order to avoid outlier effects. 2) Distances (kms) from each SA1 to the Melbourne Central Business District (CBD) were sourced from the Swinburne Institute of Social Research. 3) The Public Transport Accessibility Level (PTAL) for each SA1 was provided by SGS Economics and Planning. This index has a rating of between 0 and 10 with a value of 10 indicating closest proximity to the public transport network and the highest number, frequency and reliability of public transport services. 4) The percentage of dwellings that can be regarded as separate in each SA1 was provided by the Australian Bureau of Statistics. 5) Finally the Compiled Rates Database, as collated by the Victorian Valuer General and provided by the Victorian Department of Environment, Land, Water and Planning, was used to source the percentage of properties within each SA1 with a redevelopment potential index (RPI) of above 80%.

Outliers and severe skewness were demonstrated by the distributions for lot area and median sales price/m^2. In order to avoid model distortion these variables were therefore log transformed prior to analysis. A logit transformation was used for the proportion of MSR lots, for SA1's with at least some medium scale redevelopment. Table 1 shows the descriptive statistics for the raw SA1 data, while Table 2 shows the correlations between these variables after taking appropriate transformations. As distance to the CBD increases, average lot sizes increase, the level of access to public transport deteriorates, the percentage of separate dwellings increase, median sales prices/m^2 and the percentage of lots with an RPI greater than 80% decline. However, these correlations are generally much weaker than expected.

SA1 Variables (units)	Mean	Std.Dev.	Min.	Max.
Mean Lot Area (m^2)	1067	1541	60	33170
Distance to CBD (kms)	19.7	12.2	0.3	73.0
Percentage Separate Dwellings (%)	69.0	25.4	0.00	100.0
Median Sales Price ($/m^2)	1115	972	0.000	1180
Public Transport Access Level (PTAL)	3.22	2.27	0.00	10.00
% RPI > 80%	27.3	23.2	0.00	99.1
Proportion SA1's with MSR	0.188	0.391	0.000	1.000
Proportion MSR lots for SA1's with MSR	0.136	0.298	0.000	1.000

Table 1: Descriptive Statistics for SA1 Variables (N=6677)

SA1 Variables (units)	(1)	(2)	(3)	(4)	(5)
1.Mean Lot Area	1.00				
2.Distance to CBD	0.34	1.00			
3.% Separate Dwellings	0.13	0.36	1.00		
4.Median Sales Price ($/ m$^{2)}$	-0.28	-0.54	-0.25	1.00	
5.PTAL	-0.24	-0.61	-0.58	0.41	1.00
6.% RPI > 80%	-0.13	-0.43	0.04	0.24	0.22

Table 2: Correlations for SA1 Determinants of MSR

2.2 Clustering of SA1 Data

The relatively weak correlations with the CBD distance variable for several of the above variables suggest that a cluster analysis may be needed in order to define meaningful SA1 groupings. A hierarchical agglomerative clustering with standardized data and Ward's method was used to create six SA1 clusters on the basis of the first five of the variables in Table 1. These clusters were described in terms of their mean values and associations with the RPI variable were checked. The relationship of the clusters with evidence of medium scale redevelopment was then tested using SPSS version 23 software.

2.3 Multi-level Model for Medium Scale Lot Redevelopment

The dependent variable of interest in this study, the probability of a medium scale redevelopment, relates to individual lots but can also be explained in terms of the SA1 variables shown in Table 1. Some of the variation in the dependent variable is therefore associated with the within SA1 variation while the remainder is associated with the between SA1 variation. When the between variation proportion exceeds 5% of the total variation it is acknowledged that only models that allow for this hierarchical variance structure should be used (Snijders and Bosker, 2012; Raudenbush and Bryk, 2002). In this study it was found that roughly 34% of the variation in the type of redevelopment for any lot could be described as between SA1 variation, suggesting that it is essential for hierarchical modelling to be employed in this study. HLM7 software (SSI Central, HLM Version 7.01) was chosen for this purpose because this software 1) can accommodate a Bernoulli distribution for the dependent variable using a logistic link function 2) allows for weighting as described below and 3) facilitates moderation tests for the SA1 variables. Maximum likelihood estimates converged in all cases.

The model describing the probability (p) for a medium scale redevelopment for a lot was fitted using the equation

$$\ln(p/(1-p))=\beta_0+\beta_1\ln(\text{LotSize}) \quad (1)$$

In this model the variable ln(LotSize) was group centred. This meant subtraction of the mean value for the ln(LotSize) variable for the SA1 in which each lot resided. This is important because it means that the influence of LotSize is being described within SA1's in equation (1).

In the next stage of the modelling linear functions to describe the intercept coefficient, β_0, and the slope coefficient, β_1, were developed in terms of indicator variables representing the clusters. The cluster with the smallest CBD distance was chosen as the reference cluster. This multi-level model was then compared with a multi-level model in which the intercept and slope coefficients were modelled in terms of linear functions of the variables used to construct the clusters, rather than the clusters themselves. Grand centring was applied for all these SA1 variables. Both these models were fitted using full maximum likelihood estimation.

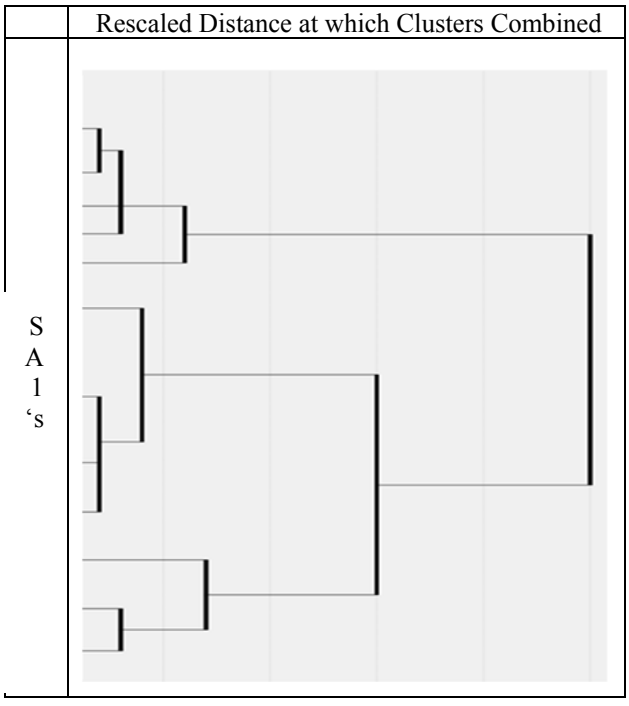
Only 4% of the lots that were redeveloped were classified as median scale redevelopments, indicating that this was a rare event in contrast to small scale redevelopment. Imbalance on this scale has implications for model fitting which cannot be ignored. Common remedies for imbalance in the response variable for a model are under-sampling of common cases, or using a cost matrix to penalise errors involving misclassification of the rare cases (type 1 error) more than errors involving misclassification of more common cases (type 2 error). However, the most popular remedy is over-sampling of rare cases and this is easily done by applying a higher weight to every rare case (Linoff and Berry, 2011; Witten et al., 2011). This approach has been followed in this study with a weight of ten applied to all medium scale redevelopment projects and a weight of one for all small scale redevelopment projects.

3 Results

3.1 Clustering of SA1 Data

Fig. 1 shows the dendrogram obtained from the hierarchical clustering. A choice of six clusters based on this dendrogram produced a suitable clustering for this analysis with cluster means shown in Table 3.

Fig. 1: Dendrogram for SA1 Cluster Analysis

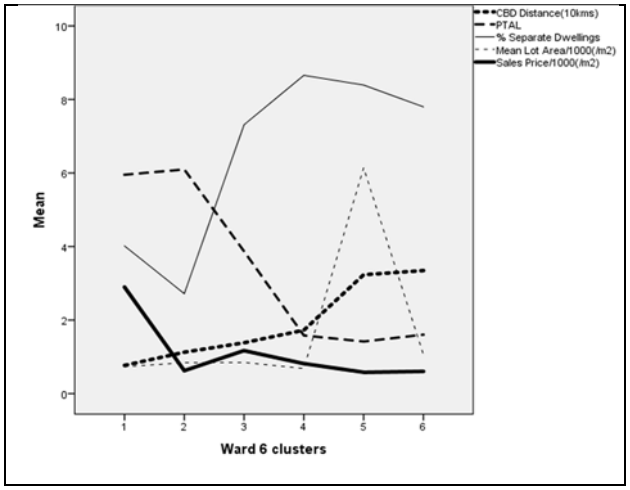


As shown in Table 3 and Fig. 2, the mean distance from the CBD increased from cluster 1 to 6. However, the other cluster characteristics did not follow the same pattern. In particular, only Clusters 1 and 2 had relatively high PTAL values and a relatively low percentage of separate dwellings, while clusters 4-6 had relatively low PTAL scores and a relatively high percentage of separate dwellings. Mean lot areas were relatively similar, except in the case of cluster 5, and although a slight downward trend was seen in median sales price/m^2 as mean distance from the CBD increased, cluster 2 had a distinctly lower median sales price/m^2 than was expected for a cluster so close to the CBD.

		Cluster					
		1	2	3	4	5	6
Variable							
Mean Lot Area (m^2)	M S	731 453	842 698	848 529	684 252	6141 5249	1060 737
Distance to CBD (kms)	M S	7.7 4.1	11.3 6.6	13.8 4.9	17.3 4.3	32.3 11.4	33.5 10.1
% Separate Dwellings	M S	40.2 20.1	27.2 17.3	73.1 14.7	86.6 91.1	83.9 15.0	78.0 22.2
Median Sales Price $/$m^2$	M S	2899 1452	624 315	1170 552	816 182	580 294	605 193
PTAL	M S	5.95 1.88	6.10 2.03	3.87 1.57	1.58 .086	1.42 1.15	1.60 1.12
% RPI > 80%	M S	27.8 15.0	21.2 13.7	43.3 20.1	31.8 29.4	11.1 13.1	11.8 15.2
% MSR lots for SA1's with MSR	M S	23.8 37.9	33.9 42.9	9.5 24.3	0.8 6.0	51.9 41.7	9.6 23.9
%SA1's with MSR		29.0	40.2	14.5	3.0	62.9	15.5
% SA1's		13.3	8.0	30.3	15.5	3.8	29.1

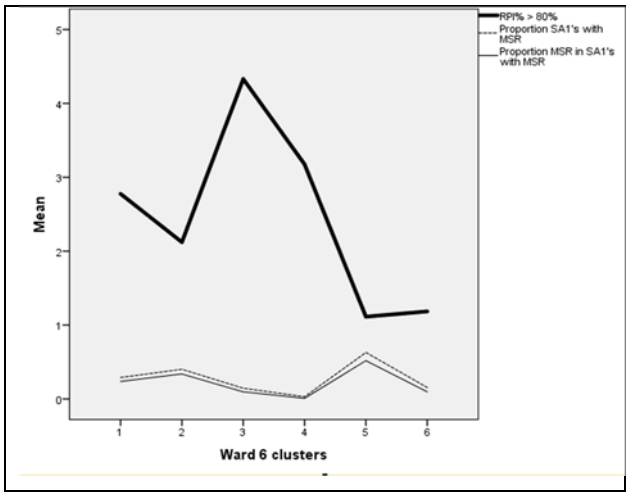
Table 3: Cluster Means(M) and Standard Deviations(S)

Fig. 2: Cluster Means by MSR Determinants



As shown in Table 3 and Fig. 3, the highest mean RPI percentages were for SA1's in cluster 3 with relatively low mean RPI percentages for clusters 5 and 6. However, contrary to expectation cluster 3 had a relatively low MSR rate for its SA1's and for lots within SA1's that had at least some MSR.

Fig. 3. Cluster Means for RPI and MSR variables



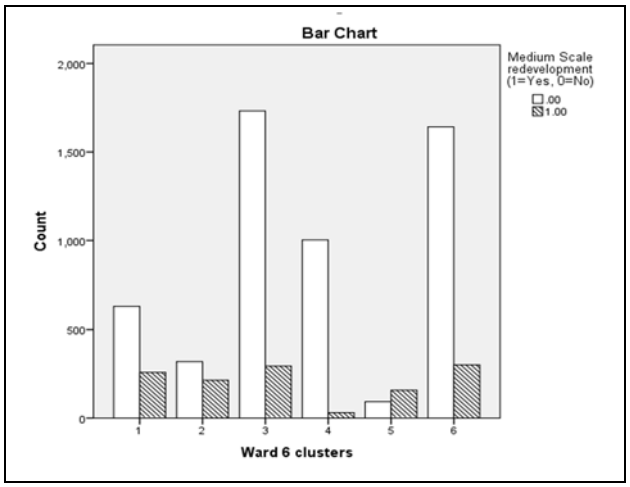
As shown in Table 4, General Linear Model Analyses with Student-Neuman-Keuls Post Hoc comparisons found significant differences ($p<.001$) between the clusters in terms of the SA1 variables used to create the clusters (CBD distance, Mean Lot Size, % separate dwellings, Median Sales Price/m^2, PTAL) with large η^2 effect sizes. In addition, there were significant differences between the clusters in terms of the RPI variable and the percentage of MSR's within SA1's where MSR occurred. A crosstab showed a significant relationship between the rate of MSR and the clusters with a large effect size (Cramer's V=.335, df=5), according to Cohen (1988).

Variable	η^2	Test Statistic	p-value
CBD Distance	0.66	F(5,6670) = 2586.0	<.001
Mean Lot Size	0.41	F(5,6670) = 929.6	<.001
Separate Dwelling %	0.52	F(5,6670) = 1455.5	<.001
Median Sales Price/m^2	0.60	F(5,6670) = 1983.7	<.001
PTAL	0.60	F(5,6670) = 1983.0	<.001
RPI > 80%	0.30	F(5,6670) = 580.0	<.001
% MSR lots for SA1's with MSR	0.27	F(5,1079) = 78.3	<.001

Table 4: Significance of Cluster Differences

Table 2 shows that on average 18.8% of the SA1's exhibited some medium scale redevelopment, however this percentage was much higher in the case of cluster 2 (40.2%) and cluster 5 (62.9%). Admittedly these are relatively small clusters, but they appear to have the winning formula for MSR. These clusters also had the highest rates for lot MSR, within SA1's having at least some MSR (i.e. 33.9% and 51.9% respectively). Fig. 4 illustrates this pattern of redevelopment across the six clusters

Fig.4. Medium scale redevelopment projects by cluster



The above results suggest that the clusters should be labelled as indicated in Table 5, moving from the CBD cluster for cluster 1 to the outer suburb clusters for cluster 5 and 6, with cluster 5 representing large lot size redevelopments and cluster 6 representing much smaller lot size redevelopments. However, the relatively high standard deviations for the CBD distances for some of the clusters confirms that there is some variation in terms of the locations of the SA1's for each cluster.

Fig. 5. Map of Melbourne with SA1 Clusters Shown

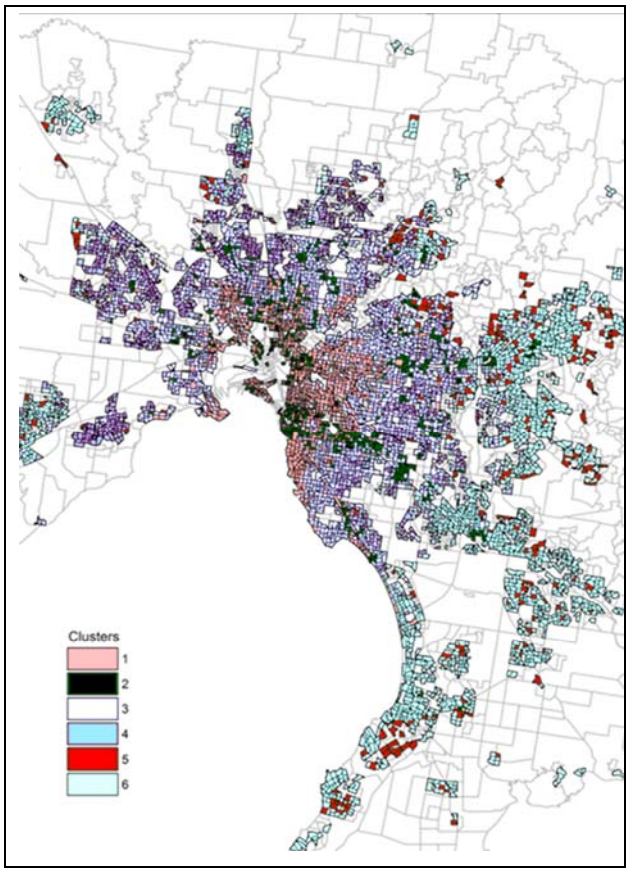


Fig. 5 confirms that the position of the SA1's for some clusters are widely dispersed. In particular, the SA1's for the important MSR cluster 2 (in black) have a wide dispersion of SA1's, mostly within both CBD and inner suburb areas. However, for the important MSR cluster 5, SA1's (in red) are found mostly in the outer suburbs, verging on greenfields redevelopment areas.

3.2 Model for the Probability of a Medium Scale Redevelopment (MSR)

In order to confirm the above results equation (1) was fitted to the data using a multi-level model, describing the probability of lot redevelopment in terms of lot size. An exploratory analysis was used to select the cluster indicator variables required to predict the coefficients β_0 and β_1 for each SA1. A weighting of 10 was applied to every lot with medium scale redevelopment, in order to ensure that the data imbalance (only 4% MSR projects) would not adversely affect the model fit. In this model Cluster 1, closest to the CBD, was treated as a reference to which the other clusters were compared.

Cluster	β_0	Odds Ratio (OR)	95% CI OR	β_1
1.CBD Cluster	.193	Ref.		1.739**
2.Lower Price Outer CBD	.659**	1.93	(1.54,2.42)	-.051
3.Inner Suburbs	-1.872**	.15	(.12,.20)	1.330**
4.Middle Suburbs	-4.625**	.009	(.005,.019)	1.973**
5.Large Lot Outer Suburbs	.572**	1.77	(1.35,2.33)	.490
6.Small Lot Outer Suburbs	-1.956**	.14	(.11,.18)	1.043**

(Ref. = reference; CI=confidence interval; * p<.01; ** p<.001)

Table 5. Estimated SA1 Cluster Model

Table 5 suggests that the odds of a lot being redeveloped as a MSR is on average 1.93 times more likely in the case of cluster 2 (Lower Price Outer CBD Cluster) and 1.77 times more likely in the case of cluster 5 (Large lot Outer Suburb) than is the case for a lot in cluster 1 (the CBD cluster). However, the MSR redevelopment of a lot in cluster 3, 4 or 6 is very unlikely compared to a lot in the CBD cluster. Interestingly the slope (β_1) estimates show that the effect of lot size on the odds of a MSR redevelopment is not significant for clusters 2 and 5, although significant for the other clusters. This suggests that for these two clusters of SA1's, MSR is not as sensitive to lot size as is the case for the other clusters where a larger lot size is associated with a much higher probability of redevelopment.

Table 6 shows the results when the multi-level model was fitted using the SA1 redevelopment potential variables themselves rather than the clusters derived from these variables. Importantly it appears that the median sales price is not a significant predictor of MSR. However, as expected, the odds of redevelopment increase in SA1'a with larger mean lot sizes and where public transport is better, while the odds of redevelopment decline for SA1's further away from the CBD and with a higher proportion of separate dwellings. However, most surprising, SA1's with a higher percentage of properties with a Redevelopment Potential Index of above 80% had lower odds for redevelopment. The signs of the estimated β_1 coefficients suggest that the effect of lot size on MSR is more positive in SA1's with larger mean lot size, and more negative in SA1's with better public transport and a higher percentage of separate dwellings.

SA1 Variables	β_0	Odds Ratio (OR)	95% CI OR	β_1
Intercept	-2.05**			2.65**
Mean ln(LotSize)	2.043**	7.71	(6.28,9.47)	1.067**
PTAL	.298**	1.35	(1.28,1.41)	-.212**
Distance CBD(kms)	-.341**	.71	(.64,.79)	-.134
% Separate Dwellings	-.330**	.72	(.69,.75)	-.102*
%RPI>80%	-.110**	.90	(.84,.95)	.030
Median ln(SalesPrice/m^2)	.046	1.05	(.90,1.22)	.018

(Ref. = reference; CI=confidence interval; * p<.01; ** p<.001)

Table 6. Estimated SA1 Variable Model

A comparison of the models fitted in Table 5 and Table 6 indicate that the Model in Table 6 has a significantly smaller error variance of 6.48 compared to 7.25 achieved for the Table 5 model. This suggests that the original supposition was incorrect. The clusters provide a less accurate way of modelling the redevelopment potential of SA1's than the variables used to construct these clusters.

4. Discussion and Conclusions

The results obtained from the comparison of the SA1 clusters and the multi-level modelling for individual medium scale lot redevelopment have provided similar results. Both these analyses have suggested that clusters 2 and 5 contain SA1's where medium scale redevelopment is most likely to occur. However, these clusters have very different characteristics.

Cluster 2 contains SA1's often located fairly close to the CBD, on average only 11.3 kms away from the CBD. Lot sizes are small on average but the actual lot size is not an inhibitor of MSR in these areas, indeed the coefficient for lot size is negative although not significant (β_1=-.051). On average, only 27% of the housing consists of separate dwellings in these SA1's, suggesting that apartment dwellings are the norm. Access to public transport is better for this cluster than any other (mean PTAL=6.10) but somewhat surprisingly the mean percentage of lots with RPI greater than 80% is relatively low at only 21%. In addition, median sales prices are low with an average of only \$624/m^2. This cluster appears to support the Property Developer Model for MSR in that cluster 2 developers are clearly favouring SA1's with cheaper purchase prices. This cluster does not provide support for the theory that higher RPI values will lead to MSR and, therefore, does not provide support for the Property Market Model. The SA1's in this cluster have prime investor characteristics in terms of PTAL and apartment living, however, it seems that sales prices are still relatively low. The MSR redevelopment of these areas is unlikely to return investors very high returns in the short term, suggesting that investors are taking a longer-term view in this case.

Cluster 5 contains SA1's quite far from the CBD, on average 32.3 kms away from the CBD. Lots are particularly large for these SA1's (6141 m^2 on average). Not surprising the actual lot size is not a significant factor for MSR in these areas (β_1=.490) because space is in such ample supply. These areas have a high percentage of separate dwellings (84% on average), and, not surprisingly, access to public transport is worse for these areas than for any other cluster (mean PTAL = 1.42). Purchase prices are also lower than for any other cluster, averaging only \$580/m^2, and the same is true for the RPI with on average only 11% of SA1's having average RPI's above 80%. Therefore, this cluster also seems to support the Property Developer Model for MSR in that developers are again seen to favour cheaper properties for MSR. It also does not provide support for the theory that higher RPI values will lead to MSR, perhaps because no demolition is needed in order to add dwellings when lot sizes are so large. However, this cluster differs fundamentally from cluster 2 in that its SA1's do not have the characteristics usually favoured by investors. The low PTAL values and high percentage of separate dwellings suggest that the new dwellings created by MSR in these areas will more probably be owner than renter occupied. This perhaps explains why the Property Market Model appears not to apply. Developers are catering for people determined to buy their own homes. These are often first time home buyers with strained budgets, making lower sales prices important. Although these developers are obviously motivated by profit to some extent, very high sales prices are not possible given the competition from other developers and the market they are serving.

The results for the model fitted in terms of SA1 redevelopment potential indicators, instead of the clusters based on these indicators, provides plenty of support for the above conclusions. Median sales prices failed to make any significant contribution to the model, confirming that developers are not pursuing redevelopment in SA1's with higher prices, although the attraction of lower prices is also not supported. This model also suggests that higher RPI values are not an incentive for MSR, in fact they appear to reduce MSR. However, this model clearly supports the view that redevelopment will be more attractive when public transport is better, lots are larger, distance from the CBD is smaller and the percentage of separate dwellings is smaller. Of course these are contradictory requirements in general, but they appear to be captured at least to some extent in clusters 2 and 5.

It was expected that the multi-level model based on the clusters would provide a more accurate model for redevelopment than a model based on the redevelopment potential variables themselves, because it was expected that the clusters would better capture interaction effects between these variables. However, the results have proved otherwise in that the error variance for the cluster model is nearly 8% larger than that obtained from the other model. This suggests that the clustering of submarkets is less important for redevelopment models than it is for house price models where spatial clusters have been found to be important components for modelling. However, ours was an aspatial clustering and it may be that a more spatially nuanced clustering would have been more successful. In addition, it may be that there are other important variables that need to be included in any spatial clustering, for example the presence of good schools within a SA1 may impact on redevelopment decisions.

In conclusion the data suggests that medium scale redevelopment in Melbourne does not follow the patterns suggested by the Property Market Model. A finer scale of data may be required for a market analysis of this nature. The current SA1 analysis may be too coarse to allow a rigorous test of the Property Market Model.

Support for the Property Development Model was found in that the clusters with the highest MSR had relatively low median prices, however, this variable was not a significant predictor of MSR in its own right, suggesting that additional testing of the Property Development Model is required. This conclusion also suggests that more variables are needed in order to capture the features of individual lots that appeal to developers. In particular, in addition to lot area, length of frontage, corner locations, shape and lot topography need to be included in any models used to predict the probability of MSR. This will also determine which of these variables are most important in the Property Development Model.

Finally, to return to the initial discussion regarding the nature of this project: data mining or statistical? The size and mix of data sources suggest that this is indeed a data mining project. A clustering of the SA1 parameters provided an interesting new picture of the city showing that there were pockets of development in the inner city and the outer city. However, in order to model the odds for MSR it was necessary to allow for the nested nature of the data (lots within SA1's) with a hierarchical linear model analysis. This suggests that a mix of data mining and statistical analysis was required in order to complete this project, and it is expected that for many projects this will be the case.

4 References

Newton, P. and Glackin, S. (2015): Regenerating Cities: Creating the Opportunity for Greyfield Precinct Infill Development, in *Instruments of Planning: Tensions and challenges for delivering more equitable and sustainable cities.* C. Legacy and R. Leshinsky (Eds) Routledge, London

Bolleter, J. (2013): Synergistic density: exploring the potential of correlating infill development with upgraded open space in greyfields suburbs. *Proc. 6th Making Cities Liveable Conference*. Melbourne.

Bourassa, S.C., Hoeslu, M. and Peng, V.S. (2003): Do housing submarket really matter? *Journal of Housing Economics*, 12, 12-28.

Charter Keck Cramer (2011a): *Urban renewal strategy – High Street and Plenty Road*, prepared for Darebin City Council & Department of Planning & Community Development Final Report 27 May 2011.

Charter Keck Cramer (2011b): *Market analysis for different types of housing in Darebin*, prepared for Darebin City Council & Department of Planning & Community Development Final Report 28 November 2011.

Cohen, J. (1988). *Statistical power and analysis for the behavioral sciences* (2nd ed.), Hillsdale, N.J., Lawrence Erlbaum Associates, Inc.

Costello, G., Leishman, C., Rowley, S. and Watkins, C. (2012): The Predictive Performance of Multi-Level Models of Housing Submarkets: A Comparative Analysis. *Proc. 18th Annual Pacific-Rim Real Estate Society Conference*, Adelaide, Australia.

Goodman, A.C. and Thibodeau, T.G. (1998): Housing Market Segmentation. *Journal of Housing Economics*, 7, 121-143.

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

Rowley, S. and Phibbs, P. (2012): *Delivering diverse and affordable housing on infill development sites. Melbourne, Australia*. Australian Housing and Urban Research Institute.

Rowley, S., Costello, G., Higgins, D. and Phibbs, P. (2014): *The financing of residential development in Australia. Melbourne, Australia*. Australian Housing and Urban Research Institute.

Snijders, T.A.B. and Bosker, R.J. (2012): *Multilevel Analysis : An Introduction to Basic and Advanced Multilevel Modeling*. Second edition. London: Sage Publishers.

Raudenbush, S.W. and Bryk, A.S. (2002): *Hierarchical linear models: Applications and data analysis methods*. London: Sage Publications.

Witten, I.H., Frank, E., Hallg, M.A. (2011): *Data Mining: Practical Machine Learning Tools and Techniques*, New York: Morgan Kaufmann Publishers.

A Temporal Classification based Predictive Model of Recurring Societal Events

Jie Chen[a], Wei Kang[a], Jiuyong Li[a], Jixue Liu[a] , Lin Liu[a], Brenton Cooper[b], Nick Lothian[b], Grant Osborne[b] and Terry Moschou[b]

[a] School of Information Technology and Mathematical Sciences
University of South Australia
Mawson Lakes, SA 5095
[b] Data to Decisions CRC (D2D CRC)
Base64, 64 North Terrace, Kent Town SA 5067

Email: firstname.lastname@unisa.edu.au, firstname.lastname@d2dcrc.com.au

Abstract

The ubiquitous nature of social media provides a range of ways for people to express their opinions and intentions against various societal issues much more conveniently, and provides a platform on which they can organise events such as protests. There is currently a lack of techniques that are able to predict these societal events in a timely manner. This paper proposes an innovative model for forecasting recurring societal events to foster new technical capability for the law enforcement agencies and other related public services. The sparse temporal event data is systematically investigated using a sliding window approach to formulate both a feature period and outcome period to capture the predictability of recurring events. The model with multiple classifiers was validated in two different measures to illustrate the effectiveness, including the state of the art metrics in the EMBERS system. The success of the model will help in research and development of the more challenging predictive models that utilising online open data sources to generate accurate predictions. The temporal-classification based model can also be applied to the prediction of events in other domains.

Keywords: societal events, classification, social media, gold standard report, temporal data mining.

1 Introduction

The relationship between a person's online and offline existence is becoming increasingly intertwined. For example, when the users are searching for information on flu, this may indicate there are possible infections in their family. The prevalence of social media and the huge increase in user-generated content create both tremendous challenges and opportunities for the law enforcement agencies and other related public services. When a government makes a change that a group of people perceive as negative (e.g., national research funding cuts) - disenfranchised individuals may take to social media to plan protest events or other social disruptions. When social media users are complaining about the government spending on national research organisations, there might be protest events being planned by the relevant communities in the near future.

Traditionally, Civil Unrest Events (CUE) are monitored by law enforcement agencies in different regions of a country under the risk assessments scenarios. The processes are largely adhoc, focus on events for their administrative regions and involve manual mechanisms to collect and summarise information. The paradigm is ineffective in dealing with the monitoring of some emerging populations, in a timely and efficiently way.

It has been realised that fast-growing adoption of social media applications provide a range of ways for people to express their opinions and intentions against various societal issues much more conveniently (Xu et al. 2014, Korkmaz 2014) using services such as Twitter, Facebook, Instagram, Tumblr and blogs. Therefore people are interested in apps/platforms that can analyse the information and opinions of the associated populations. For example, they are concerned about the discovery of trends of topics/hashtags, time series trends in a region or in regards to particular topics, and gaining insights into the precursors of domain-specific events through the tracking of the organisers' online accounts, and sentiments expressed in the social media contents.

In general, CUE in different locations and time periods may be related to different reasons and populations. A range of factors can trigger these events together or independently, including economic situations, natural disasters or human suffering, political scandals, the changes of policies of governments, call for actions by active social groups and so on (Hua et al. 2013). The information related to these factors may not be available in the databases of traditional law enforcement or intelligence agencies. The Beat the News (BTN) project of D2D CRC, Australia, intend to predict CUE in the interest of Australasian agencies with all available open data sources.

Historical CUE data is one of the most important data source in the building of predictive models. In this paper, we propose a temporal classification based model of recurring societal events. The model is inspired from the belief that that "History repeats itself". To our best knowledge, no research has been done on mining these spatio-temporal events data though there is research work

on extracting event database from news sources (Leetaru and Schrodt 2013).

2 Background

2.1 Gold Standard Report

The collection and quality control of historical CUE data is a non-trivial task. MITRE organises a gold standard report (GSR) of protests by surveying newspapers for reportings of civil unrest (OSI Team 2015). In this paper, we use the MITRE dataset which is compiled by an independent group comprised of social scientists and experts on Latin America. MITRE dataset identifies the instances of civil unrest events, when, where, who and why of such an event, i.e. the date of the event, its geographic location, the population protesting (e.g. labour, medical workers, general population) and the reason for the protest (i.e., the event type, e.g., economic, political, resource) (OSI Team 2015). 10 countries in MITER dataset are - Argentina, Brazil, Chile, Colombia, Ecuador, El Salvador, Mexico, Paraguay, Uruguay, and Venezuela. The quality of the dataset is controlled by the professional team in MITRE. Note that the labour intensive process of building GSR dataset has been eased by the tools developed in (Saraf and Ramakrishnan 2016).

Table 1. Civil Unrest Event Types

Event Code	Event Description
011x (city-level) 071x (widespread)	Employment and Wages
012x (city-level) 072x (widespread)	Housing and Shelter
013x (city-level) 073x (widespread)	Land, Energy, and Resources
014x (city-level) 074x (widespread)	Other Economic Issues
015x (city-level) 075x (widespread)	Other Government and Political Issues
016x (city-level) 076x (widespread)	Other Civil Unrest (including religious or culture marches, peace demonstrations, or any others)

Table 2. Civil Unrest Populations

Population
General Population
Business
Ethnic
Legal
Education
Religious
Media
Medical
Labour
Refugees/Displaced
Agricultural

As a concise description in Table 1, an event is represented with a 4-digits event code, where the first 3 digits are for the event type, e.g. 011x represents a Civil Unrest: Employment and Wages event (OSI Team 2015), and the fourth digit of the event code indicates violence (1 is non-violent and 2 is violent). Table 2 lists the populations of the events.

2.2 Characteristics of GSR

The MITRE GSR dataset contains all the CUE in the 10 Latin American countries between May 2013 and July 2014, which covers more than one year (14 months), with approximately 13k CUE events (OSI Team 2015). Figures 1 to 4 list the summary of the GSR dataset in the paper. The observations are listed as follows.

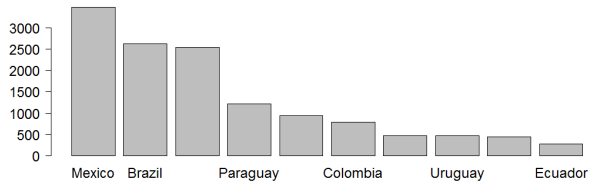


Figure 1. Distribution of country

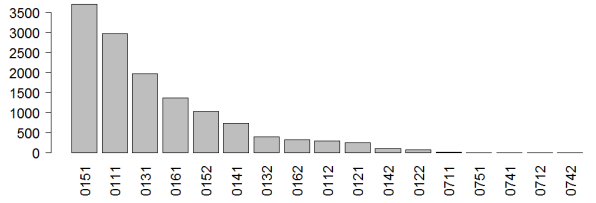


Figure 2. Distribution of event type

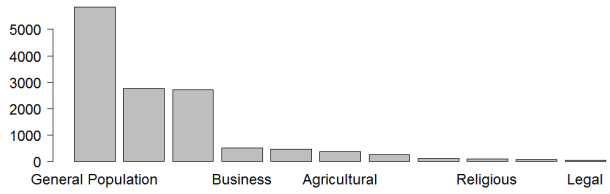


Figure 3. Distribution of population

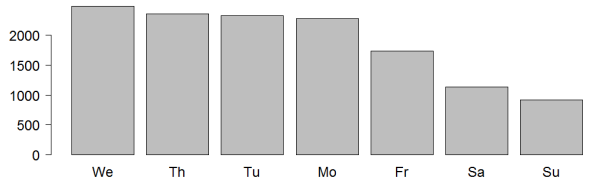


Figure 4. Distribution of day of the week

- Figure 1 shows that Mexico, Brazil and Venezuna are the three major countries with events in the dataset.

- Figure 2 shows that the top 3 event types are non-violent Other Government and Political Issues (0151), Employment and Wages(0111) and Land, Energy, and Resources(0131) in the dataset.
- Figure 3 shows that the top 3 populations are General Population, Labour and Education in the dataset.
- Figure 4 shows that the weekend days tend to have below average number of events in the dataset.

The locations of events are recorded at three levels: Country, State and City. There are more than a thousand unique city names for all the 10 countries. Given the number of days and the number of cities, it is not surprising the event data is usually very sparse for any specific location at the city level.

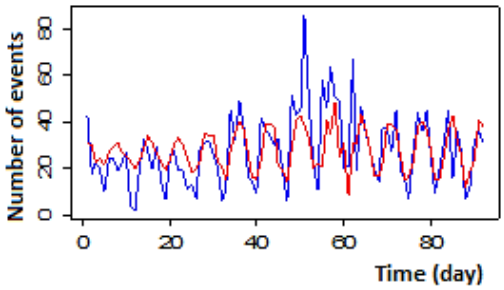


Figure 5. Daily time series of 10 countries and ARMA fitting

Weekly seasonality can be observed when the occurrences of the events are aggregated across 10 countries. Figure 5 shows the weekly seasonality in a three month period can be observed. Note that the curve is initially fitted with an ARMA model (Hyndman et al 2015) in R, fit.arima <− arima(Timeseries, order=c(3,0,3)), shown in red.

3 Related work

Temporal event datasets have been analysed with different data mining techniques, such as classifications (Chen et al. 2005) and associations (Roddick and Spiliopoulou 2002, Chen et al. 2010).

The formal compiling and generation events globally have progressed, e.g. GDELT [2] system (Leetaru and Schrodt 2013). However, these systems don't address the extremely important problem of early detection and forecasting societal events before they break out in reality.

Early events detection from open source data and news has been extensively studied in recent years (Compton et al. 2013, Agarwal and Sureka 2015, Muthiah et al. 2015).

Ramkrishnan et al. (2014) developed a maximum likelihood estimate baseline model using a historical database of protests in EMBERS (Early Model Based Event Recognition using Surrogates) system. The idea behind this model is that, even in absence of any explicit signal, the distribution of events in the recent past is a good guide for the civil unrest events that may happen in the future. The baseline model makes predictions on the basis of the distribution of event schema[3]— frequency in the most recent part of the GSR. In a typical 3-month interval, two-thirds of schemas appear once with the remaining third split evenly between those that appear twice, and those that appear three or more times. Predictions are generated with a minimum threshold of 2, and a 3-month training interval, and issued with a lead time of 2 weeks.

Similarly, Intelligence Advanced Research Projects Activity (IARPA) Open Source Indicators Program uses a similar Base Rate model (OSI Team 2015). These models are effective in EMBERS system (Ramkrishnan et al. 2014) because it can predict recurring events with the above simple logic but it only utilise the frequency of the historical events, ignoring other covariates, e.g. locations.

Korkmaz et al. (2015) build a baseline model in order to set a benchmark and to measure the added predictability provided by Twitter and other social media data. The baseline model uses no external input in the regression model. It uses lagged values of itself as the predictor. It only has a lead time of 1 day and the output of the model does not include any detailed information such as the location, event type or population involved. The intention is to forecast the first day of nationwide, relatively larger protests that span multiple locations in a country. They then propose a model using logistic regression models with Lasso to select a sparse feature set from diverse data sources.

In political sciences, Poisson regression models (King 1988) are limited because they assume events are independent and the models may suffer from the sparsity in GSR dataset.

4 Formulation of the temporal-classification based model

Inspired by the work by Ramkrishnan et al. (2014), we formulise the problem of predicting recurring societal events as a temporal classification problem. The basic idea is to partition the temporal event data with 3 months as feature period and 1 month as outcome period. The aggregated information of the events data in the two periods will form the feature space and classes vector. The aggregation should at least include location, event type and population so that it can embed the event data from the 3D space into a condensed feature space. Note that it is possible to consider the day of the week as well but our current experiments do not due to the sparsity of data. Consequently, a variety of classification algorithms

[2] http://gdeltproject.org/

[3] An event schema is a combination of a location, an event type, a population, and a day of the week.

can be integrated in training of the models and evaluated against the ground-truth data later.

Given the sparsity of the GSR events data discussed in Section 2 and the complexity of recurring characteristics of societal events, we propose to investigate different classifiers including Random Forest (RF), Decision Tree (DT), Support Vector Machine (SVM) and Naïve Bayes (NB) (Tan et al. 2005, Liu and Li 2005, Williams 2011) as candidate models across different sliding windows of the study period. The following formulates a list of concepts.

Definition 1. $\boldsymbol{F_i}$ is the feature matrix of the i-th feature period that cover 3 months of window, each row contains the feature values of the combination $< loc_j , type_k , pop_l >$.The dimensions of $\boldsymbol{F_i}$ is $\boldsymbol{N \times P}$, where $\boldsymbol{N}$ is the number of non-zero frequencies in the instances of the combination, $\boldsymbol{P}$ is the number of the feature variables to be considered.

Definition 2. $\boldsymbol{O_i}$ is the binary Outcome vector $(\boldsymbol{N \times 1})$ of the i-th training period in the outcome period, i.e. the next month of $\boldsymbol{F_i}$, the value 1 means that the instance of the schema $< loc_j , type_k , pop_l >$ has corresponding recurring events in the outcome period of the i-th training period, otherwise 0.

Definition 3. $\boldsymbol{C_i^m}$ is the m-th classifier that will be trained with the above input data $\boldsymbol{F_i}$ and $\boldsymbol{O_i}$.

Definition 4. $\boldsymbol{OP_i^m}$ is the prediction output vector $(\boldsymbol{N \times 1})$ from the trained $\boldsymbol{C_i^m}$ given the input $\boldsymbol{F_i}$. $\boldsymbol{OP_{i+1}^m}$ is the one from the trained $\boldsymbol{C_i^m}$ given the input $\boldsymbol{F_{i+1}}$ of the next feature period by increment of one month.

Definition 5. $\boldsymbol{GT_i}$ $\boldsymbol{and}$ $\boldsymbol{GT_{i+1}}$ are the ground truth vectors $(\boldsymbol{N \times 1})$ for the evaluation of $\boldsymbol{OP_i^m}$ and $\boldsymbol{OP_{i+1}^m}$ respectively. They will come from the same GSR dataset.

Algorithm 1. lists the pseudo code of the temporal classification based model to train and predict the recurring CUE.

Algorithm 1. Temporal Classification based Model for recurring events

---

Input: GSR dataset

Output: A list of classifiers $\boldsymbol{C_i^m}$ and its predictions $\boldsymbol{OP_i^m}$ and $\boldsymbol{OP_{i+1}^m}$;

1 Set the initial Window Size for feature and outcome periods, i.e. 3 and 1 month respectively, and determine the maximum number of sliding windows I;

2 Set the number of feature variables, and different types of classifiers

3 $i \leftarrow 1$;

4 **while** $i < I$ **do**

5 **for** each m:

6 Train the classifier $\boldsymbol{C_i^m}$ with $\boldsymbol{F_i}$ and $\boldsymbol{O_i}$;

7 Collect the trained classifier $\boldsymbol{C_i^m}$ and the predictions $\boldsymbol{OP_i^m}$ and $\boldsymbol{OP_{i+1}^m}$;

8 **end**

9 $i++$;

10 **end**

The rationale of this model can be explained with the statistical evidence shown in Figure 6. It simply plots the scatter plot of the frequency of a feature period against its outcome period fitted with a linear curve.

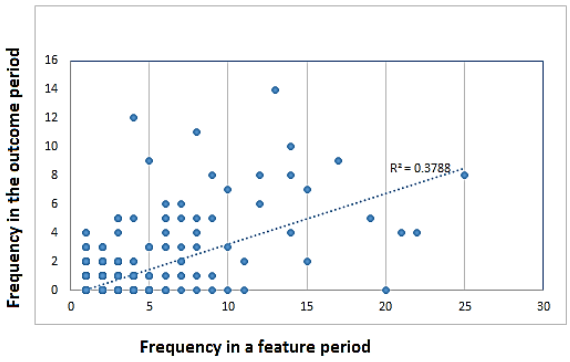


Figure 6. Correlation of counts for events between feature and outcome periods

Figure 6. illustrates that frequency in the feature period of a sliding window is the most effective feature for a classifier. In this paper, we introduce other features such as the event type and population, and especially a new binary feature "*CapitalCity*" to indicate the capital city of the 10 countries as shown in Table 3.

Table 3. Capital cities of the 10 countries

Country	Capital City
Argentina	Buenos Aires
Brazil	Brasília
Chile	Santiago
Colombia	Bogotá
Ecuador	Quito
El Salvador	San Salvador
Mexico	Ciudad de México
Paraguay	Asunción
Venezuela	Caracas
Uruguay	Montevideo

In the next section, we integrate a variety of classifiers and examine the effectiveness of the proposed model.

5 Experimental results

5.1 Evaluation

Using the temporal classification based model defined in Algorithm 1, we conducted a series of experiments using different combinations of our feature sets and classifiers and generated their evaluation results for each sliding window using OP_i^m,OP_{i+1}^m, GT_i and GT_{i+1}. All the four types of classifiers were trained using historical GSR data to predict the recurring of event with previous occurrences of events. Table 4 shows the means and standard deviations of F1-scores of different classifiers with the same feature input across 10 months of training and testing periods between Oct 2013 and July 2014.

Figure 7 , 8 and 9 show the boxplots of F1-score, precision and recall for six input feature variable sets for the Naïve Bayes classifier given its relative advantage shown in Table 4, i.e. "*Frequency*", "*Frequency+Population*", "*Frequency+ Type*", "*Frequency+CapitalCity*","*Frequency+Type+CapitalCity*" and *Frequency+Population+Type+CapitalCity*".

Table 4. The evaluation results for different classifiers

	Training				Testing			
	RF	**DT**	**SVM**	**NB**	**RF**	**DT**	**SVM**	**NB**
mean	0.38	0.39	0.41	0.44	0.28	0.29	0.29	**0.34**
std	0.09	0.08	0.07	0.05	0.09	0.09	0.08	0.08

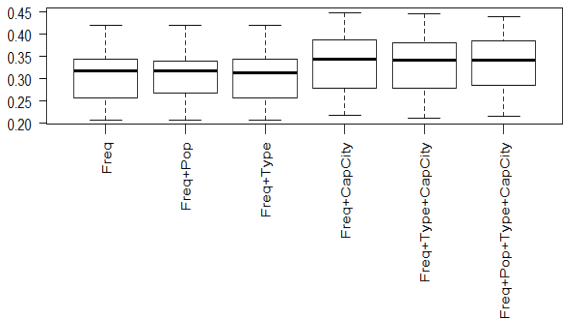


Figure 7. Boxplots of F1 score

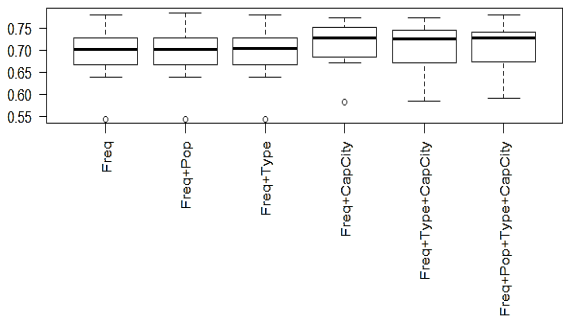


Figure 8. Boxplots of precision

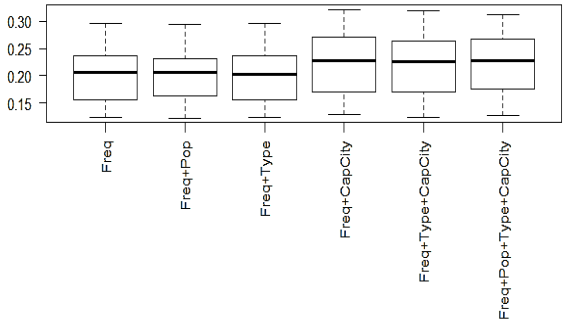


Figure 9. Boxplots of recall

It can be seen that adding "*CapitalCity*" can improve the prediction performance. The inclusion of event "*Type*" and "*Population*" in the input doesn't appear to improve the results.

Figure 10. shows a detailed comparison of the F1-score over the 10 months of testing period. The blue curve is for the input with only "*Frequency*", while the red one is with both "*Frequency*" and "*CapitalCity*". Thus the results confirm that the new feature can constantly improve the prediction performance of the model.

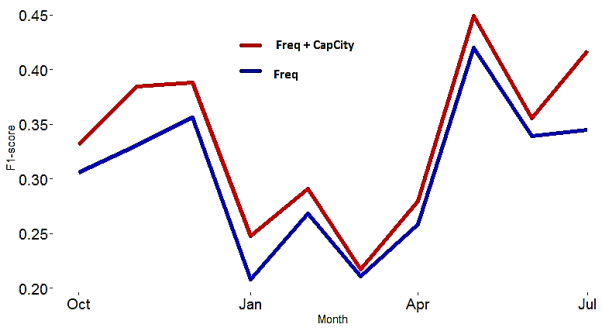


Figure 10. Comparison of F1-score with or without "*CapitalCity*"

Regarding the trained models, both Decision Trees and Naïve Bayes are interpretable, meaning than human can easily understand the information presented in the models. Here we present a decision tree visualised in Rattle in Figure 11, which is used to explain why the classification models can predict recurring events with the simple rules listed as follows.

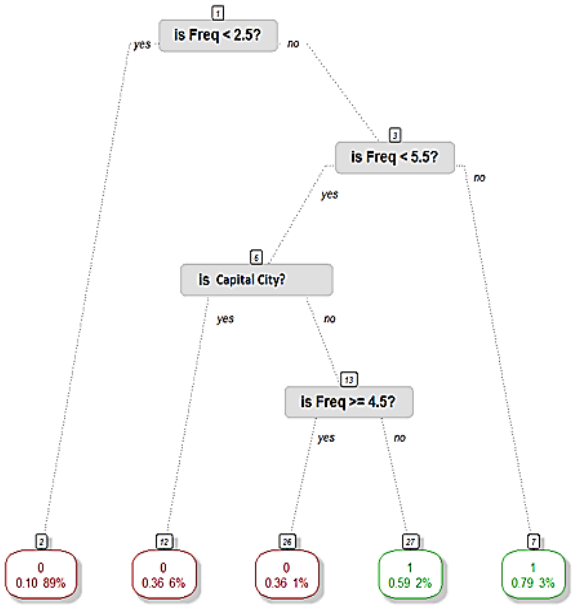


Figure 11. A decision tree explanation

- If *Freq* >= 5.5, the events will recur with a probability of 0.79, it covers 3% of the training cases
- If *Freq* < 2.5, the events won't recur with a probability 0.9, it covers 89% of the training cases
- If 2.5< *Freq* < 5.5 , and the city is one of the 10 capital cities, then events won't recur with a probability 0.64, covering 6% of the training cases
- If 2.5=< *Freq* < 4.5 , and the city is not one of the 10 capital cities, then events will recur with a probability 0.59, covering 2% of the training cases.

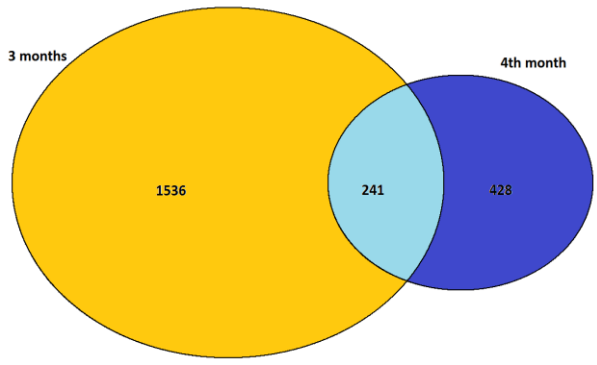


Figure 12. Illustration of distributions in 3 month feature period and 1 month outcome period

Lastly, it is worthy noting that there is a limitation of recall of this kind of method as shown in Figure 12. Take one example from the training and outcome periods. Feature window 3 months, only 241 intersection of the 3 months (1777 rows) and 1 month (669) Outcome period, thus there are 428 combinations (64%) that were not expected to be predicted. This explains that the recall of baseline models only based on historical GSR data is generally not quite high.

5.2 Evaluation based on EMBERS measures

The current evaluation of baseline models generated models are monthly averaged measures without the consideration of the predicted date. The dates of the warnings/predictions generated from the model are controlled by a uniform distribution random variable. In Table 5, it is the reported the performance metrics using the evaluation library based on the EMBERS system. The evaluation methodology developed in EMBERS system is comprehensive because there are five criteria - lead time, quality score, probability, recall and precision of the warnings against the ground-truth GSR data based on the non-crossing matching (Malucelli et al. 1993).

Table 5. The evaluation results based on Embers measures

Metric	Value
quality	**3.22**
leadTime	15
probability	0.96
precision	0.98
recall	0.37

It can be seen that the average quality score of warnings is 3.22 versus 3.0 published in (Ramkrishnan et al. 2014) for the same 10 Latin American countries.

6 Discussion and Conclusions

Predicting societal events especially CUE is becoming a critical requirement from the government agencies. The proposed temporal classification based model is effective in the prediction of the events repeating themselves, and provide improved performance as indicated in the evaluation with the state of the art metrics of EMBERS system. The model developed in this paper will serve as the baseline to optimise the utilisation of open data sources to eventually forecast CUE as the normal services in government agencies. The future work includes:

- Use the predicted information from the aggregated time series of 10 countries to benefit the prediction at much finer resolution of locations.
- Develop models that take the input of twitter and other open data sources to add more informative feature variables.
- Our research team will also use summary information from pairs instead of triplets in the paper to develop predictive models. This work is intended for using the D2D CRC's GSR with Australian records.
- The temporal classification based model can be extended to the prediction of spatio-temporal events in other domains.

7 Acknowledgement

Our work has been conducted with the support from the BTN team at D2D CRC. In addition, the GSR data

provided by MITRE were a key component in undertaking this research.

8 References

S. Agarwal S. and Sureka A. (2015): "Applying social media intelligence for predicting and identifying on-line radicalization and civil unrest oriented threats," arXiv preprint arXiv:1511.06858

Compton, R. et al. (2013): Detecting future social unrest in unprocessed twitter data:"emerging phenomena and big data," in: *Intelligence and Security Informatics (ISI), 2013 IEEE International Conference On*. IEEE, pp. 56–60

Chen, J. et al. (2005): Representing association classification rules mined from health data, *Knowledge Based Intelligent Systems for Healthcare in KES*, 1225-1231.

Chen, J. et al. (2010): Mining consequence events in temporal health data. *Intell. Data Anal.* 14(2): 245-261

Hua, T. et al. (2013): semi-supervised targeted-interest event detection in in twitter. *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 1466-1469.

Hyndman, R.J. et al. (2015): Package "forecast." http://cran.r-project.org/web/packages/forecast/forecast.pdf

King, G. (1988): "Statistical Models for Political Science Event Counts: Bias in Conventional Procedures and Evidence for The Exponential Poisson Regression Model." *American Journal of Political Science*, 32: 838-863

Korkmaz, G. et al. (2015): Combining Heterogeneous Data Sources for Civil Unrest Fore-casting. ASONAM 2015: 258-265

Korkmaz, G. (2014): "Challenges of Twitter-based Predictions of Civil Unrest in Latin America." http://mappingideas.sdsu.edu/old_Mappingideas/SummerWorkshop/2014/Position/Korkmaz.pdf

Leetaru, K. and Schrodt P. (2013): GDELT: Global data on events, location, and tone, 1979-2012. *ISA Annual Convention*, pages 1979-2012, 2013.

Liu, L. and Li, J. (2013): Building Naïve Bayes classifiers with high-dimensional and small-sized datasets , in Arvin Agah (ed.), *Medical Applications of Artificial Intelligence*, CRC Press, US, pp. 115-137.

Malucelli, F. et al. (1993): Efficient labelling algorithms for the maximum noncrossing matching problem. *Discrete Applied Mathematics*, 47(2)

Muthiah, S. et al. (2015): Planned Protest Modeling in News and Social Media. In *AAAI*, pp. 3920-3927

OSI Team (2015): *Open Source Indicators Handbook*. MITRE

Pang-Ning Tan, Michael Steinbach, Vipin Kumar (2005): "*Introduction to Data Mining*," Addison Wesley, Boston, MA, May.

Ramkrishnan, N. et al (2014): 'Beating the news' with EMBERS: forecasting civil unrest using open source indicators. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 1799-1808.

Roddick, J.F. and Spiliopoulou M.(2002): A Survey of Temporal Knowledge Discovery Paradigms and Methods. *IEEE Trans. on Knowl. and Data Eng.* 14, 4 (July 2002), 750-767.

Saraf, P. and Ramakrishnan, N. (2016): EMBERS AutoGSR: Automated Coding of Civil Unrest Events, *Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'16)*, San Francisco, CA, Aug 2016

Williams, G. J. (2011): *Data Mining with Rattle and R: The Art of Excavating Data for Knowledge Discovery*, Springer.

Xu, J., Lu, T.-C., Compton, R. & Allen, D. (2014): Civil unrest prediction: A tumblr-based exploration. Social Computing, Behavioral-Cultural Modeling and Prediction. *Proceedings of 7th International Conference, SBP 2014, Washington, DC*. Springer

Knowledge Discovery from a Data Set on Dementia through Decision Forest

Md Nasim Adnan **Md Zahidul Islam**

School of Computing and Mathematics
Charles Sturt University,
Panorama Avenue, Bathurst, NSW 2795,
Email: madnan@.csu.edu.au, zislam@csu.edu.au.

Abstract

Decision tree is one of the most popular classifiers used in a wide range of real world problems for both prediction and classification (logic) rules discovery. A decision forest is an ensemble of decision trees, and it is often built for predicting class values more accurately than a single decision tree. Besides improving predictive performance, a decision forest can be seen as a pool of logic rules with great potential for knowledge discovery. However, an standard-sized decision forest usually generates a large number of logic rules that a user may not able to manage for effective knowledge analysis. In this paper, we propose a novel, problem (data set) independent framework for extracting those rules that are comparatively more accurate as well as reliable than others. We apply the proposed framework on rule sets generated from two different decision forest algorithms from a publicly available data set on dementia and compare the subsets of rules with the rules generated from a single J48 decision tree in order to show the effectiveness of the proposed framework.

Keywords: Decision Tree, Decision Forest, Random Forest, Dementia, Knowledge Discovery.

1 Introduction

Nowadays, data generation is accelerating in an unprecedented rate; data generation will be 44 times greater in 2020 than it was in 2009 (*Big Data Universe Beginning to Explode* n.d.). Different data mining tasks such as classification is applied on these data in order to identify useful patterns. Classification aims to generate a function (commonly known as a classifier) that maps the set of non-class attributes $\boldsymbol{m} = \{A_1, A_2, ..., A_m\}$ to a predefined class attribute C from an existing data set $\boldsymbol{D}$ (Tan, Steinbach & Kumar 2006). A data set generally has two types of attributes such as numerical (e.g. Salary) and categorical (e.g. Degree). Out of all categorical attributes, one is chosen as the class attribute. All other attributes are termed as non-class attributes. A classifier is then built from an existing data set (i.e. training data set) where the values of the class attribute are present and then applied on unseen/test records in order to predict their class values.

There are different types of classifiers including Artificial Neural Networks (Zhang 2000), Bayesian Classifiers (Bishop 2008), Decision Trees (Breiman, Friedman, Olshen & Stone 1985), (Quinlan 1993), (Quinlan 1996*b*), and Support Vector Machines (Burges 1998). Some of these classifiers such as a Support Vector Machine works similar to "black box" where they only give classification results without providing any reasoning for the results (Liu, Patel, Daga, Liu, Fu, Doerksen, Chen & Wilkins 2012). On the other hand, decision tree is a flow-chart like structure that closely resembles human reasoning and thus very popular to the real-world users (Murthy 1998).

A decision tree consists of nodes (denoted by rectangles) and leaves (denoted by ovals) as shown in Figure 1. The node of a decision tree symbolizes a splitting event where the splitting attribute (label of the node) partitions the data set according to its domain values. As a result, a disjoint set of horizontal segments of the data sets are generated and each segment contains one set of domain values of the splitting attribute. A leaf of a decision tree represents a horizontal segment of a data set where no further splitting is carried out. In this way, the records of an entire training data set are distributed among the terminal data segments or leaves. The path from the root node to a leaf node makes up a logic rule (i.e. pattern) that identifies a relationship between the non-class attributes (splitting attributes along the path) and the class values. For example, the logic rule for Leaf 1 is "if *Degree* = *Masters* AND *Income* $\leq$ *85K* $\rightarrow$ *Lecturer*" as majority records (all four in this case) belonging to the segment represented by Leaf 1 have the class value *Lecturer*. Here, "if *Degree* = *Masters* AND *Income* $\leq$ *85K*" is the antecedent of the logic rule and "*Lecturer*" is the consequent. A decision tree is then used to predict the class values of unseen records of a testing data set for which the class values are unknown (i.e. records are unlabeled). Based on the values of non-class attribute/s of an unlabeled record it passes through an antecedent to be predicted as the consequent (Adnan & Islam 2014).

Hunt's algorithm is in general a greedy top-down recursive partitioning strategy attempting to secure "purest class value distribution" in the succeeding partitions. Different tree induction algorithms that follow the same structure differs only in using impurity measures (for measuring the purity of class distribution) to find the splitting attributes. For example, CART (Breiman et al. 1985) uses Gini Index while C4.5 (Quinlan 1993), (Quinlan 1996*b*) uses Gain Ratio as the impurity measures. Each of the greedy strategies for generating decision tree picks the locally best attribute that delivers the "purest class value distribution" in succeeding partitions as the splitting attribute. However, this locally best attribute may not be the ultimate best attribute selected for that

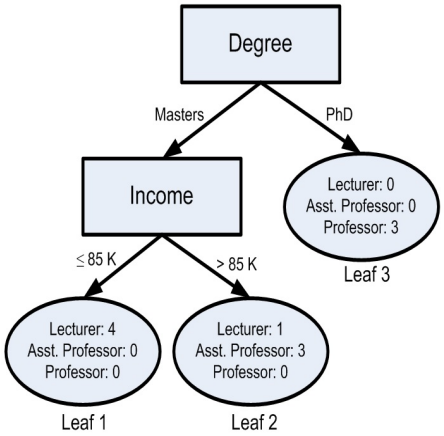


Figure 1: Decision Tree

particular partition. Moreover, a single decision tree can discover only one set of logic rules and thus may miss many potentially important logic rules useful for both classification and knowledge discovery.

The use of an ensemble of classifiers for classification is a comparatively newer area of research (Polikar 2006), (Rodriguez, Kuncheva & Alonso 2006). Interestingly, an ensemble of classifiers is found to be effective for unstable classifiers such as decision trees (Tan et al. 2006). Decision trees are considered to be an unstable classifier because a slight change in a training data set can cause a significant change between the resulting decision trees obtained from the original and modified data sets. A decision forest is an ensemble of decision trees where an individual decision tree acts as a base classifier. The classification is performed by taking a vote based on the predictions made by each decision tree of the decision forest (Tan et al. 2006).

In order to achieve better ensemble accuracy a decision forest requires both accurate and diverse individual decision trees as base classifiers (Polikar 2006), (Ho 1998). An accurate decision tree can be generated by feeding a training data set to a decision tree algorithm such as C4.5 (Quinlan 1993), (Quinlan 1996*b*). As discussed earlier, a single decision tree can discover only one set of logic rules and thus may wrongly predict the class value of a test record which could have been predicted correctly by a more appropriate logic rule. A different decision tree can be obtained from a differentiated data set which may include a more appropriate logic rule for the given test record. If a decision forest contains a set of decision trees which are different from each other then some of the trees may discover appropriate logic rules for a set of test records while some other trees may discover appropriate logic rules for another set of test records resulting in better predictive performance for the forest.

Several decision forest algorithms exist aiming to generate more accurate and diverse decision trees by manipulating the training data set. Bagging (Breiman 1996) generates a new training data set $\boldsymbol{D}_i$ where the records of $\boldsymbol{D}_i$ are chosen randomly from the original training data set $\boldsymbol{D}$. A new training data set $\boldsymbol{D}_i$ contains the same number of records as in $\boldsymbol{D}$. Thus, some records of $\boldsymbol{D}$ can be chosen multiple times and some records may not be chosen at all. Theoretically, 63.2% of the original records can be chosen in a bootstrap sample (Han & Kamber 2006). Bagging generates a predefined number (T) of bootstrap samples $\boldsymbol{D}_1, \boldsymbol{D}_2, ..., \boldsymbol{D}_T$ using the above approach. A decision tree building algorithm is then applied on each bootstrap sample $\boldsymbol{D}_i$ ($i = 1, 2, \ldots, T$) in order to generate altogether T number of trees for the forest. The Random Subspace algorithm (Ho 1998) randomly draws a subset of attributes (subspace) $\boldsymbol{f}$ from the entire attribute set $\boldsymbol{m}$ in order to determine the splitting attribute for each node of a decision tree. Random Forest (Breiman 2001) is regarded as a state-of-the-art decision forest building algorithm (Bernard, Heutte & Adam 2008), (Bernard, Adam & Heutte 2012) that is technically a fusion of Bagging (Breiman 1996) and Random Subspace (Ho 1998) algorithms.

Besides improving predictive performance, a decision forest can be seen as a pool of logic rules with great potential as a source of knowledge discovery. However, the main challenge of knowledge discovery comes from the enormous number of logic rules that an standard-sized decision forest (usually a 100-tree decision forest (Geurts, Ernst & Wehenkel 2006), (Bernard et al. 2012), (Amasyali & Ersoy 2014), (Adnan & Islam 2015*c*), (Adnan & Islam 2015*b*)) generates. For effective knowledge discovery, we need to extract accurate, reliable, concise as well as surprising logic rules (Geng & Hamilton 2006). An obvious technique to extract a subset of logic rules is to apply some cut points based on accuracy, coverage or length of the rules. For example, we can extract only those logic rules that have accuracy $\geq 80\%$. However, any such cut points may react differently from data set to data set. As a result, for one data set a cut point may net more than manageable rules whereas for another data set it can acquire too few rules. To avoid such situation, a user may need to expedient on different cut points for a single variable (such as accuracy) and subsequently on all possible combinations of the variables for each data set which may not be manageable either.

There are some existing rule extraction techniques in literature (Liu et al. 2012), (Mashayekhi & Gras 2015) that mainly prune forest rules in order to increase the prediction accuracy (just like pruning decision trees from a forest) and consequently do not focus on issues in knowledge discovery. In this paper, we propose a novel, problem (data set) independent framework for extracting those rules that are comparatively more accurate as well as reliable to help uncover valuable knowledge. We then apply the proposed framework on rule sets generated from two different decision forest algorithms from the same data set on dementia (publicly available, (*OASIS* n.d.)) and compare the subsets of rules with the rules generated from a single J48 tree (Hall, Frank, Holmes, Pfahringer, Reutemann & Witten 2009) (the Weka implementation of C4.5 (Quinlan 1993), (Quinlan 1996*b*)) in order to show the effectiveness of the proposed framework.

The remainder of this paper is organized as follows: In Section 2 we discuss about dementia, the data set on dementia and challenges of knowledge discovery from existing decision tree and decision forest algorithms. Section 3 explains the proposed knowledge extraction framework in detail. Section 4 discusses the experimental results. Finally, we offer some concluding remarks in Section 5.

2 Dementia, Data Set on Dementia and Challenges of Knowledge Discovery from Random Forest

Dementia is a medical term linked with memory loss that is severe enough to interfere with daily life (*What*

is Dementia? n.d.). Alzheimer's disease (AD) is the most common cause of dementia, accounting for almost 70% of all dementia cases (*Dementia* n.d.). AD is caused by abnormal deposits of protein in the brain that destroy cells in the areas of the brain (cerebral cortex) that is responsible for thoughts, memories, actions, and personality. Unfortunately, AD is degenerative (destroys brain cells that causes to shrink the size of the brain over time), progressive (gradual decline of functioning of effected areas of the brain) and thus irreversible (*Dementia* n.d.). AD is closely related with advancing of age (About 5% of people above 65 years of age, about 20% of those over 80 years and about 30% of those over 90 suffer from AD) and family history (statistically, people who have a parent or sibling affected by AD are two to three times more likely to develop the disease than those with no family history; if more than one close relative has been affected by the disease, the risk increases even more) (*Dementia* n.d.). Vascular dementia (VaD) is the second most common cause of dementia, accounting for about 20% of cases (*Dementia* n.d.). VaD is mainly caused by full or partial blocking of arteries in the brain from deposits of fats, dead cells, and other debris that ultimately disrupts the blood flow. Vascular dementia is often related to high blood pressure, high cholesterol, heart disease, diabetes, and other related conditions. Treating those conditions can slowdown the progress of vascular dementia, but as usual the brain functions that are lost are not recoverable. In VaD, the size the brain may not shrink at all (*Dementia* n.d.).

Overall, dementia is estimated to afflict over 35.5 million people worldwide – this includes nearly 10 million people in Europe, 4.4 million in North America, 7 million in South and South-east Asia, 5.5 million in China and East Asia and 3 million in Latin America (*Dementia: A general introduction* n.d.). Moreover, World Health Organization (WHO) projects that the total population of sufferers will double by 2030 and triple by 2050 (Ertek, Tokdil & Gunaydin 2014). Emphasised by these statistics and forecasts, we need to develop a deeper insight on dementia based on available data to help avoiding/curing the disease.

In this paper, we use a data set named "OASIS: Longitudinal MRI Data in Nondemented and Demented Older Adults" that consists of a collection of 354 observations (records) on 142 subjects (patients) aged between 60 to 98 and all the observations are distributed among three (3) class values (Nondemented, Converted and Demented) (*OASIS* n.d.). Converted class value refers to the patients that develop dementia during the period of the data collection. Other than MRI data, the data set also includes information about patient's Gender, Age, Education Level and Socio-economic status. Some scores on dementia-related examinations are also included. In Table 1, we present the attributes of the OASIS data set (*OASIS* n.d.) and explain their meanings with their respective value ranges in detail.

As presented in Table 1, both MMSE and CDR indicate the cognitive situation of a patient that strongly associate with the final diagnosis of a patient (whether he/she is demented or not). In addition, nWBV, eTIV and ASF values are obtained from the MRI images and are directly linked with the size of the brain appeared in the MRI images. In (Ertek et al. 2014), the authors viewed only AD causing dementia (ignoring that VaD too can cause dementia as discussed earlier) and as a result all their discussions concentrated on generating a risk map of having AD based on factors such as Gender, Age, EDUC, SES and other results of medical tests. Uncertainty is al-

Table 1: Description of the OASIS data set

Attributes	**Explanation**
MRI ID	The unique number of tests (354 in total).
Subject ID	The number of unique patients (142 in total). One patient may be visiting multiple times for MRI tests, so the number of MRI tests (354) is larger than the number of subjects.
Visit	Chronological visit number of a patient.
MR Delay	The delay since the last visit.
Gender	Male (M) or Female (F).
Hand	Right (R) or Left (L)
Age	Ages of the patients vary between 60 to 98.
EDUC	EDUCation level of the patients vary between 6 to 23 representing years of education.
SES	Socio-Economic Status of the patients assigned through the Hollingshead Index of Social Position. 1 representing the highest status to 5 representing the lowest status (Hollingshead 1975).
MMSE	Mini-Mental State Examination value ranges between 0 to 30. In MMSE, a health professional asks a patient a series of questions designed to test a range of everyday mental skills. The questions mainly cover preliminary arithmetic problems, simple memory tests, and recognition of different orientations of objects. A score of 20 to 24 suggests mild dementia, 13 to 20 suggests moderate dementia, and less than 12 indicates severe dementia (*What is Dementia?* n.d.), (Folstein et al. 1975).
CDR	Clinical Dementia Rating. 0 indicates No dementia, 0.5 indicates very mild dementia, 1 indicates mild dementia and 2 indicates moderate dementia (Morris 1994).
eTIV	estimated Total Intra-cranial Volume (in cm^3) of the brain (proportional to the size of the skull, can be obtained from MRI image) (Buckner et al. 2004).
nWBV	normalized Whole-Brain Volume, expressed as a percent of all voxels (can be obtained from MRI image) (Buckner et al. 2005).
ASF	Atlas Scale Factor is the volume scaling factor for brain size (proportional to nWBS and eTIV (Buckner et al. 2004)).
Class values	Three class values. Nondemented, Converted and Demented.

ways the central and critical fact about medical diagnosis that necessitates doctors to prescribe medical tests of different spheres (Szolovits 1995). For example, all the patients contained in the OASIS data set (*OASIS* n.d.) may not suffer from AD; some of them may also suffer from VaD. Moreover, in (Ertek et al. 2014) the insights on AD was generated using a single J48 tree.

We already know, decision trees are more favoured to the real-world users as they are more easy-to-interpret over decision forests even though decision trees are not competitive against decision forests in terms of prediction accuracy. It is worth mentioning that a single decision tree cannot reveal logic rules of different angles as all logic rules generated from the tree start with the same attribute placed in the root node (see Figure 1). Many sensitive application areas such as medical diagnosis require comprehensive data analysis covering some different important angles to be explored. Interestingly, a decision forest can accommodate different angles of logic rules; however even state-of-the-art Random Forest (Breiman 2001) lacks any systematic mechanism to explore different angles of knowledge rather completely depends on randomness to generate different decision trees (as discussed earlier). In addition, the hyperparameter $(int(\log_2 |\boldsymbol{m}|)+1)$ (Bernard et al. 2008) used in Random Forest that determines the number attributes (to be drawn randomly) for a subspace ($\boldsymbol{f}$) is not evenly suited for low and high dimensional data sets (Adnan & Islam 2016*a*). For example, let us assume that we have a low dimensional data set consisting of 4 attributes. Thus a splitting attribute is determined from a randomly selected subspace of 3 attributes $(int(\log_2 4)+1 = 3)$ encompassing 75% of the total attributes. As a result, the chance of appearing similar attributes in different subspaces becomes high, resulting in decreasing diversity (may cause fewer angles to explore) among the trees. On the other hand, when $|\boldsymbol{m}|$ is large say, 150 then $|\boldsymbol{f}|$ contains 8 randomly chosen attributes $(int(\log_2 150)+1 = 8)$ covering only 5% of the total attributes. Hence, if the number of good attributes is not high enough in $\boldsymbol{m}$ then the chance of containing adequate number of good attributes in a subspace $\boldsymbol{f}$ becomes low, which is supposed to cause low individual accuracy (may miss many worthwhile angles to explore) of the trees (Adnan & Islam 2015*a*).

Recently, we have proposed a new decision forest algorithm called "*Forest by Penalizing Attributes (Forest PA)*" (Adnan & Islam 2016*b*) that imposes penalties on attributes systematically in such as way that an attribute tested at lower level (such as in the root node) receives higher penalty (lower weight) than an attribute tested at higher level. The reason is that an attribute tested at lower level may influence more logic rules than an attribute tested at higher level as discussed earlier. Thus in order to discover diverse set of logic rules, attributes tested at lower levels are supposed to be avoided in a future tree more seriously than those attributes that are tested at higher levels. Furthermore, to increase the chance of having different weights among attributes in the same level, *Forest PA* randomly selects the weight of an attribute from the Weight-Range (WR) allocated for the attribute's level (Adnan & Islam 2016*b*).

Forest PA also has a mechanism to gradually increase weights (withdraw penalties) from the attributes that have not been tested in the subsequent trees. This addresses the situation where all good attributes (attributes with high classification capacity) suffer from high penalties and thus poor quality trees can be generated at some later stage of the forest building process. Moreover, *Forest PA* uses bootstrap samples of the original training data set (Breiman 1996) to ensure further diversity. Even if two trees somehow end up having very similar weight distributions their corresponding training data sets are likely to be different due to the use of bootstrap samples (Adnan & Islam 2016*b*).

Apparently, *Forest PA* seems to be more suitable for knowledge discovery than state-of-the-art Random Forest. However, the number of logic rules generated from a standard-sized *Forest PA* may still hinder users from smooth knowledge discovery. Users may need to look for an accurate and reliable subset of rules by tuning different combinations of cut points based on accuracy, coverage or length of the rules as discussed earlier. Therefore, we propose a data set independent framework for knowledge discovery that can select a subset of logic rules that are comparatively more accurate as well as reliable than others.

3 The Proposed Rule Extracting Framework

We already know, the path from the root node to a leaf node makes up the antecedent and the majority class of the leaf node is the consequent of a logic rule in a decision tree. For example, the logic rule for Leaf 2 in Figure 1 is "if *Degree = Masters* AND *Income > 85K* → *Class Value = Asst. Professor*" as majority records (Lecturer: 1, Asst. Professor: 3) belonging to Leaf 2 have the class value *Asst. Professor*. The minority record(s) are viewed as if they are misplaced under a logic rule. Thus the Accuracy (*Acc*) of the logic rule for Leaf 2 is: $\frac{3}{4} = 0.75$. All the records of a data set from which a decision tree was built are distributed among the leaves. The Coverage (*Cov*) of the logic rule for a leaf indicates the proportion of records that fall in the leaf to the total records. For example, *Cov* of the logic rule for Leaf 2 is: $\frac{3}{11} = 0.27$. In general, *Acc* and *Cov* are independent of each other and thus no data set independent relationships exist between them. However, they are independently comparable among themselves. For example, we can find the set of rules $\mathbf{R}^{Acc}_{Avg}$ that are more (or equally) accurate than the average accuracy Acc_{Avg} of all rules ($\mathbf{R} = \{R_1, R_2, \ldots R_z\}$) in a forest as follows, where R^{Acc}_i is the accuracy of R_i.

$$\mathbf{R}^{Acc}_{Avg} = \{R_i : R^{Acc}_i \geq Acc_{Avg}\} | Acc_{Avg} = \frac{1}{|\mathbf{R}|}\sum_{j=1}^{|\mathbf{R}|} R^{Acc}_j \quad (1)$$

Similarly, we find the set of rules $\mathbf{R}^{Cov}_{Avg}$ that have more (or equal) coverage than the average coverage Cov_{Avg} of all rules in a forest as follows, where R^{C}_i is the coverage of R_i.

$$\mathbf{R}^{Cov}_{Avg} = \{R_i : R^{Cov}_i \geq Cov_{Avg}\} | Cov_{Avg} = \frac{1}{|\mathbf{R}|}\sum_{j=1}^{|\mathbf{R}|} R^{Cov}_j \quad (2)$$

Now, by applying both Equation 3 and Equation 2, we are able to recognize those high quality rules that have simultaneously more (or equal) accuracy and coverage than the averages of in a forest (see

Equation 3).

$$\mathbf{R}_{Avg}^{Acc.Cov} = \mathbf{R}_{Avg}^{Acc} \cap \mathbf{R}_{Avg}^{Cov} \quad (3)$$

After applying 3 on a pool of forest rules $\mathbf{R}$, we get a subset $\mathbf{R}_{Avg}^{Acc.Cov}$. As $\mathbf{R}_{Avg}^{Acc.Cov}$ is obtained from multiple trees, some of the rules may be exactly the same to others. In our framework, those rules appear once however they are emphasized by their number of appearances.

4 Experiments

In (Ertek et al. 2014), the authors applied J48 (Hall et al. 2009) on a modified version of OASIS data set in order to generate a decision tree. From the original OASIS data set, at first the authors excluded the identifier attributes ("MRI ID" and "Subject ID") and then built a preliminary decision tree. However, the first split of the tree based on "CDR" perfectly distinguished the demented patients (when "CDR" = 1) from others (non-demented and converted). This showed that "CDR" was too good attribute to be included in the analysis. Furthermore, attributes "MR Delay" and "Visit" were found strongly dependent on "CDR" in the preliminary tree as the following splits ware based on them (Ertek et al. 2014). Observing the "near perfect" results in the preliminary decision tree due to "CDR" and the inherent dependency problem of "MR Delay" and "Visit", the authors (Ertek et al. 2014) excluded them to generate the final decision tree. We further exclude the attribute "Hand" as each patient is right-handed in the data set to obtain $OASIS'$ used for all experiments in this paper.

Table 2 presents the logic rules generated from the final J48 decision tree as reported in (Ertek et al. 2014) with their respective accuracies (except we prune a few lengthy logic rules to contain at most five attributes in their antecedents for better understanding). We also report the coverages of the logic rules of the final J48 decision tree even though they were not reported in (Ertek et al. 2014). Nondemented, Converted and Demented are abbreviated as ND, C and D respectively to be presented in subsequent tables. Also Antecedent, Consequent, Accuracy, Coverage and number of Occurrences of logic rules are abbreviated as **Ante**, **Cons**, **Acc**, **Cov** and **Occ** respectively.

We then apply Random Forest and *Forest PA* to generate logic rules from the $OASIS'$. We already know, both Random Forest and *Forest PA* use bootstrap samples of a training data set. Bootstrap samples are generally used for inducing diversity among the base classifiers (Munoz & Suarez 2010), (Quinlan 1996*a*). However, bootstrap samples are somehow deviated from the original data set and thus can not be regarded as a valid source of knowledge. Therefore, for an even comparison among J48, Random Forest and *Forest PA*, we do not use bootstrap samples for both Random Forest and *Forest PA*. As a result, Random Forest is effectively converted to Random Subspace (RS) and *Forest PA* is converted to *Forest PA* WithOut Bootstrap Samples (*Forest PA* WOBS). We then apply the proposed framework on logic rules generated by RS and *Forest PA* WOBS from the $OASIS'$ data set to obtain their respective $\mathbf{R}_{Avg}^{Acc.Cov}$.

A 20-tree RS generates as many as 504 logic rules from $OASIS'$, even a cut point with accuracy $\geq 95\%$ will extract 318 rules from them, whereas $\mathbf{R}_{Avg}^{Acc.Cov}$ consists of as low as 46 rules. Similarly, a 20-tree *Forest PA* WOBS generates as many as 698 logic rules from $OASIS'$, a cut point with accuracy $\geq 95\%$ will extract 378 rules from them, whereas $\mathbf{R}_{Avg}^{Acc.Cov}$ consists of as low as 62 rules. Though smaller in number, it is difficult to accommodate each $\mathbf{R}_{Avg}^{Acc.Cov}$ rules for detailed comparison in this paper. However, we present a brief comparison between $\mathbf{R}_{Avg}^{Acc.Cov}$ from RS and *Forest PA* WOBS in Table 3.

Table 2: Logic Rules from J48 Decision Tree

Ante	**Cons**	**Acc**	**Cov**
If MMSE $\leq$ 26	D	94%	24%
If MMSE $>$ 28 and Gender = F	ND	85%	36%
If MMSE $>$ 28 and Gender = M and EDUC $\leq$ 13	ND	79%	5%
If MMSE $>$ 28 and Gender = M and EDUC between 13 and 15	D	86%	2%
If MMSE $>$ 28 and Gender = M and EDUC $\leq$ 13	ND	80%	5%
If MMSE $>$ 28 and Gender = M and EDUC $>$ 15 and ASF $\leq$ 0.93	D	80%	1%
If MMSE $>$ 28 and Gender = M and EDUC $>$ 15 and ASF $>$ 0.93	ND	69%	7%
If MMSE between 27 and 28 and Gender = M and nWBV $\leq$ 0.68	ND	80%	1%
If MMSE between 27 and 28 and Gender = M and nWBV $>$ 0.68	D	62%	8%
If MMSE between 27 and 28 and Gender = F and nWBV $\leq$ 0.71	D	63%	2%
If MMSE between 27 and 28 and Gender = F and nWBV $>$ 0.71	ND	66%	9%

Table 3: Comparison between $\mathbf{R}_{Avg}^{Acc.Cov}$ from RS and *Forest PA* WOBS

Criteria	$\mathbf{R}_{Avg}^{Acc.Cov}$ from RS	$\mathbf{R}_{Avg}^{Acc.Cov}$ from *Forest PA* WOBS
Total Rules	46	62
Average Accuracy	97%	98%
Average Coverage	10%	8%
Average Rule Length	8	8.5
Total Unique Rules	39	45
Unique Root Nodes	3	4

From Table 3, we see that $\mathbf{R}_{Avg}^{Acc.Cov}$ from *Forest PA* WOBS generates more number of unique logic rules with more unique root nodes than that from RS. Moreover, the average accuracy of $\mathbf{R}_{Avg}^{Acc.Cov}$ from *Forest PA* WOBS are higher; however with less average coverage. We now select a part of logic rules from both $\mathbf{R}_{Avg}^{Acc.Cov}$ of RS and *Forest PA* WOBS for a more detailed comparison. For this purpose, we select 11 logic rules each having at most five attributes in their antecedents from both $\mathbf{R}_{Avg}^{Acc.Cov}$ of RS and *Forest PA* WOBS (in accordance with the J48 logic rules presented in Table 2) and present them with their respective accuracies, coverages and number of occurrences in $\mathbf{R}_{Avg}^{Acc.Cov}$ in Table 4 and Table 5, respectively.

We now discuss the pros and cons of the logic

Table 4: Part of $\mathbf{R}_{Avg}^{Acc.Cov}$ from RS

Ante	Cons	Acc	Cov	Occ
If MMSE $\leq$ 26	D	94%	24%	3
If MMSE $\leq$ 26 and nWBV $\leq$ 0.74	D	97%	20%	3
If MMSE $\leq$ 26 and nWBV $\leq$ 0.74 and eTIV $\leq$ 1443.71 and EDUC $\leq$ 16	D	100%	6%	1
If MMSE $>$ 27 and Gender = F and EDUC $>$ 16	ND	100%	11%	1
If nWBV $\leq$ 0.73 and MMSE $\leq$ 26	D	96%	18%	1
If nWBV $\leq$ 0.73 and SES = 2 and Gender = F and MMSE $>$ 28	ND	100%	4%	1
If nWBV $\leq$ 0.73 and MMSE $>$ 26 and EDUC $>$ 16 and eTIV $\leq$ 1825.54 and Age $\leq$ 78	ND	100%	4%	1
If SES = 1 and MMSE $>$ 27 and eTIV between 1423.48 and 1559.41	ND	100%	4%	1
If SES = 2 and MMSE $>$ 28 and nWBV $\leq$ 71	ND	96%	14%	2
If SES = 3 and eTIV $\leq$ 1491.22 and nWBV $\leq$ 0.71 and Age $\leq$ 86	D	100%	5%	1
If SES = 4 and nWBV $\leq$ 0.78 and Age $\leq$ 72 and MMSE $\leq$ 28	D	100%	5%	1

Table 5: Part of $\mathbf{R}_{Avg}^{Acc.Cov}$ from *Forest PA* WOBS

Ante	Cons	Acc	Cov	Occ
If MMSE $\leq$ 26	D	94%	24%	6
If MMSE $\leq$ 27 and nWBV $\leq$ 0.75	D	97%	20%	1
If nWBV $\leq$ 0.72 and SES = 3 and EDUC $\leq$ 16 and Age $\leq$ 84	D	96%	6%	1
If nWBV $\leq$ 0.73 and EDUC between 14 and 16 and Age $\leq$ 72	D	100%	5%	1
If nWBV $\leq$ 0.73 and EDUC $>$ 16 and ASF $>$ 0.92 and Age between 67 and 78	ND	100%	4%	1
If nWBV $\leq$ 0.73 and SES = 4 and MMSE $\leq$ 27.0 and Age $\leq$ 88.0	D	100%	7%	1
If nWBV $>$ 73 and Age $>$ 79 and eTIV $>$ 1483.25	ND	100%	5%	1
If nWBV $>$ 0.73 and SES = 2 and EDUC $>$ 12	ND	100%	7%	2
If SES = 4 and EDUC $>$ 8 and Age $>$ 72 and eTIV $>$ 1493.89	ND	92%	4%	1
If Gender = M and eTIV $\leq$ 1604.47 and Age $>$ 78 and EDUC $\leq$ 15	D	100%	5%	1
If Gender = F and SES = 1 and eTIV $>$ 1423.47 and Age $\leq$ 94.0	ND	100%	4%	1

rules reported in Table 2 (logic rules from J48 decision tree), Table 4 (Part of $\mathbf{R}_{Avg}^{Acc.Cov}$ from RS) and Table 5 (Part of $\mathbf{R}_{Avg}^{Acc.Cov}$ from *Forest PA* WOBS). We see that some of the logic rules obtained J48 decision tree have very low coverage. For example, out of 11 logic rules 4 of them are having $\leq 2\%$ coverages to secure vary little or no statistical support to be reliable (Fayyad & Irani 1992). Moreover, the average accuracy of J48 rules is 77%. On the contrary, the average accuracy of $\mathbf{R}_{Avg}^{Acc.Cov}$ from RS is 97% and the average accuracy of $\mathbf{R}_{Avg}^{Acc.Cov}$ from *Forest PA* WOBS is 98% (The average of the reported rules for both RS and *Forest PA* WOBS is 98%). Besides, no rule in $\mathbf{R}_{Avg}^{Acc.Cov}$ for both RS and *Forest PA* WOBS has coverage as low as 1% or 2%. Now we understand that $\mathbf{R}_{Avg}^{Acc.Cov}$ can cover comparatively more accurate as well as reliable logic rules but what about covering different angles of logic rules? In search of this question, we assess how the reported part of $\mathbf{R}_{Avg}^{Acc.Cov}$ rules accommodate different angles compared to J48 rules.

For RS, one rule from the reported part of $\mathbf{R}_{Avg}^{Acc.Cov}$ improves over a J48 rule "If MMSE $>$ 28 and Gender = F then Nondemented" with 85% accuracy by containing "If MMSE $>$ 27 and Gender = F and EDUC $>$ 16 then Nondemented" with 100% accuracy. This rule exposes an important insight that EDUC plays an important role in dementia: i.e. the higher the EDUC, the lower the chance to be demented. Besides, from the last four rules of $\mathbf{R}_{Avg}^{Acc.Cov}$ for RS (see Table 4) we understand that SES also plays an important role in dementia: i.e. the higher the socio-economic status, the lower the chance to be demented. However, 10 out of 11 reported rules of $\mathbf{R}_{Avg}^{Acc.Cov}$ from RS (42 out of all 46 rules) contain the attribute "MMSE" which is found as good as "CDR" in distinguishing the class values of patients (also similar in a sense that both of them measure the cognitive situation of a patient). The reason behind this is RS has limited scope in excluding MMSE for low dimensionality of $OASIS'$.

For *Forest PA* WOBS, the reported part of $\mathbf{R}_{Avg}^{Acc.Cov}$ shows very interesting finding on dementia. For example, we find the rule "If MMSE $\leq$ 27 and nWBV $\leq$ 0.75 then Demented" with 97% accuracy and 20% coverage and thus improving the most common rule "If MMSE $\leq$ 26 then Demented" with 94% accuracy. Moreover, even the reported part of $\mathbf{R}_{Avg}^{Acc.Cov}$ for *Forest PA* WOBS accommodate different angles of knowledge exploration suppressing MMSE (unlike J48 and $\mathbf{R}_{Avg}^{Acc.Cov}$ of RS rules). For example, these rules validate the importance of EDUC by the following two rules with 100% accuracy: "If nWBV $\leq$ 0.73 and EDUC between 14 and 16 and Age $\leq$ 72 the Demented", If nWBV $\leq$ 0.73 and EDUC $>$ 16 and ASF $>$ 0.92 and Age between 67 and 78 then Nondemented. Here we find that, even if the brain size shrinks (nWBV $\leq$ 0.73), higher education (EDUC $>$ 16) can play its role to prevent dementia. Also, one rule ("If SES = 4 and EDUC $>$ 8 and Age $>$ 72 and eTIV $>$ 1493.89 then Nondemented" with 92% accuracy) finds out the importance of eTIV, i.e. the higher the eTIV (size of the head, in general), the lower the chance of having dementia even for people with lower socio-economic status. Furthermore, only 28 out of 62 logic rules contain the attribute "MMSE" in $\mathbf{R}_{Avg}^{Acc.Cov}$ from *Forest PA* WOBS and thus the logic

rules are in general more exploratory.

5 Conclusion

In this paper we propose a novel, problem (data set) independent framework for extracting a subset of decision forest rules that are comparatively more accurate as well as reliable than others. We apply both Random Subspace and *Forest PA* WithOut Bootstrap Samples on OASIS data set to generate forest rules and then employ the proposed the framework in order to obtain the subset of rules. Compared to J48 rules, the subset of rules are far more accurate, does not contain any rule with very low coverage and can accommodate different angles for knowledge exploration. Thus, the proposed framework enables us to use decision forests as a source of knowledge discovery more effectively. However, a major drawback of the proposed framework is that it often extracts partial rules. For example, a rule "If Gender = M and eTIV $\leq$ 1604.47 and Age $>$ 78 and EDUC $\leq$ 15 then Demented" can be extracted whereas the "else" rule may get abandoned (as opposed to a decision tree). Also, rules from majority class may dominate the subset and thus may suppress surprising rules. In future, we shall address the problems of the proposed framework.

References

Adnan, M. N. & Islam, M. Z. (2014), Combosplit: Combining various splitting criteria for building a single decision tree, *in* 'Proceedings of the International Conference on Artificial Intelligence and Pattern Recognition', pp. 1–8.

Adnan, M. N. & Islam, M. Z. (2015*a*), Complement random forest, *in* 'Proceedings of the 13th Australasian Data Mining Conference (AusDM)', pp. 89–97.

Adnan, M. N. & Islam, M. Z. (2015*b*), Improving the random forest algorithm by randomly varying the size of the bootstrap samples for low dimensional data sets, *in* 'Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning', pp. 391–396.

Adnan, M. N. & Islam, M. Z. (2015*c*), One-vs-all binarization technique in the context of random forest, *in* 'Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning', pp. 385–390.

Adnan, M. N. & Islam, M. Z. (2016*a*), Forest CERN: A new decision forest building technique, *in* 'In proceedings of the The 20th Pacific Asia Conference on Knowledge Discovery and Data Mining (PAKDD)', pp. 304–315.

Adnan, M. N. & Islam, M. Z. (2016*b*), 'Forest PA: Constructing a decision forest by penalizing attributes used in previous trees', *Data Mining and Knowledge Discovery* . In Review.

Amasyali, M. F. & Ersoy, O. K. (2014), 'Classifier ensembles with the extended space forest', *IEEE Transactions on Knowledge and Data Engineering* **16**, 145–153.

Bernard, S., Adam, S. & Heutte, L. (2012), 'Dynamic random forests', *Pattern Recognition Letters* **33**, 1580–1586.

Bernard, S., Heutte, L. & Adam, S. (2008), 'Forest-RK: A new random forest induction method', *Advanced Intelligent Computing Theories and Applications, Lecture Notes in Computer Science* **5227**, 430–437.

Big Data Universe Beginning to Explode (n.d.), http://www.csc.com/insights/flxwd/78931-big_data_universe_beginning_to_explode. Last Accessed: 15/03/2016.

Bishop, C. M. (2008), *Pattern Recognition and Machine Learning*, Springer-Verlag, NY, U.S.A.

Breiman, L. (1996), 'Bagging predictors', *Machine Learning* **24**, 123–140.

Breiman, L. (2001), 'Random forests', *Machine Learning* **45**, 5–32.

Breiman, L., Friedman, J., Olshen, R. & Stone, C. (1985), *Classification and Regression Trees*, Wadsworth International Group, CA, U.S.A.

Buckner, R. L., Head, D., Parker, J., Fotenos, A. F., Marcus, D., Morris, J. C. & Snyder, A. Z. (2004), 'A unified approach for morphometric and functional data analysis in young, old, and demented adults using automated atlas-based head size normalization: reliability and validation against manual measurement of total intracranial volume', *NeuroImage* **23**, 724–738.

Buckner, R. L., Snyder, A. Z., Shannon, B. J., LaRossa, G., Sachs, R., Fotenos, A. F., Sheline, Y. I., Klunk, W. E., Mathis, C. A., Morris, J. C. & Mintun, M. A. (2005), 'Molecular, structural, and functional characterization of alzheimer's disease: Evidence for a relationship between default activity, amyloid, and memory', *The Journal of Neurocsience* **25**(34), 7709–7717.

Burges, C. J. C. (1998), 'A tutorial on support vector machines for pattern recognition', *Data Mining and Knowledge Discovery* **2**, 121–167.

Dementia (n.d.), http://www.dementia.com/causes.html. Last Accessed: 15/03/2016.

Dementia: A general introduction (n.d.), http://www.memorykey.com/problems/dementia. Last Accessed: 15/03/2016.

Ertek, G., Tokdil, B. & Gunaydin, I. (2014), Risk factors and identifiers for alzheimers disease: A data mining analysis, *in* 'In proceedings of the 13th Industrial Conference on Data Mining (ICDM)', pp. 1–11.

Fayyad, U. M. & Irani, K. B. (1992), The attribute selection problem in decision tree generation, *in* 'Proceedings of the AAAI-92', pp. 104–110.

Folstein, M. F., Folstein, S. E. & McHugh, P. R. (1975), "'mini-mental state" a practical method for grading the cognitive state of patients for the clinician', *Journal of psychiatric research* **12**(3), 189–198.

Geng, L. & Hamilton, H. J. (2006), 'Interestingness measures for data mining: A survey', *ACM Computing Surveys* **38**, 1–32.

Geurts, P., Ernst, D. & Wehenkel, L. (2006), 'Extremely randomized trees', *Machine Learning* **63**, 3–42.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. & Witten, I. H. (2009), The weka data mining software: An update, *in* 'SIGKDD Explorations', Vol. 11.

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

Ho, T. K. (1998), 'The random subspace method for constructing decision forests', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**, 832–844.

Hollingshead, A. A. (1975), 'Four-factor index of social status', Unpublished manuscript, Yale University, New Haven, USA.

Liu, S., Patel, R. Y., Daga, P. R., Liu, H., Fu, G., Doerksen, R. J., Chen, Y. & Wilkins, D. E. (2012), 'Combined rule extraction and feature elimination in supervised classification', *IEEE Transactions on NanoBioscience* **11**(3), 228–236.

Mashayekhi, M. & Gras, R. (2015), Rule extraction from random forest: The rf+hc methods, *in* 'Proceedings of the 28th Canadian Conference on Artificial Intelligence, Canadian AI 2015', pp. 223–237.

Morris, J. C. (1994), 'The clinical dementia rating (cdr): current version and scoring rules', *Neurology* **43**(11), 2412–2414.

Munoz, G. M. & Suarez, A. (2010), 'Out-of-bag estimation of the optimal sample size in bagging', *Pattern Recognition* **43**, 143–152.

Murthy, S. K. (1998), 'Automatic construction of decision trees from data: A multi-disciplinary survey', *Data Mining and Knowledge Discovery* **2**, 345–389.

OASIS (n.d.), http://www.oasis-brains.org/. Last Accessed: 15/03/2016.

Polikar, R. (2006), 'Ensemble based systems in decision making', *IEEE Circuits and Systems Magazine* **6**, 21–45.

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

Quinlan, J. R. (1996*a*), Bagging, boosting, and c4.5, *in* 'Proceedings of the Thirteenth National Conference on Artificial Intelligence', pp. 725–730.

Quinlan, J. R. (1996*b*), 'Improved use of continuous attributes in c4.5', *Journal of Artificial Intelligence Research* **4**, 77–90.

Rodriguez, J. J., Kuncheva, L. I. & Alonso, C. J. (2006), 'Rotation forest: A new classifier ensemble method', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **28**, 1619–1630.

Szolovits, P. (1995), 'Uncertainty and decisions in medical informatics', *Methods of Information in Medicine* **34**(11), 1–121.

Tan, P. N., Steinbach, M. & Kumar, V. (2006), *Introduction to Data Mining*, Pearson Education.

What is Dementia? (n.d.), http://www.alz.org/what-is-dementia.asp. Last Accessed: 15/03/2016.

Zhang, G. P. (2000), 'Neural networks for classification: A survey', *IEEE Transactions on Systems, Man, and Cybernetics* **30**, 451–462.

EEG Biometric-Based Cryptographic Key Generation

Dang Nguyen[1] **Binh Nguyen**[1] **Dat Tran**[1] **Dharmendra Sharma**[1]
Wanli Ma[1]

[1] Faculty of Education, Science, Technology and Mathematics
University of Canberra, ACT 2601, Australia.
Email: Dang.van.Nguyen@canberra.edu.au

Abstract

Biometric-based cryptographic key generation is regarded as a data mining application that uses knowledge discovery techniques to extract biometric information to generate cryptographic keys. This application has been widely used in security systems to limit the weakness of passwords. Although conventional biometrics such as fingerprint, face, voice, and handwriting contain biometric information that is unique and repeatable for each individual, they are difficult to change to be used in different purposes. In this paper, we propose a system to exploit human electroencephalography (EEG) data as a new biometric for cryptographic key generation. This system provides high potential because EEG is impossible to be faked or compromised. Our experiments show that the proposed system provides good performance comparing with other biometric-based cryptographic key generation systems.

Keywords: Data mining, cryptographic key, EEG, biometrics.

1 Introduction

Data mining has become a new powerful technology for extraction of hidden predictive information from large databases. Biometric-based cryptographic key generation is regarded as a data mining application that uses knowledge discovery techniques to extract biometric information to generate cryptographic keys. These cryptographic keys are required to be sufficient long otherwise the security of cryptosystems will be broken. For instance, RSA keys are a length of 1024 bits as recommended in (Barker et al. 2012). People are hardly to remember these very long keys, but they can overcome this difficulty by using short and memorizing passwords to store them. However, using passwords has the following disadvantages: passwords can be guessed, or discovered by attacks (Feldmeier & Karn 1989, Klein 1990, De Alvaré 1988); and people are seemingly to use only a password to secure many different keys to remember them easily, therefore if this key is leaked, an adversary can break the security of many systems.

In order to avoid this limitation, a combination of an individual's conventional biometrics and cryptography have been proposed for cryptographic key generation purpose. The reason behind these proposals is that biometric features extracted from an individual contain biometric information that is unique and repeatable. Conventional biometrics include fingerprint (Soutar & Tomko 1996, Uludag et al. 2004), face (Goh & Ngo 2003), voice (Monrose et al. 2001, Carrara & Adams 2010), handwriting (Ballard, Kamara, Monrose & Reiter 2008, Vielhauer & Steinmetz 2004), signature (Feng & Choong Wah 2002), and iris (Hao et al. 2006). These conventional biometrics, unlike passwords, are beneficial. They are very difficult to be lost, forgotten, stolen, changed or compromised. Moreover, generated keys are believed to be unguessable and reproducible.

Although these biometrics have been widely used in security systems, they have some disadvantages. Firstly, cryptographic keys need to be changed to avoid key compromise or to be used in different purposes such as bank account and email account. However, conventional biometrics are difficult to change, because they are individually inherent. Moreover, they can be copied or mimicked. For example, face and iris information can be photographed, voice can be recorded or changed when the user is sick, and handwriting may be mimicked (Matyáš & Říha 2010). Lastly, an adversary can be easy to capture them by forcing the legitimate user such as threatening with a gun. These disadvantages require a better biometric modality for security systems.

In recent years, human electroencephalography (EEG) has emerged as an alternative of conventional biometric modalities (Marcel & Del Millan 2007, Nguyen et al. 2012). EEG is similar to these conventional biometrics that has unique individual information. In particular, it has been used for person identification (Palaniappan & Ravi 2003, Poulos et al. 1999). Moreover, using EEG can avoid the limitations of other conventional biometrics. EEG is believed to be random (Sanei & Chambers 2013), and hence it is very useful to generate different keys. Unlike conventional types of biometrics, EEG has the following characteristics that are impossible to be faked or compromised (Marcel & Del Millan 2007):

a) EEG is confidential, because it corresponds to a secret mental task which cannot be observed.

b) EEG signal is very difficult to mimic, because mental tasks are person dependent.

c) EEG is brain wave so it is almost impossible to steal, because the brain activity is sensitive to the user's stress and mood. The user cannot be forced to reproduce the same EEG signal while he is under stress.

d) EEG signals require alive person for recording by nature.

This motivates us to investigate the use of EEG signals in cryptographic key generation. Although EEG has been used for person identification, authentication and classification, there is a little work on using EEG to generate cryptographic keys. Therefore in this paper, we propose a new EEG-based cryptographic key generation system that exploits the paralinguistic feature extraction technique to extract EEG features, and a key generation algorithm to generate cryptographic keys from EEG signals. Our experimental results show that the proposed cryptographic key generation system using the gamma bandwidth of EEG signals provides good performance comparing with other biometrics-based cryptographic key generation systems.

This paper is constructed as follows. The next section presents related works. Section 3 presents the paralinguistic feature extraction technique for EEG signals. We describe the key generation method in Section 4, and the EEG datasets used for our experiments in Section 5. The main section 6 proposes an algorithm to generate a cryptographic key from EEG signals. Finally, we present the experimental results and compare the results with other biometrics in Section 7 and conclude the paper in Section 8.

2 Related works

Biometrics include physiological and behavioral characteristics of each individual. While physiological biometric features measure biological traits of an individual including fingerprints, face, iris or retina, behavioral features measure the way of performing actions from an individual. For example, how to speak, write or think are all behavioral features. Behaviors are more useful than physiologies because they change when changing actions, and lead to make keys generated differently.

In solving with behavioral characteristics, Monrose et al. had a contribution to discover features which distinguish between individuals (Monrose et al. 1999). They created a global threshold for each feature, and a bit for a user would be 0 if his/her extracted feature was below the global threshold or 1 otherwise. This idea was first applied for keystroke latencies. The authors showed that an adversary needed to search more than 2^{15} times for keys than using only passwords. They then applied to voice of a spoken password with a better experimental result in generating keys by increasing its entropy to 46 bits, and decreasing its false reject rate to 20%.

An alternative approach called Fuzzy Cryptography was proposed to generate cryptographic keys from biometrics and noisy data. The first work (Juels & Wattenberg 1999) in this approach proposed "fuzzy commitment scheme" on an iris code with a random key to create a key. Another work was fuzzy vault (Juels & Sudan 2002) that was lately used to construct securely fuzzy extractors (Dodis et al. 2004, Dodis & Smith 2005, Boyen 2004). However, these fuzzy extractors used biometric features as non-uniform distributed inputs to error correction algorithms to generate keys. In addition, the error-correction algorithms did not know how correct errors created from behavioral features. Fuzzy extractors also had a requirement for high value of min-entropy from biometric data, but did not show how to choose features for this requirement.

On aspect of attacks to biometrics systems, there are some efforts concerned with the security of templates generated from biometric data. Jain et al. (Jain et al. 2005) presented some possible threats that exploited the weakness of stored templates to discover information of original biometrics. They also described some practical techniques to strengthen the protection of template security, for example using watermarking or stenography. Other attacks called hill-climbing methods can be performed based on fingerprint (Uludag & Jain 2004), face (Adler 2004), or handwriting (Yamazaki et al. 2005). In (Ballard, Kamara & Reiter 2008), Ballard et al. reconsidered the requirements of biometric cryptographic key generations (BKG). They argued that a BKG is secure if biometric data is random and secret, and generated key is random. They also proposed a heuristic method that is called Guessing Distance to determine the number of guesses for discovering keys that can apply to attack (Hao et al. 2006) with an opportunity of 22% to regenerate the key at the first attempt. To combat this attack and adapt their demands, Ballard et al. (Ballard, Kamara, Monrose & Reiter 2008) proposed an algorithm to generate cryptographic keys called Randomized Biometric Templates (RBTs). The results on a handwriting database was impressive to show that 40% of all users can generate keys being 2^{30} times stronger than using passwords only. This method has attractive advantages. It can be applied to other modalities of biometrics including voice (Carrara & Adams 2010). Moreover, it is believed that this method protects a template from hill-climbing attacks (Ballard 2008). This method can also deal with a feature that is not in Monroses method to increase entropy and decrease false accept rates. This method is lastly computed efficiently using block ciphers and hash functions.

On the idea of generating strong keys from brain waves, Ravi et al. (Ravi et al. 2007) proposed to extract features from gamma signal for specific channels of EEG signals. The authors generated a 61-bit key that applied to both encryption and compression purposes. However, the authors did not evaluate entropy of the generated key, and the database used was only 40 EEG signals obtained from 10 subjects that was seemingly not large enough.

3 Paralinguistic Feature Extraction

It has been found that EEG signals are non-stationary and their characteristics are changing differently over the long period of time to reflect the variations of brain activities. Although this property makes EEG signals being a good modality of biometrics for generating different cryptographic keys, the challenge is what EEG features are reliable and how to extract them to generate keys correctly. Study on speech signals showed that they are non-stationary, but their characteristics vary slowly in a sufficiently short time between 5 and $100ms$. In other words, speech signals are quasi-stationary and speech features can be used for key generation (Monrose et al. 2001). We have found that EEG signals are also quasi-stationary (Sanei & Chambers 2013) and they have some similar characteristics to speech signals. Therefore we propose to use paralinguistic feature extraction techniques to extract features from EEG signals. The popular paralinguistic features are Perceptual Linear Prediction (PLP), Mel-Frequency Cepstral Coefficients (MFCCs) and Spectral Energy. We start with the definition of speech frame which is the product of a shifted window function and the speech signal $s(n)$ of sample n (Deller Jr et al. 1993):

$$f_s(n;m) = s(n)w(m-n) \quad (1)$$

where $w(m-n)$ is a function for a window of length

$N_w = m - n$ ending at sample $m, m > n$. The Hamming window is normally used so that discontinuities at the window edges are attenuated.

3.1 Perceptual Linear Predictive

PLP was firstly proposed by Hermansky et al. (Hermansky 1990) and widely used in speech recognition. This technique was reported to be more robust if there is a mismatch between training and test data (Woodland et al. 1996). The technique is based on three ideas of psychophysical hearing that are the critical-band spectral resolution, the equal-loudness curve, and the intensity-loudness power law. The most significance of this technique is that it can be computed efficiently and results in low-dimension coefficients of the spectral estimate. Because this analysis was still sensitive to slow changes in speech, Hermansky and Morgan improved the PLP technique to overcome this limitation (Hermansky & Morgan 1994). They proposed RASTA-PLP that can be briefly described as follows (computing formulations in (Hermansky 1990)):

1. Computing the critical-ban power spectrum using PLP technique
2. Transforming the amplitude of spectral by a compressing static nonlinear transformation.
3. Filtering the time trajectory of each spectral component
4. Transforming this speech by expanding static nonlinear transformation
5. Simulating the power law of hearing by multiplying the power of 0.33
6. Computing the spectrum by an all-pole model using PLP technique.

3.2 Mel-Frequency Cepstral Coefficients

Another popular feature extraction technique is Mel-Frequency Cepstral Coefficient (MFCC). MFCC has been widely used in speech signal processing with its advantages for presenting a short representation of speech as a low-dimension vector of features. The technique was first proposed by Bridle and Brown (Bridle & Brown 1974), and developed further by Mermelstein (Mermelstein 1976). In some cases, this technique performed better than the PLP analysis (Hönig et al. 2005).

MFCCs with an order of d are calculated from the log filterbank amplitudes a_j using Discrete Cosine Transform as follows (Steve et al. 2009):

$$c_i = \sqrt{\frac{2}{T}} \sum_{j=1}^{n} a_j \cos(\frac{\pi i}{T}(j - 0.5)) \qquad (2)$$

where T is the number of filterbank channels, c_i are the cepstral coefficients, and $i = 1, 2, \ldots, d$.

3.3 Energy

This feature is computed from amplitude as the average of signal energy. It means that for speech samples $s(n)$, the short term energy of a speech frame ending at m is (Ververidis & Kotropoulos 2006)

$$E_s(m) = \frac{1}{N_w} \sum_{n=m-N_w+1}^{m} f_s(n; m)^2 \qquad (3)$$

4 Cryptographic Key Generation

We used the method to generate a cryptographic key from handwriting data proposed by Ballard et al. (Ballard 2008). Before presenting algorithms in this method, we introduce some notations and cryptographic primitives.

4.1 Notation

Let π be a password of an user and $x \xleftarrow{R} X$ be the uniform selection of x at random from a set X, and $x \leftarrow A$ is to show that x is an output of algorithm A. Let $[a, b]_k$ be a set of integers selected between a and b according to k: $\{a + ik : i \in [0, \lfloor (b - a/k) \rfloor]\}$. E and D are two cryptographic primitives which are encryption and decryption algorithms, respectively.

Four cryptographic hash functions are used: H_0 and H_1 are two functions that map a password in a set of passwords into two different elements, H_{ver} is used to generate a token to check whether a generated key is correct, and H_{key} is to generate a cryptographic key.

Lastly, let $B = \{\beta_1, \ldots, \beta_{M+1}\}$ be a set of $(M+1)$ EEG samples from a user, $\Delta = 1 + max_i(\delta_i)$, and $\Phi = \{\phi_1, \ldots, \phi_N\}$ be a set of N feature vectors extracted from B that will be presented in the next section.

4.2 Algorithms

This section presents algorithms from the Ballard's method. The purpose of this method is to generate templates used for cryptographic key generation from legitimate users that prevents adversaries to discover biometric inputs. The method uses cryptographic primitives such as encryption algorithms and hash functions to protect the templates from attacks. The method also uses a vector quantization technique for error correction to overcome small variations between different acquisitions of the same biometric to regenerate keys precisely. This technique is shown to be useful for voice (Chang et al. 2004) and for EEG as well. This method comprises of three algorithms: RBT Setup modification, Enrollment, and KeyGen algorithms.

Algorithm 1 Modification of RBT Setup algorithm

Input: Feature set $\Phi = \{\phi_1, \ldots, \phi_N\}$ and sample set $\{\beta_1, \ldots, \beta_M\}$ of a user.
Output: Quantization thresholds $\delta_1, \ldots, \delta_N$

1. For $i = 1 \ldots N$
   (a) Sorting $\phi_i(\beta_1), \ldots, \phi_i(\beta_M)$ in ascending order
   (b) $\lambda_i = mean(\phi_i(\beta_1), \ldots, \phi_i(\beta_M))$
   (c) $t_i = max(\phi_i(\beta_M) - \lambda_i, \lambda_i - \phi_i(\beta_1))$
   (d) $\epsilon_i = \frac{1}{M-1} \sum_{k=1}^{M-1} (\phi_i(\beta_{k+1}) - \phi_i(\beta_k))$
   (e) $\delta_i = 2(t_i + \epsilon_i)$
2. Return $\delta_1, \ldots, \delta_N$

5 EEG Datasets

5.1 The Alcoholism Dataset

The dataset used comes from a study to examine EEG correlations of genetic predisposition to alcoholism (Begleiter 1999). The dataset was obtained

Algorithm 2 Enrollment Algorithm

Input: Password π, sample set $\{\beta_1, \ldots, \beta_M\}$, feature set Φ, and thresholds $\delta_1, \ldots, \delta_N$
Output: Key K and template T

1. $L \leftarrow$ Permute $\{1, \ldots, N\}$
2. $k_0 \leftarrow H_0(\pi)$, $k_1 \leftarrow H_1(\pi)$
3. For $j = 0$ to $\mid L \mid -1$
   (a) $i \leftarrow L[j]$
   (b) $\mu_i \leftarrow Mean(\phi_i(\beta_1), \ldots, \phi_i(\beta_M))$
   (c) $\alpha_i \leftarrow \lfloor \mu_i - \delta_i/2 \rfloor \bmod \delta_i$ if $\mu_i \geq \delta_i/2$. Otherwise, $\lfloor \mu_i + \delta_i/2 \rfloor$
   (d) $x_i \leftarrow \max(0, \lfloor \mu_i - \delta_i/2 \rfloor)$
   (e) $\rho_i \overset{R}{\leftarrow} [\alpha_i, \Delta]_{\delta_i}$
   (f) $C_i = (E^N_{k_0}(i), E^{\Delta}_{k_1}(\rho_i))$
   (g) $K_i = i \parallel x_i$
   (h) $K \leftarrow H_{key}(\pi \parallel K_0 \parallel \ldots \parallel K_{|\Phi|-1})$
   (i) $T \leftarrow (C, v) = ((C_0, \ldots, C_{|L|-1}), H_{ver}(\pi \parallel K_0 \parallel \ldots \parallel K_{|\Phi|-1}))$
4. Return K and T

Algorithm 3 KeyGen Algorithm

Input: Template $T = (C, v)$, password π, sample β_{M+1}, and thresholds $\delta_1, \ldots, \delta_N$
Output: Key K or $\perp$

1. $k_0 \leftarrow H_0(\pi)$, $k_1 \leftarrow H_1(\pi)$
2. For $j = 0$ to $\mid C \mid -1$
   (a) $i \leftarrow D^N_{k_0}(C[j][0])$
   (b) $\alpha_i \leftarrow D^N_{k_1}(C[j][1])$s
   (c) $x_i \leftarrow \max_{x \in 0 \cup [\alpha_i, \phi_i(\beta_{M+1})]_{\delta_i}} x$
   (d) $K_i = i \parallel x_i$
   (e) if $H_{ver}(\pi \parallel K_0 \parallel \ldots \parallel K_i) = v$ then $K = H_{key}(\pi \parallel K_0 \parallel \ldots \parallel K_i)$
   (f) Return K
3. Return $\perp$

from the University of California, Irvine Knowledge Discovery in Databases (UCI KDD) Archive, and consisted of EEG recordings of 122 alcoholic and control subjects. Each of these subjects was measured by placing 64 electrodes on their scalps sampled at 256 Hz for one second. In the recording stage, each subject was exposed to either a single stimulus (S1) or to two stimuli (S1 and S2) which were pictures of objects chosen from the 1980 Snodgrass and Vanderwart picture set. When two stimuli were shown, they were presented in either a matched condition where S1 was identical to S2 or in a non-matched condition where S1 differed from S2. The dataset had three versions where the large dataset contains data for 10 alcoholic and 10 control subjects, and the full dataset contains data of 77 alcoholic and 45 control subjects. We used both the large dataset and the full dataset for our experiment. The large dataset includes two data for training and testing purpose already, but has a small number of subjects, while the full dataset contains a large number of subjects for our algorithm.

5.2 The Graz IIIa dataset

The Graz IIIa dataset (Schlögl & Pfurtscheller 2005) comes from the Department of Medical Informatics, Institute of Biomedical Engineering, Graz University of Technology for motor imagery classification problem in BCI Competition in 2005. The dataset contains EEG recordings from 3 subjects. Each of them is guided to do cued motor imagery with 4 classes including left hand, right hand, foot and tongue. The experiments comprise of more than 6 runs with 40 trials each. The recording was made with an EEG amplifier of 64 channels from Neuroscan at 250 Hz in a time length of $7s$ for each trial.

6 The Proposed EEG Cryptographic Key Generation System

Our proposed system of cryptographic key generation from EEG signals is shown in Figure 1. Let u be a number of dimension for the best matching unit (BMU) algorithm's output, and c be a number of clusters in the clustering analysis. We use the following metrics FAR (false reject rate), FRR (false accept rate), and EER (equal error rate) where FAR = FRR to evaluate our method. The algorithm includes three phases: preprocessing, feature extraction, key generation algorithm as follows.

6.1 Preprocessing

Initially, we arrange each epoch of EEG signals into a matrix, in which its columns are a number of channels (64 for the Alcoholism, and 60 for Graz IIIa, respectively), and its rows are a number of samples per second (256 for the Alcoholism, and 250 for Graz IIIa, respectively). The ith column comprises of the signal observed from the i^{th} channel, and the j^{th} row comprises of the samples observed at the time point jth for all channels.

6.1.1 Channel Selection

We experimented our method by combining all of EEG channels. This selection aimed to produce a cryptographic key that contained the maximum amount of information achieved from subjects. The method may delete theoretically the irrelevant channels that contained insignificant information, and kept the relevant channels which contributes the most

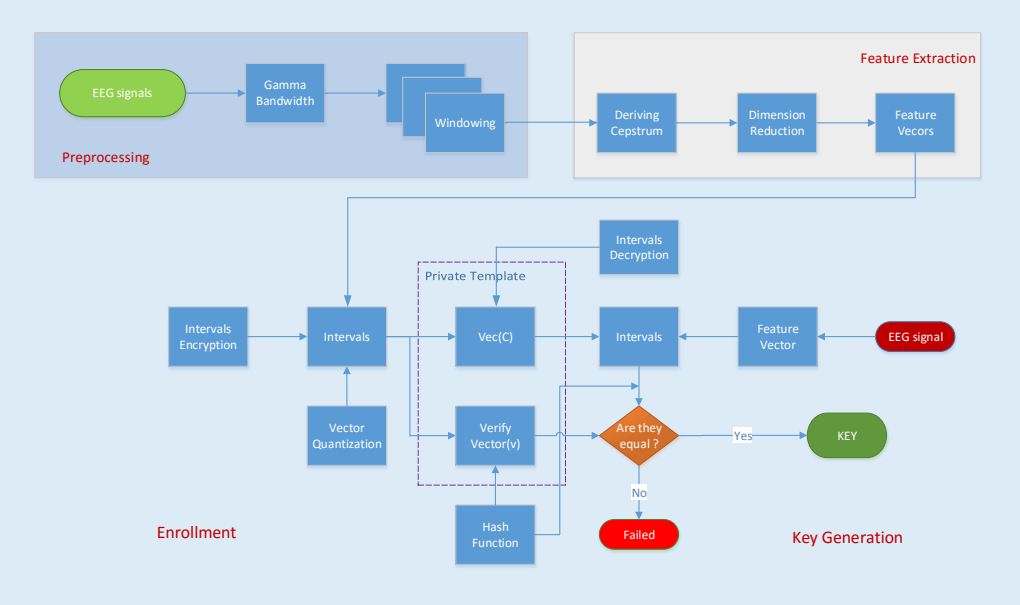
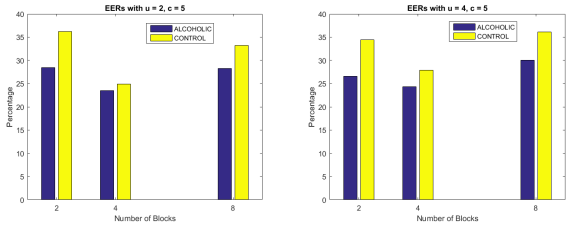


Figure 1: EEG-based Cryptographic Key Generation Algorithm



(a) (b)

Figure 2: EERs with (a): $u = 2$ and $c = 5$, (b): $u = 4$ and $c = 5$

significant information for key generation. However, this leads to a limitation of how to select relevant, or irrelevant channels. Therefore, this approach is advantageous that we do not need any assumptions about the selection of relevant channels.

Then, we experimented our algorithm in detail in the next section to select a number of blocks derived from by dividing a matrix of epoch. For example, each block was a matrix of 256×16 if 4 blocks was used for the Alcoholism dataset. This selection was based on a trade-off between a number of extracted features and EERs. We proposed to use 4 blocks, and studied the impact of this selection by implementing our method for three cases of 8, 4 and 2 blocks. The dataset used was the large database of Alcoholism dataset, because it contained the training and testing data with the same subjects. Experimental results for 8 blocks showed that both a number of features generated and EERs were higher than for 4 blocks where the best EER was nearly 25% for alcoholic, and was 30.82% for control. For a 2-block case, although a number of features generated was lower than for 4 blocks, EERs were very different between alcoholic and control subjects that were 20.03% and 30.08%, respectively. For illustration, the 2 showed results of our method for two choices of parameters. Therefore, we decided to use this approach for the remainder of experiments.

6.1.2 Frequency Band Selection

In (Tzelepi et al. 2000), a brain wave can be divided into 5 groups of frequency activity bands: Delta (4 Hz), Theta ($4-7$ Hz), Alpha ($8-12$ Hz), Beta ($12-30$ Hz), and Gamma ($30-100$ Hz), where the Gamma band occurred when subjects are doing cognitive and motor tasks. Moreover, Ong et al (Ong et al. 2005) suggested to use it for classification of alcoholic and non-alcoholic subjects. Thus, we used the Gamma band in our method and applied bandpass filters for extract it from EEG signals.

6.1.3 Window Selection

Each Gamma bandwidth was split into short time interval windows of 30ms with overlapping of 10ms each to exploit its quasi-stationary characteristic for extracting repeatable features as being used in (Monrose et al. 2001) for speech signals.

6.2 Feature Extraction

6.2.1 MFCC Derivation

After preprocessing, we used speech-based feature extraction methods to extract EEG features. First, we computed PLP cepstral coefficients as shown in section 3.1 for an order of 13. Then, we compute 13 ceptrals coefficients of MFCCs by an equation 2), and one spectral energy coefficient (equation 3). Lastly, all these coefficients were concatenated to derive a frame vector of 27 coefficients for each window. As a result, we obtain a matrix for each block where its rows were a number of windows, and its columns were 27.

6.2.2 Dimension Reduction

To reduce dimension this matrix, we considered its rows as an input of BMU algorithm to achieve u rows, and its columns as an input of the clustering analysis to get c clusters. Then, a matrix of $c \times u$ was generated, and it was rearranged into an array to form a vector of $c \times u$ coordinates for each block. Lastly, these 4 vectors were concatenated to generate a $4uc$-coordinate feature vector.

Different users have different brain activity, so their EEG signals may differ from others. This motivated to use this dimension reduction approach to hopefully extract common features that are repeatable for all users, even if these features are not repeatable for a user.

6.2.3 Feature Space

To summarize, every EEG signal is represented by a feature vector given by:

$$\Phi = (\phi_1, ..., \phi_N) \text{ with } N = 4uc \quad (4)$$

This process was repeated $M+1$ times for all samples $\beta_1, \ldots, \beta_{M+1}$ to get $M+1$ feature vectors.

6.3 Key Generation Algorithm

6.3.1 Enrollment

Input: Password π, samples $\beta_1, ..., \beta_M$, features Φ, and thresholds $\delta_1, ..., \delta_N$

(a) Computing the quantization thresholds $\delta_1, ..., \delta_N$ (algorithm 1) for features $(\phi_1, ..., \phi_N)$ from samples $\beta_1, ..., \beta_M$.

(b) Correcting errors using the vector quantization technique. Its purpose is to correct a user's samples into a single, repeatable value by partitioning EEG features into intervals, and performed as follows (see step 5-8 in algorithm 2). First, we randomly select an index i from the set of indices $1, \ldots, N$ then compute μ_i as the mean of $\phi_i(\beta_1), \ldots, \phi_i(\beta_M)$ for each index i. Then, the range $R_i = [0, r_i]$ of each feature ϕ_i is partitioned around μ_i into intervals $([\alpha_i + k\delta_i, \alpha_i + (k+1)\delta_i, k \in [0, \lfloor (r_i/k) \rfloor] - 1)$ of length δ_i by computing a lowest boundary: $\alpha_i = \lfloor \mu_i - \delta_i/2 \rfloor \bmod \delta_i$ if $\mu_i \geq \delta_i/2$, otherwise $\alpha_i = \lfloor \mu_i + \delta_i/2 \rceil$. Finally, an offset ρ_i of quantization is randomly chosen in the range $[\alpha_i, \Delta]_{\delta_i}$ with $\Delta = \max(\delta_i)_i$, and $x_i = \max(0, \lfloor \mu_i - \delta_i/2 \rfloor)$ is computed as the border of partition containing μ_i.

(c) Generating a key and a template. After error correction process, a key K and a template $T = (C, v)$ are computed (see step 10-13 in algorithm 2, in which C is used to store the feature indexes and the quantization offsets, and v is for verifying purpose.

Output: Key K and template T

6.3.2 Key Generation

Input: Template $T = (C, v)$, passwords π, sample β_{M+1}, and thresholds $\delta_1, ..., \delta_N$

(a) Extracting features $\phi_1(\beta_{M+1}), ..., \phi_N(\beta_{M+1})$ from the sample β_{M+1} in the testing data.

(b) Decrypting the vector C to extract the quantization offsets and feature indices, and computing the largest boundary of partition that is smaller than $\phi_i(\beta_{M+1})$ for each $i \in [1, N]$, then concatenating these values to produce a temporary key (see Steps 2-5 in algorithm 3).

(c) Hashing the temporary key and comparing the result with v. If they are equal, the key is output, otherwise the algorithm fails (see Steps 7-10 in Algorithm 3).

Output: Key K or $\perp$

7 Experiments and Results

7.1 Experimental methodology

Our experiment was performed using the resource of Matlab code (Ellis 2005) (from the Lab of Columbia University) to compute PLP, MFCCs coefficients and spectral energy, and the SOM toolbox version 2.0 coded in Matlab from the Laboratory of Computer and Information Science (CIS) of the Helsinky University of Tecnology (Alhoniemi et al. 2000) to find the best matching unit. We also used the k-means algorithm to do clustering analysis.

We used two datasets of EEG Alcoholism databases, namely the large and full datasets. For the large dataset, it has been already divided into the 30-trial training and 30-trial testing datasets. For the full dataset, we randomly selected 60 trials to generate a data and used the Cross-Validation technique with the ratio of 3 : 1 for the training and testing data.

For the Graz IIIa dataset, we extracted four datasets from cued motor imagery tasks of movements with 4 classes (right hand, left hand, foot and tongue). Then, these datasets containing 30 samples each were also used to do the Cross-Validation technique with the ratio of 2 : 1 for the training and testing data.

7.2 Experimental Results

We present results for the algorithm depending on the choices of two parameters u and c to explore the trade-off between a number of features used for key generation and the forgery resistance. If a larger number of features is used, it will result to lower theoretical entropy and increase an opportunity of forge by an adversary. On the other hand, using less features will make them be difficult to forge and rise theoretical entropy up. We used two metrics FRR and FAR to evaluate the ability of forgery resistance.

To compute the FRR for each subject, we used the training data to generate features and a template, then used the testing data to create keys with the template and compute FRR as the number of features that are not being successfully regenerated. This process was performed on all subjects, and the FRR was calculated as the average of all runs.

To compute FAR for each subject, we use the training data to generate a template and a key. We then used all testing data of all other subjects to test the ability to regenerate the correct key with the template. We performed this process for all subjects and measured the FAR as the average of all runs.

We evaluated the selection of parameters u and c based on two factors: the number of features and EER. By trading off these factors, two sets of parameters are chosen: $u = 2$, $c = 5$ (40 features) and $u = 4$, $c = 5$ (80 features).

Table 1 provides the EERs for these parameters, and a comparison with the Ballard's method. We compared our method with RBTs - a state of the art method for biometric-based key generation that can be applied to EEG signals. As shown in Table 1, the best result of EERs is approximately 18% achieved for the alcoholic subjects in the full dataset. On the other hand, the worst achievement is nearly 28% for the control subjects in the large dataset. This result is similar to the best result of RBTs which is 16%. Moreover, the EERs for the alcoholic and control subjects in datasets are closely, and it results to the observation that cryptographic key generation based on brain waves is less sensitive to using alcoholic. Furthermore, Table 2 illustrates EERs of a mixed data

Table 1: EERs for Alcoholism data of the proposed method in comparison with the RBTs of Ballard

The Proposed Method								RBTs
$u = 2, c = 5$				$u = 4, c = 5$				
Large Data		Full Data		Large Data		Full Data		
Alcoholic	Control	Alcoholic	Control	Alcoholic	Control	Alcoholic	Control	
23.54%	24.86%	18.01%	20.93%	24.37%	27.93%	18.18%	23.46%	17.7%

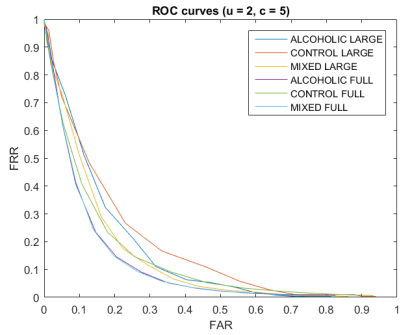


Figure 3: Performance of the algorithm for Alcoholism data with $u = 2$ and $c = 5$

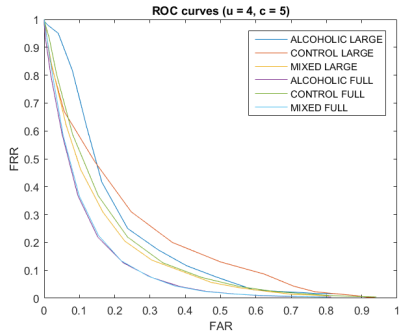


Figure 4: Performance of the algorithm for Alcoholism data with $u = 4$ and $c = 5$

of alcoholic and control subjects. The result shows that EERs for alcoholic subjects are less sensitive by control subjects, and vice versa. The ROC curves in figure 3 and 4 show more details of the implementations of our method.

Table 3 provides the EERs of Graz IIIa dataset, and the results are higher than the EERs of Alcoholic. The reason may be that there is a high variation in data (4 imagery tasks that occurs randomly), and the training data is not enough. The data for training may come from different runs to build a good performance in testing.

To sum up, Table 4 presents a summary of implementations for cryptographic key generation from biometric data in comparison. Although our method is not as good as iris, but it is similar to voice and handwriting, and better than other biometrics.

Table 2: EERs for the mix of Alcoholism data

$u = 2, c = 5$		$u = 4, c = 5$	
Large Data	Full Data	Large Data	Full Data
20.65%	17.89%	21.96%	18.37%

8 Conclusion and Future work

We have proposed a cryptographic key generation system that used paralinguistic feature extraction techniques to extract EEG features, and Ballard's algorithm to generate strong cryptographic keys. Our experiments were performed on the EEG Alcoholism and the Graz IIIa dataset. The empirical evaluation shows that by choosing properly parameters would result to a reliable cryptographic key generation system. For future work, other feature extraction techniques and other datasets will be applied to evaluate further our proposed cryptographic key generation system.

References

Adler, A. (2004), Images can be regenerated from quantized biometric match score data, *in* 'Electrical and Computer Engineering, 2004. Canadian Conference on', Vol. 1, IEEE, pp. 469–472.

Alhoniemi, E., Himberg, J., Parhankangas, J. & Vesanto, J. (2000), 'Som toolbox', *Online: http://www. cis. hut. fi/projects/somtoolbox* .

Ballard, L. K. (2008), 'Robust techniques for evaluating biometric cryptographic key generators', *Johns Hopkins University, Baltimore, MD* .

Ballard, L., Kamara, S., Monrose, F. & Reiter, M. K. (2008), Towards practical biometric key generation with randomized biometric templates, *in* 'Proceedings of the 15th ACM conference on Computer and communications security', ACM, pp. 235–244.

Ballard, L., Kamara, S. & Reiter, M. K. (2008), The practical subtleties of biometric key generation., *in* 'USENIX Security Symposium', pp. 61–74.

Barker, E., Barker, W., Burr, W., Polk, W., Smid, M., Gallagher, P. D. et al. (2012), 'Nist special publication 800-57 recommendation for key management–part 1: General'.

Begleiter, H. (1999), 'Eeg alcoholism database', *Online: https://kdd.ics.uci.edu/databases/eeg/eeg.data.html* .

Boyen, X. (2004), Reusable cryptographic fuzzy extractors, *in* 'Proceedings of the 11th ACM conference on Computer and communications security', ACM, pp. 82–91.

Bridle, J. & Brown, M. (1974), 'An experimental automatic word recognition system', *JSRU Report* **1003**(5).

Carrara, B. & Adams, C. (2010), You are the key: generating cryptographic keys from voice biometrics, *in* 'Privacy Security and Trust (PST), 2010 Eighth Annual International Conference on', IEEE, pp. 213–222.

Table 3: EERs of Graz IIIa dataset with imagination of movements

$u = 2, c = 5$					$u = 4, c = 5$				
Left	Right	Foot	Tongue	Mixed	Left	Right	Foot	Tongue	Mixed
27.70%	29.50%	27.64%	27.38%	32.88%	30.92%	25.75%	31.68%	33.67%	36.30%

Table 4: Summary of biometrics implementations for cryptographic key generation

Biometrics (Refs)	Bit	FRR	FAR
Keystroke (Monrose et al. 1999)	12	48%	–
Voice (Monrose et al. 2001)	46	20%	–
Signature (Feng & Choong Wah 2002)	40	28%	1.2%
Fingerprint (Clancy et al. 2003)	69	30%	–
Face (Goh & Ngo 2003)	–	–	–
Iris (Hao et al. 2006)	140	0.47%	0%
Handwriting (Ballard 2008)	–	17.7%	17.7%
EEG	–	18.01%	18.01%

Chang, Y.-J., Zhang, W. & Chen, T. (2004), Biometrics-based cryptographic key generation, *in* 'Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on', Vol. 3, IEEE, pp. 2203–2206.

Clancy, T. C., Kiyavash, N. & Lin, D. J. (2003), Secure smartcardbased fingerprint authentication, *in* 'Proceedings of the 2003 ACM SIGMM workshop on Biometrics methods and applications', ACM, pp. 45–52.

De Alvaré, A. M. (1988), How crackers crack passwords or what passwords to avoid, Technical report, Lawrence Livermore National Lab., CA (USA).

Deller Jr, J. R., Proakis, J. G. & Hansen, J. H. (1993), *Discrete time processing of speech signals*, Prentice Hall PTR.

Dodis, Y., Reyzin, L. & Smith, A. (2004), Fuzzy extractors: How to generate strong keys from biometrics and other noisy data, *in* 'Advances in cryptology-Eurocrypt 2004', Springer, pp. 523–540.

Dodis, Y. & Smith, A. (2005), Correcting errors without leaking partial information, *in* 'Proceedings of the thirty-seventh annual ACM symposium on Theory of computing', ACM, pp. 654–663.

Ellis, D. P. W. (2005), 'PLP and RASTA (and MFCC, and inversion) in Matlab'.

Feldmeier, D. C. & Karn, P. R. (1989), Unix password security-ten years later, *in* 'Advances in CryptologyCRYPTO89 Proceedings', Springer, pp. 44–63.

Feng, H. & Choong Wah, C. (2002), 'Private key generation from on-line handwritten signatures', *Information Management & Computer Security* **10**(4), 159–164.

Goh, A. & Ngo, D. C. (2003), Computation of cryptographic keys from face biometrics, *in* 'Communications and Multimedia Security. Advanced Techniques for Network and Data Protection', Springer, pp. 1–13.

Hao, F., Anderson, R. & Daugman, J. (2006), 'Combining crypto with biometrics effectively', *Computers, IEEE Transactions on* **55**(9), 1081–1088.

Hermansky, H. (1990), 'Perceptual linear predictive (plp) analysis of speech', *the Journal of the Acoustical Society of America* **87**(4), 1738–1752.

Hermansky, H. & Morgan, N. (1994), 'Rasta processing of speech', *Speech and Audio Processing, IEEE Transactions on* **2**(4), 578–589.

Hönig, F., Stemmer, G., Hacker, C. & Brugnara, F. (2005), Revising perceptual linear prediction (plp)., *in* 'INTERSPEECH', pp. 2997–3000.

Jain, A. K., Ross, A. & Uludag, U. (2005), Biometric template security: Challenges and solutions, *in* 'Signal Processing Conference, 2005 13th European', IEEE, pp. 1–4.

Juels, A. & Sudan, M. (2002), A fuzzy vault scheme, *in* 'In International Symposium on Information Theory (ISIT'.

Juels, A. & Wattenberg, M. (1999), A fuzzy commitment scheme, *in* 'Proceedings of the 6th ACM conference on Computer and communications security', ACM, pp. 28–36.

Klein, D. V. (1990), Foiling the cracker: A survey of, and improvements to, password security, *in* 'Proceedings of the 2nd USENIX Security Workshop', pp. 5–14.

Marcel, S. & Del Millan, J. R. (2007), 'Person authentication using brainwaves (eeg) and maximum a posteriori model adaptation', *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **29**(4), 743–752.

Matyáš, V. & Říha, Z. (2010), Security of biometric authentication systems, *in* 'Computer Information Systems and Industrial Management Applications (CISIM), 2010 International Conference on', IEEE, pp. 19–28.

Mermelstein, P. (1976), 'Distance measures for speech recognition, psychological and instrumental', *Pattern recognition and artificial intelligence* **116**, 374–388.

Monrose, F., Reiter, M. K., Li, Q. & Wetzel, S. (2001), Cryptographic key generation from voice, *in* 'Security and Privacy, 2001. S&P 2001. Proceedings. 2001 IEEE Symposium on', IEEE, pp. 202–213.

Monrose, F., Reiter, M. K. & Wetzel, S. (1999), 'Password hardening based on keystroke dynamics'.

Nguyen, P., Tran, D., Huang, X. & Sharma, D. (2012), A proposed feature extraction method for eeg-based person identification, *in* 'International Conference on Artificial Intelligence'.

Ong, K.-M., Thung, K.-H., Wee, C.-Y. & Paramesranle, R. (2005), Selection of a subset of eeg channels using pca to classify alcoholics and non-alcoholics, *in* 'Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference (Shanghai, China, 2005)'.

Palaniappan, R. & Ravi, K. (2003), A new method to identify individuals using signals from the brain, *in* 'Information, Communications and Signal Processing, 2003 and Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on', Vol. 3, IEEE, pp. 1442–1445.

Poulos, M., Rangoussi, M., Chrissikopoulos, V. & Evangelou, A. (1999), Person identification based on parametric processing of the eeg, *in* 'Electronics, Circuits and Systems, 1999. Proceedings of ICECS'99. The 6th IEEE International Conference on', Vol. 1, IEEE, pp. 283–286.

Ravi, K., Palaniappan, R., Eswaran, C. & Phon-Amnuaisuk, S. (2007), Data encryption using event-related brain signals, *in* 'Conference on Computational Intelligence and Multimedia Applications, 2007. International Conference on', Vol. 1, IEEE, pp. 540–544.

Sanei, S. & Chambers, J. A. (2013), *EEG signal processing*, John Wiley & Sons.

Schlögl, A. & Pfurtscheller, G. (2005), 'Dataset iiia: 4-class eeg data'.

Soutar, C. & Tomko, G. (1996), Secure private key generation using a fingerprint, *in* 'Cardtech/Securetech Conference Proceedings', Vol. 1, pp. 245–252.

Steve, Y., Gunnar, E., MARK, A. et al. (2009), 'The htk book (for htk version 3.4)', *Steve Young, Gunnar Evermann, Dan Kershaw, Gareth Moore, Julian Odell, Dave Ollason, Valtcho Valtchev, Phil Woodland* .

Tzelepi, A., Bezerianos, T. & Bodis-Wollner, I. (2000), 'Functional properties of sub-bands of oscillatory brain waves to pattern visual stimulation in man', *Clinical Neurophysiology* **111**(2), 259–269.

Uludag, U. & Jain, A. K. (2004), Attacks on biometric systems: a case study in fingerprints, *in* 'Electronic Imaging 2004', International Society for Optics and Photonics, pp. 622–633.

Uludag, U., Pankanti, S., Prabhakar, S. & Jain, A. K. (2004), 'Biometric cryptosystems: issues and challenges', *Proceedings of the IEEE* **92**(6), 948–960.

Ververidis, D. & Kotropoulos, C. (2006), 'Emotional speech recognition: Resources, features, and methods', *Speech communication* **48**(9), 1162–1181.

Vielhauer, C. & Steinmetz, R. (2004), 'Handwriting: feature correlation analysis for biometric hashes', *EURASIP Journal on Applied Signal Processing* **2004**, 542–558.

Woodland, P. C., Gales, M. J. F. & Pye, D. (1996), Improving environmental robustness in large vocabulary speech recognition, *in* 'Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on', Vol. 1, IEEE, pp. 65–68.

Yamazaki, Y., Nakashima, A., Tasaka, K. & Komatsu, N. (2005), A study on vulnerability in online writer verification system, *in* 'Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on', IEEE, pp. 640–644.

Ownership protection based on optimized watermarking for biomedical and health systems in data mining

Trung Pham Duy **Dat Tran** **Wanli Ma**

Faculty of Education, Science, Technology and Mathematics
University of Canberra, ACT 2601, Australia,
Email: dat.tran@canberra.edu.au

Abstract

Data mining with descriptive and predictive power has been widely used in the biomedical and healthcare fields. The extracted hidden knowledge and patterns using data mining techniques need to be shared between their owners and data mining experts. Consequently, the ownership of the shared data becomes a challenging issue. Watermarking has been proved as a right-protection mechanism that can provide detectable evidence for the legal ownership of a shared dataset, without compromising its usability under a wide range of data mining. Watermarking is the commonly used mechanism to enforce and prove the ownership for digital data in different formats such as audio, video, image, relational database, text and software. However, issues related to time series biomedical data such as electroencephalography (EEG) or Electrocardiography (ECG) have not been addressed. This paper proposes an optimized watermarking scheme to protect the ownership for biomedical and health system in data mining. To achieve the highest possible robustness without losing watermark transparency, Particle Swarm Optimization (PSO) technique is used to optimize quantization steps to find a suitable one. The proposed scheme has been carried on electroencephalography (EEG) data and experimental results show that the proposed scheme provides good imperceptibility and more robust against various signal processing techniques and common attacks such as noise addition, low-pass filtering, and re-sampling.

Keywords: Ownership protection, EEG, Watermarking, Data mining, Particle Swarm Optimization (PSO).

1 Introduction

Data mining has emerged as a significant technology in various application for gaining knowledge from vast quantities of data (Han 2001). It can be defined as the process of finding previous unknown patterns and trends in databases and using that information to build predictive model (Kincade 1998). As its descriptive and predictive power, data mining becomes increasingly popular including in biomedical and healthcare fields (Koh et al. 2011).

Data mining can help researchers gain both novel and deep insights and can facilitate unprecedented understanding of large biomedical datasets. Data mining can uncover new biomedical and healthcare knowledge for clinical and administrative decision making as well as generate scientific hypotheses from large experimental data, clinical databases, and/or biomedical literature (Yoo et al. 2012). For example, using data mining technologies, healthcare professionals can predict health insurance fraud, underdiagnosed patients, healthcare cost, disease prognosis, disease diagnosis, and the length of stay in a hospital. In addition, they can obtain frequent patterns from biomedical and healthcarea databases, such as relationships between health conditions and a disease, relationships among diseases, and relationships among drugs.

Many large biomedical datasets are being mined to extract hidden knowledge and patterns that assist decision makers in making effective, efficient, and timely decision. This type of "knowledge-driven" data mining activity is not possible without sharing the "dataset" between their owners and data mining experts (or corporations). As a consequence, protecting ownership on the datasets is becoming relevant (Kamran & Farooq 2013).

Data sharing or information sharing is necessary for distributed systems in data mining. However, there are multiple danger zones like copyright and integrity violations of digital objects (Gupta & Raval 2012). In healthcare, each collaborator (hospital) needs to share their private local database with other collaborators in telemedicine system, telediagnosis, telesurgery, and hospital information system (Bhatnagar & Wu 2013). The sharing of medical data also exposes data holders to threat of data theft (Bertino et al. 2005). Data owners, nonetheless, also need to maintain the principal rights over the datasets that they share, which in many cases have been obtained after expensive and laborious procedures. It is necessary to have a right-protection mechanism that can provide detectable evidence for the legal ownership of a shared dataset, without compromising its usability under a wide range of data mining.

In such scenarios, the shared data might be illegal sold to third parties by an unauthorized party. In order to cater for such a situation, the data need to be right protected so that an unauthorized party might be sued in a court of law. This is only possible, if the data owner is able to prove that the illegally sold data are his property. Therefore, it is important to not only protect the privacy of patient (Alhaqbani & Fidge 2008), but also the ownership (copyright) of the medical data shared with collaborative partners or third party vendors. Therefore, it is important that medical data are right protected in a manner where ownership could unambiguously be determined (Kamran & Farooq 2012).

The important requirement regarding shared medical data is to protect data ownership (copyright). We need effective mechanisms to establish and protect the holders' rightful possession of the data. Watermarking allows the users to hide innocuous pieces of information inside the data. The value of watermarking becomes increasingly important because of the ease of data sharing particularly

through data clouds. Watermarking has many applications such as ownership identification, proof of ownership, tamper detection and leak identification. Digital watermarking techniques represent a viable solution for the problem of enforcing ownership of medical data (Bertino et al. 2005).

There are two important issues that watermarking algorithms need to address. First, watermarking scheme is required to provide trustworthy evidence for protecting the rightful ownership. The perceptual difference between the watermarked and the original documents should be unnoticeable to the human observer. Second, good watermarking scheme should satisfy the requirement of robustness and resist distortions due to common attacks. The watermark should be detectable and extractable after data manipulations were applied to the watermarked data. The watermark scheme needs to address a significant challenge related to biomedical data that is insertion of a watermark must not result in changing health and medical data of a patient to a level where a decision maker (or system) can misdiagnose the patient. If a patient is misdiagnosed, it might not only put his life on risk but also results in significantly enhancing the cost of health care. Moreover, the inserted watermark should be imperceptible to intruders and they should not be able to corrupt it by launching malicious attack.

On the design of watermarking scheme, two significant but conflicting requirements, imperceptibility and robustness, should be taken into consideration because the targeted balance between the two requirements. Hence, dealing with the trade-off as an optimization problem may adaptively achieve the specific balance for the intended application. There is no exact algorithm to choose the value of scaling factor. Most of them are based on trial-and-error method, others use fixed parameters without any optimization. Instead of using the fixed parameters, another trend is witnessed, intelligent systems and evolutionary algorithms are integrated into watermarking approach to optimize the parameters.

Actually, the performance of a watermarking scheme can be improved with artifical intelligence techniques since watermarking scheme can be regarded as an optimization problem. Artificial intelligence techniques based on objective function such as tabu search (Sriyingyong & Attakitmongcol 2006), differential evolution (DE), and genetic algorithm (GA) (Kumsawat 2010) have been demonstrated to be very effective in solving conflicting requirements of watermarking. In addition, the meta-heuristic algorithm is also chosen to find the suitable scaling factor to overcome the optimization problem. There are several kinds of meta-heuristic algorithms, e.g. genetic algorithm, ant colony, particle swarm optimization, etc. Recent studies prove that computational intelligence techniques, especially Particle Swarm Optimization (PSO) (Hassan et al. 2005) are ideally suited for solving constrained optimization problem in real-time.

PSO has the following advantages: 1) PSO is easier to implement and there are fewer parameters to adjust; 2) PSO has a more effective memory capability since every particle remembers its own previous best value as well as the neighborhood best; and 3) Because all the particles use the information related to the most successful particle, PSO is more efficient in maintaining the diversity of the swarm. In order to achieve a trade-off between imperceptibility and robustness, PSO is utilized to search for the optimal embedding parameter in this study.

Electroencephalography (EEG) is a neuroimaging technique for recording the brain's electrical potentials, which are commonly used to study the dynamics of neural information processing in the brain, and diagnose brain disorders and cognitive processes (Amin et al. 2015). EEG is also used in telemedicine and brain-computer interface (BCI) applications. Digital watermarking is one of the solutions to address this problem and is used for digital rights protection, ownership verification and security purposes (Bender et al. 1996). Preliminary research in watermarking or information hiding techniques has been developed for embedding text, images, audio or video in a host signal. However, techniques developed for these data do not transpose well to other data modalities for some reasons such as: 1) The redundancy in a time series of biomedical signals such as EEG and ECG that is less compared with an image or audio. Therefore, embedding data in time series data such as EEG or Electrocardiography (ECG) which has a low redundancy is much more difficult due to the reduced redundancy limiting possibilities of hiding data and has not been investigated; and 2) Audio signal has slow time-varying feature while EEG signal is the fast changing-time series. For the EEG data, conventional watermarking is not appropriate because of the signal characteristics and distortion problems. Therefore, the evidence strongly suggests that a new watermarking scheme should be designed.

The key objective of this paper is to design a blind and robust watermarking system aiming to effectively prevent the illegal use of the outsourced biomedical data without affecting the perceptual quality. In this paper, we propose a novel ownership protection based optimized watermarking scheme for biomedical and health system in data mining.

The main contributions of this research are as follows:

1. An appropriate efficient watermarking algorithm that is suitable for outsourced time series biomedical data such as EEG in data mining;
2. As quantization step for watermarking is signal-dependent so that a fixed value is not optimized, in our scheme, PSO is used to find the optimal quantization steps for different host EEG signals and watermarks, thus achieving an optimal balance of the contradictory watermarking requirements; and
3. An EEG watermarking detection algorithm that can extract the watermark without original EEG signals (blind watermarking scheme). A lot of space is needed for storing the original data and original watermark if non-blind watermarking scheme is used.

The effectiveness of the proposed scheme is qualified using metrics like Peak Signal Noise Ratio (PSNR), NC and BER to analyze the watermarked signal in terms of imperceptibility and robustness. Experimental results show that our proposed watermarking scheme yields a good imperceptibility and more robust against various signal processing and common attacks.

The rest of the paper is organized as follows. We describe some fundamental concepts of Singular Value Decomposition (SVD), discrete wavelet transform (DWT) and Particle Swarm Optimization Algorithm (PSO) in Section 2. In Section 3, the proposed watermarking scheme is discussed in detail. Experiments and results are presented in Section 4. Finally, the paper is ended with conclusion and future work in Section 5.

2 Background

2.1 Singular Value Decomposition (SVD)

Let $A = \{a_{ij}\}_{N\times N}$ be an $N \times N$ matrix. The SVD of matrix A is represented in the form $A = USV_T$, where U and V are orthogonal matrices, S is a diagonal matrix with nonnegative elements, and superscript T denotes matrix transposition.

The diagonal elements of S, denoted by σ_i , are called the singular values (SVs) of A and are assumed to be arranged in decreasing order $\sigma_i > \sigma_{i+1}$. The columns of

U, denoted by U_i , are called the left singular vectors, while the columns of V , denoted by V_i , are called the right singular vectors of A. The SVD has some interesting properties: (i) the sizes of the matrices for SVD transformation are not fixed, and the matrices need not be square, (ii) changing SVs slightly does not affect the quality of the signal much, (iii) the SVs are invariant under common signal processing operations, and (iv) the SVs satisfy intrinsic algebraic properties.

2.2 Discrete wavelet transform (DWT)

DWT employs extensive time window for low frequencies and short time window for higher frequencies. DWT is widely used for the time-frequency analysis of biomedical signals (Orhan et al. 2011, Jahankhani et al. 2006), especially in an EEG signal analysis due to its non-stationary characteristics. As EEG signal is the fast changing-time series with continuous random changes, we use the Haar wavelet which is more suitable for such fast changing time-series compared to Daubechies wavelets, Mexican Hat wavelets and Morlet wavelets which are better suited for smoothly changing time series (Percival & Walden 2006). In addition, the Haar wavelet is also simple, fast and exactly reversible which is necessary to reconstruct cover signal in digital watermarking. Each wavelet decomposition of the original signal halves the frequency and length of the signal. The Haar function Ψ used as the mother wavelet generates a set of wavelets as follows:

$$C_{a,b} = \sum_{N_{samp}} c(t)\Psi_{a,b}(t) \tag{1}$$

where $\Psi_{a,b}(t) = \frac{1}{\sqrt{s}}\Psi(\frac{t-\tau}{s})$, a denotes the dilation index, b the translation index, s the scale factor and τ the displacement. DWT is basically an application of set of filters resulting in and approximate C_a and fine detailed C_b representation of c.

2.3 Particle Swarm Optimization Algorithm (PSO)

PSO (Kennedy 2011) is a population-based stochastic algorithm developed for continuous optimization. In PSO, each particle which represents a potential solution will searches for optimal coordinates in the problem space. Each particle has its own set of attributes including *position*, *velocity*, and a *fitness value* which is obtained by evaluating a fitness function at its current position.

The algorithm starts with the initialization of particles with random position and velocities so that they can move in the solution space. Then, these particles search the solution space for finding better solutions. Each particle keep track of its *personal best* position found so far by storing the coordinates in the solution space. The best position found so far by any particle during any stage of the algorithm is also stored and is termed as the *global best* position. The velocity of every particle is influenced by its personal best position (autobiographical memory) and the global best position (publicized knowledge). The new position for every particle is calculated by adding its new velocity value to every component of its position vector.

Let D denote the swarm size. Each individual particle $i(1 \leq i \leq D)$ has its own position p_i and velocity v_i. These particles search for the optimal value of a given objective function iteratively, then locate their individual best positions $p_i^{best}(pbest)$ and keep track of the global position $p_{gi}^{best}(gbest)$ from all best positions through a search space. With respect to the two best values, the velocity and position of particle $i(1 \leq i \leq D)$ in iteration $t+1$ are updated by:

$$p_i(t+1) = p_i(t) + v_i(t+1) \tag{2}$$

$$\begin{aligned} v_i(t+1) = c_0 v_i(t) + c_1 r_1(p_i^{best}(t) - x_i(t)) \\ + c_2 r_2(p_{gi}^{best}(t) - x_i(t)) \end{aligned} \tag{3}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{4}$$

where c_1 and c_2 are referred as cognitive and social parameters respectively, which are positive constants, r_1 and r_2 are random numbers uniformly distributed in the range of $[0, 1]$. c_0 is the inertia weight in the range of $[0, 1]$, which controls the momentum of the particle and tunes changes in these values. These coefficients control how far a particle moves in a single iteration. v_i is the moving distance in one-step for a particle i and is limited to the range of $[v_{min}, v_{max}]$, where v_{min} and v_{max} are the minimum and maximum moving distance in one-step, respectively. The qualitative measure of the selection of PSO algorithm parameter can be found in (Trelea 2003). From the PSO equations, we can know that the trade-off is related to the PSO parameters such as c_0, $itermax$, c_1 and c_2. Based on the analysis in (Trelea 2003), the parameter couples that are close to the center of the stability triangle lead to quick convergence, while parameter couples that are close to its borders need many iterations to converge. After going through the whole process iteratively, objective evaluation function reaches the desired termination criterion.

3 The proposed EEG Watermarking Approach

The proposed approach is presented in Fig.1 including three phases: watermarking embedding, watermarking extraction and quantization step optimization with PSO.

3.1 Watermarking embedding

The steps of the EEG watermarking embedding are summarized as follows (Fig.1):

1. The watermark W is converted into a one dimensional watermark sequence of length K.
2. The original EEG signal X is decomposed up by two-level DWT using a Haar wavelet filter in order to get three sets of coefficients $D1$, $D2$ and $A2$, where $D1$, $D2$ and $A2$ represent the detailed and approximate coefficients, respectively. In our observations, larger decomposition level will not increase the watermarking robustness while it causes intensive computation. Thus we choose two as decomposition level of the Haar wavelet to trade-off between robustness and imperceptibility.
3. The sets of coefficients $A2$, corresponding to low frequency part, is divided into non overlapping K segments, so that each watermark bit is inserted into one segment. This ensure a better distribution of watermark bits over the entire EEG signal, improving thus the robustness of the watermark against different attacks.
4. Each segment is rearranged into $r \times r$ 2-D matrix block, named M_k.
5. SVD is performed to decompose each matrix M_k into three matrices: U_k, S_k, and V_k. The SVD operation is represented as follows:

$$M_k = U_k \times S_k \times V_k^T \tag{5}$$

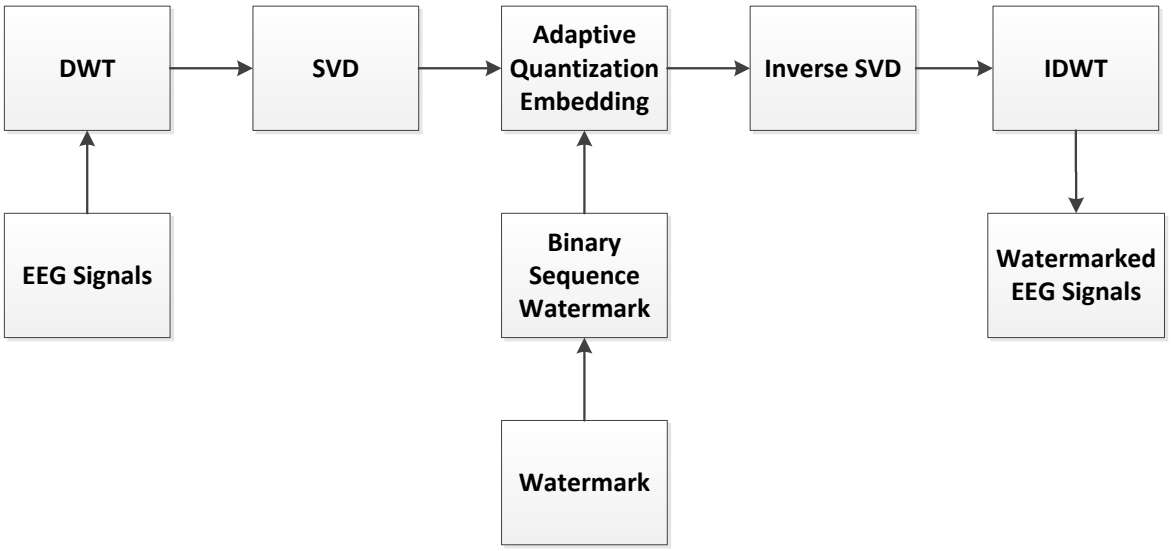


Figure 1: Diagram of the watermark embedding procedure.

6. Insertion of the watermark: Due to the stability of the matrix S_k under different attacks, the insertion of the watermark is performed by manipulating the coefficient in the highest singular value $S_i(1,1)$ of each matrix S_i by adaptive DM quantization methods. As the first singular values have the highest energy values, thus they are used to embed the watermark in order to guarantee the robustness and transparency. In addition,the popular DM quantization method has good robustness and blind nature, thus it is used in the embedding process. This embedding strategy can be formulated by following quantization function:

$$\begin{cases} \bar{S}_i(1,1) = round\left[\dfrac{S_i(1,1)}{\Delta}\right] + \dfrac{3}{4}\Delta & \text{if } w_k = 1 \\ \bar{S}_i(1,1) = round\left[\dfrac{S_i(1,1)}{\Delta}\right] + \dfrac{1}{4}\Delta & \text{if } w_k = 0 \end{cases} \tag{6}$$

where Δ is quantization step, $round[.]$ is rounding to the nearest integer value. It is obvious that the quantization step is significant in terms of both robustness and inaudibility. The larger the quantization step is, the more robust, but less transparent, the watermarking scheme is. Therefore, the quantization step should be specially developed to achieve optimal performance.

7. Each modified singular value is reinserted into matrix S_i and inverse SVD transformation is conducted to obtain the watermarked block $\bar{M}_k$ which is given by

$$\bar{M}_k = U_k \times \bar{S}_k \times V_k^T \tag{7}$$

8. The modified blocks are rearranged to one dimension vector and concatenated in order to obtain the modified approximation vector $\bar{A}_2$

9. The watermarked signal $\bar{X}$ is obtained by calculating the two-level inverse DWT using the modified approximation vectors $\bar{A}_2$ and the original detail vectors $D1$, $D2$.

3.2 Watermarking extraction

The detection is rather simple when only the watermarked signal and the watermark embedded positions used in watermark embedding are needed to extract the watermark. Fig. 2 which describe the watermarking extraction including:

1. Performed two-level DWT to the watermarked and possibly distorted host EEG signal $\bar{X}$ using a Haar wavelet filter to obtain three sets of coefficients $\bar{D}1$, $\bar{D}2$ and $\bar{A}2$.

2. DWT coefficient, $\bar{A}2$ is divided into different non-overlapping blocks $\bar{M}_k$ with the same block length as that in the watermark embedding process.

3. SVD transformation is applied on each block to produce singular values

$$\bar{M}_k = \bar{U}_k \times \bar{S}_k \times {\bar{V}_k}^T \tag{8}$$

4. The largest singular value of each diagonal matrix $\bar{S}_i$ located at the same position in the pre-embedding process is calculated.

5. Let $\phi = (\bar{S}_k - floor[\dfrac{\bar{S}_k}{\Delta}]) \times \Delta$, where $floor[.]$ is the rounding function which rounds the elements to the nearest integers. The embedded meaningful watermark sequence is extracted in the following rule:

$$\begin{cases} \bar{w_k} = 1 & \text{if } \phi \geq \dfrac{\Delta}{2} \\ \bar{w_k} = 0 & \text{if } \phi < \dfrac{\Delta}{2} \end{cases} \tag{9}$$

6. The binary watermark image $\bar{W}$ is obtained by organized as a 2-D matrix from the watermark sequence.

The EEG watermarking extraction phrase does not require original cover signal at the receiver, thus the proposed approach constitutes a blind watermarking scheme.

3.3 Performance optimization using PSO

In EEG watermarking, the quantization step is signal-dependent where different EEG signals require different

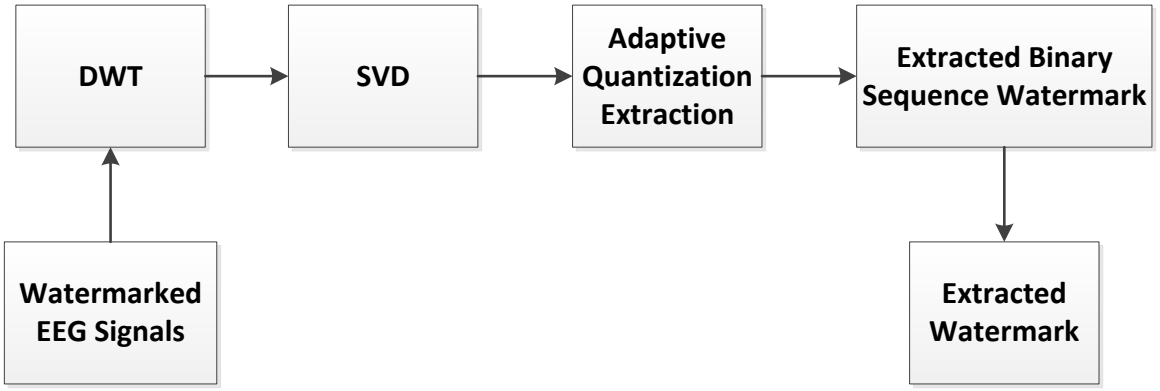


Figure 2: Diagram of the watermark extraction procedure.

quantization steps, rather than a fixed one. As seen in insertion process, the watermark is carried into the audio signal by quantization of the selected coefficient of each matrix S_k. Large quantization step (high Δ) of coefficients results in better robustness and more distortion of EEG signal, while a small quantization step(low Δ) leads to low robustness and low distortion. Hence, the parameter Δ must be carefully selected, for each EEG signal, in order to ensure best performances in terms of imperceptibility and robustness. However, the empirical selection of Δ is not an optimal solution. The selection process must be automatized in order to improve the performance of the proposed watermarking method.

In our method, $PSNR(X, X')$ is used to represent the imperceptibility, which denotes the Peak Signal To Noise Ratio between the original signal X and watermarked signal S'. $PSNR(X, X')$ measure defined as follows

$$PSNR = \frac{20log_{10}max(x)}{\sqrt{\frac{1}{N}\sum_{n=1}^{N}(x - x')^2}} \tag{10}$$

Normalized Correlation (NC) between original watermark (W) and extracted watermark after attack W' is a metric to determine the robustness, calculate as follows

$$NC(W, W') = \frac{\sum_{n=1}^{N}\sum_{m=1}^{M} w(n,m)w'(n,m)}{\sqrt{\sum_{n=1}^{N}\sum_{m=1}^{M} w(n,m)^2}\sqrt{\sum_{n=1}^{N}\sum_{m=1}^{M} w'(n,m)^2}} \tag{11}$$

We investigated the variation of $NC(W, W)$ with respect to quantization steps (Δ). Fig. 4 plots the effect of quantization steps on $NC(W, W)$ value of Subject 1, Fp1 channel in DEAP database (Koelstra et al. 2012) with different attacks. It is clear from this figure that $NC(W, W)$ values vary only between the range of $\Delta = 1 - 20$. Beyond $\Delta = 20$, the $NC(W, W)$ values get stabilized. This is specifically true for all attacks including noise addition, re-sampling and low-pass filtering. For these attacks, the $NC(W, W)$ values with respect to Δ are throughout constant. Thus, it is concluded that for the attacks used by us in this simulation, the range of $\Delta = 1 - 20$ is an appropriate range to determine suitable quantization step. We therefore use this range for Δ for our future computations for Subject 1, Fp1 channel.

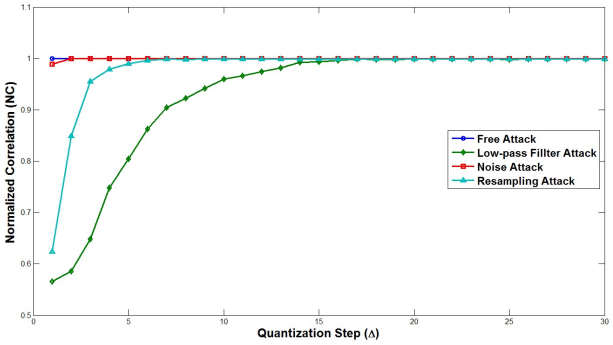


Figure 3: Effect of quantization steps on NC values with different attacks

To examine the effect of quantization on visual quality of watermarked EEG signals, we plot its corresponding computed values as a function of Δ. Fig.5 depicts the effect of quantization steps on $PSNR$ for 5 EEG channels. It is clear that $PSNR$ and Δ are inversely proportional to each other for all channels.

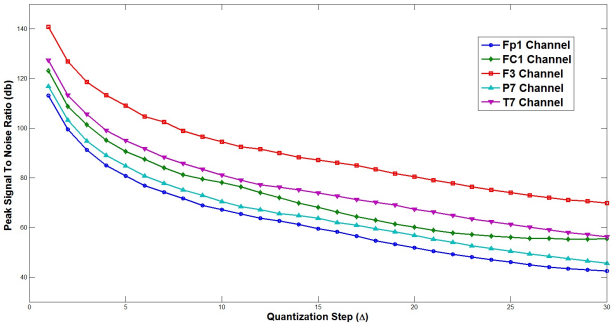


Figure 4: Effect of quantization steps on PSNR in different EEG channels

It is clear from this discussion that any objective function used to optimize watermark embedding should take both $PSNR$ and $NC(W, W)$ into account. To solve this problem, the PSO optimization is used to find the adequate quantization step, for each signal, which guarantees the best imperceptibility-robustness compromise. The optimization process for finding the suitable quantization step, Δ in our watermarking scheme is shown in Fig. 3.

The values of quantization step are obtained by implementing a meta-heuristic algorithm, in this paper is PSO.

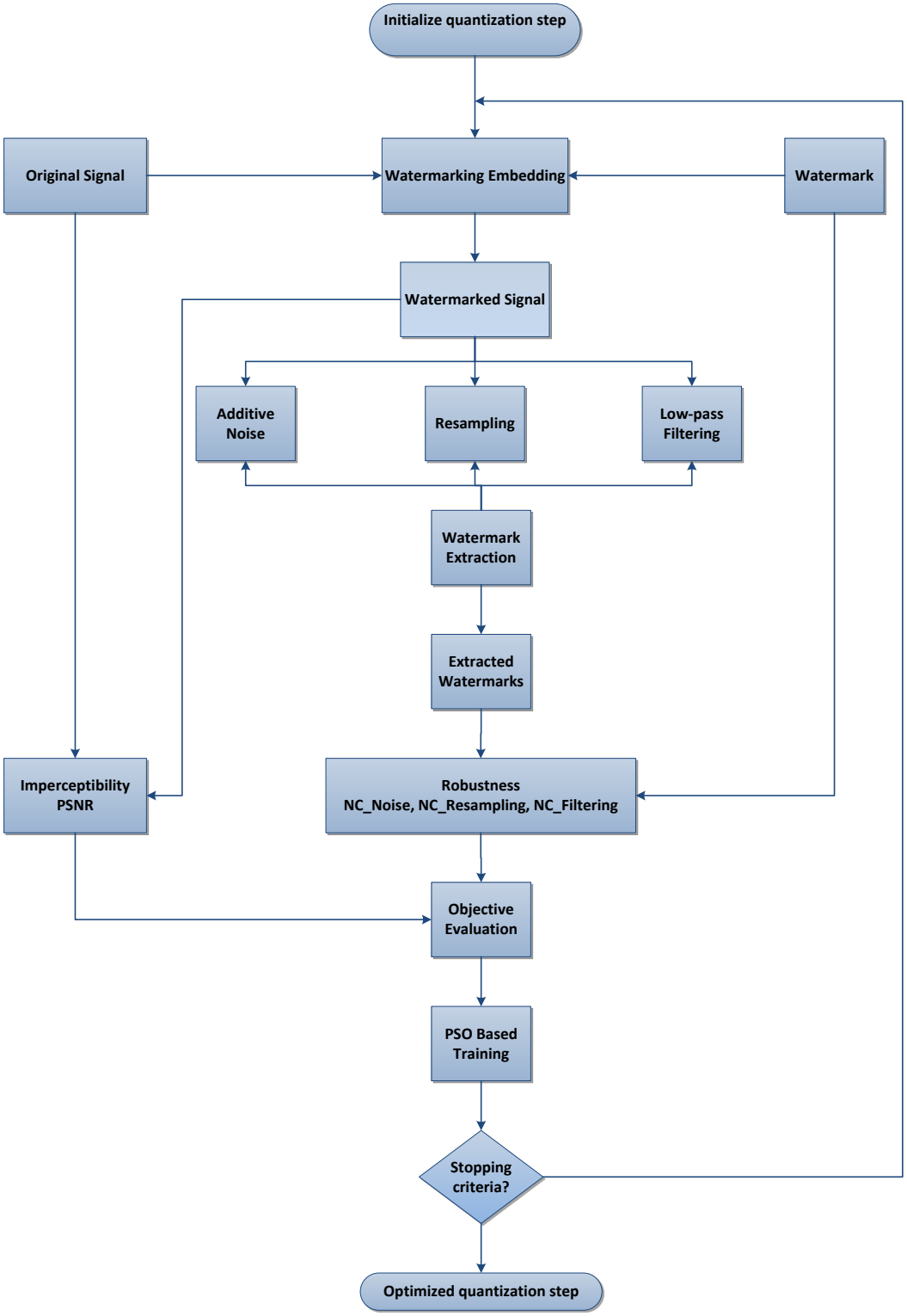


Figure 5: PSO optimisation process

For each iteration in PSO, the value of Δ is examined for several attacks, such as noise attack, re-sampling attack, and cropping attack. Due to the flexibility of the developed system, the other attacking schemes can easily be added to the system or replaced with those used in the PSO optimization process. At the end of PSO iteration, we will obtain the near optimum quantization step.

To find an optimal solution in the objective function, PSO is first initialized with a group of random particles, each of which represents a candidate solution to the problem. Then both mutation and crossover rates are utilized for the update of the solutions. Mutation rate has a control of step size of population mutating, whereas crossover rate decides search variable numbers, respectively.

The objective value function should be designed as a function of both imperceptibility and robustness to obtain the optimal performance by PSO, the objective function

Table 1: Performance metrics on average for different EEG signal channels of 32 subjects

EEG Channel	Quantization Step (Δ)	PSNR (db)	NC	BER
FP1	19.867	60.558	1	0
FC1	19.476	64.829	1	0
F3	19.573	59.118	1	0
P7	20.925	66.395	1	0
T7	19.978	70.275	1	0
Average	**19.964**	**64.235**	**1**	**0**

can be designed as

$$Objective = f(imperceptibility, robustness) \quad (12)$$

$PSNR$ and NC are employed to represent the imperceptibility and robustness of our scheme. Therefore, the objective function is designed as:

$$Objective = max(PSNR + \gamma \times \frac{1}{R}\sum_{i=1}^{k} NCi) \quad (13)$$

where R is the number of attacks. Weighting factors γ is introduced as significant difference might take place between the metrics of the watermarked EEG host and the extracted watermark. As the PSNR is much larger as compared to the associated NC values therefore; a weighting factor γ is used to balanced out the influences cause by the two parameters. Generally speaking, the PSNR value should be greater than 40dB, while the NC value lies between 0 and 1. Therefore, the inclusion of weighting factor is necessary.

4 Experimental Results and Discussion

In our experiments, the DEAP dataset (Dataset for Emotion Analysis using Electroencephalogram, Physiological and Video Signals) which is an open database proposed by Koelstra et al. (Koelstra et al. 2012) was used as the original EEG signal source. The 5 random channels (FP1, FC1, F3, P7 and T7) of 32 subjects were chosen to test. A binary logo image with size 32x32 was used as the watermark image shown in Fig.7 (a). Based on empirical experience and the trial and error, the PSO optimization parameters chosen to achieve the optimal robustness and transparency are as follows $c_0 = 0.4$, $c_1 = c_2 = 1.8$, the number of particles being 30, the number of generation being 50, and the weighting factor γ being 50.

4.1 Imperceptibility

In our scheme, $PSNR$ was employed to evaluate the differences between original EEG signals and watermarked EEG signals. It should be noted that the larger $PSNR$, the better imperceptibility. A larger $PSNR$ value indicates that the watermarked EEG signal more closely resembles its original signal, meaning that watermarked EEG signal has better imperceptibility. According to Chen et al. (Chen et al. 1998), $PSNR$ above 40 dB indicates a good perceptual fidelity. The $PSNR$ (in dB) of the watermarked EEG are shown in Table 1, all of them are higher than 40 dB, thus this indicates that diagnosability is not lost and degradation to the overall signal is acceptable. It also shows that our watermarked EEG signal is near identical to the original EEG signal (Fig.6).

4.2 Robustness

In order to evaluate the robustness of the proposed method against the common signal processing attacks, we used *Bit Error Rate* (BER) and NC measures. The NC measures are defined in Eq. (11), BER between original watermark (W) and extracted watermark after attack W' are calculated as following:

$$BER(W, W') = \frac{\sum_{n=1}^{N}\sum_{m=1}^{M} w(n,m) \oplus w'(n,m)}{N \times M} \quad (14)$$

where the symbol $\oplus$ is exclusive-or (XOR) operator.

The following signal attacks were performed in Matlab:

1. Noise addition: Additive white Gaussian noise (AWGN) was added to the watermarked EEG signal with 20dB.
2. Low-pass filtering: The low-pass filter with cut-off frequency of 40Hz was applied to all watermarked EEG signals.
3. Re-sampling: The original EEG signals were sampled with a sampling rate of 128 Hz. Watermarked EEG signals were resampled at 64 Hz and then restored by sampling again at 128 Hz.

In attack-free case, we extracted watermark from watermarked EEG signals using the proposed watermark extraction algorithm. Table 1 shows $BER = 0$, $NC = 1$ in case of no attack, meaning that watermark can be accurately extracted from the watermarked EEG signal. In addition, the proposed watermarking scheme uses the trained SVDD model to extract the watermark without the original EEG signal, thus our proposed watermarking scheme is blind. As seen from Table 2, after applying MATLAB attacks on watermarked EEG signals, it is observed that the values of BER is very low (less than 3%) while the values of NC is very high (close to one), which implies extracted watermark is very similar to the original watermark. Therefore, this indicates that the robustness of the proposed scheme is very good. In addition, $BER < 3\%$ could be corrected with the use of error correcting codes (Wicker 1995).

4.3 Error Analysis

The performance of a watermarking system is generally characterized by two types of errors (Fan & Wang 2009), the false-positive error and false-negative error. The the false-positive error is the probability that an unwatermarked EEG signal declared as watermarked by the decoder, while false-negative error is the probability that a watermarked EEG signal declared as unwatermarked by the decoder. The probability of false-positive error P_{FP} and probability of false-negative error P_{NE} can be computed as:

$$P_{FP} = 2^{-m}\sum_{h=\lceil \rho m \rceil}^{m}\binom{m}{h} \quad (15)$$

$$P_{FN} = \sum_{h=0}^{\lceil \rho m \rceil - 1}\left[\binom{h}{m}P^h(1-P)^{m-h}\right] \quad (16)$$

where $\binom{m}{h}$ is the binomial coefficient, m is the total number of watermark bits, h is the total number of matching bits, and P is probability of difference between extracted watermark and original watermark ($w \neq w'$). According to (Bhat et al. 2010), the desired false alarm error must be smaller than 10^{-6} order of magnitude. We have $h = \lceil (1 - BER) \times m \rceil$, therefore BER less than

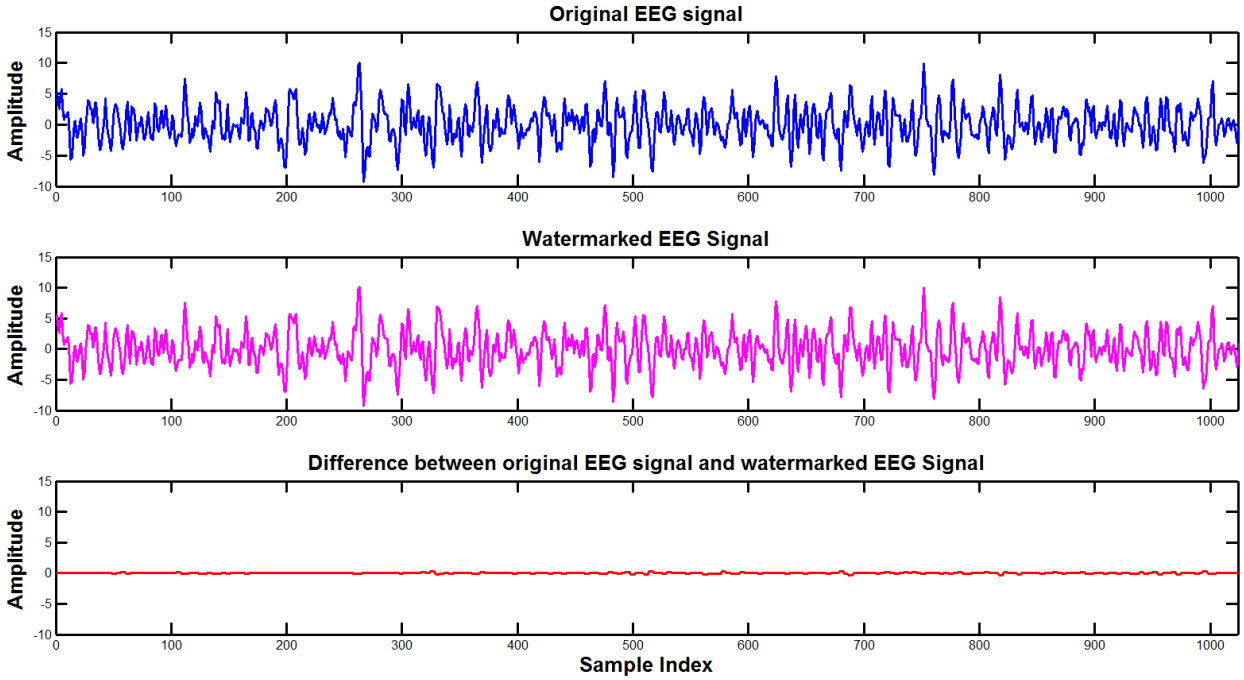


Figure 6: Original EEG signal vs Watermarked EEG signal (above), Difference between original EEG signal and watermarked EEG signal (below) in channel Fp1, subject 01

Table 2: Performance metrics for different EEG signal channels under different attacks

EEG Channel	Noise Addition		Low-pass Filtering		Resampling	
	BER (%)	NC	BER(%)	NC	BER(%)	NC
Fp1	0.32	0.9971	2.93	0.9762	3.71	0.9703
FC1	0.67	0.9947	1.41	0.9892	3.52	0.9715
F3	1.21	0.9861	3.51	0.9718	2.48	0.9802
P7	0.49	0.9961	3.42	0.9726	0.65	0.9947
T7	0.88	0.9929	2.72	0.9812	1.39	0.9869
Average	**0.71**	**0.9933**	**2.80**	**0.9782**	**2.35**	**0.9807**

(a) (b) (c) (d) (e)

Figure 7: Result of watermark extraction at channel Fp1 of subject 01. (a) Original watermark. (b) Free attack. (c) Noise Addition. (d) Low-pass filtering. (e) Re-sampling.

20% meets this demand. If we set $BER = 20\%$, then $\rho = 0.8$. Figure 3 shows the P_{FP} for watermark length in range $(0, 100]$, which indicates that the P_{FP} approaches 0 when watermark length is larger than 20. In our method, $m = 1024$, Eq.(15) gives $P_{FP} = 2.6209 \times 10^{-88}$, hence the false positive is close to 0.

In Eq.(16), the approximate value of P can be obtained from the BER under different attacks. As can be seen from Table 1 and Table 2, the average of BER is less than 3%, so P can be taken as 0.97. Figure 4 shows the P_{NP} for watermark lengh in range $(0, 100]$.

By substituting the values of m, ρ, and P, Eq. (16) gives $P_{FN} = 1.5286 \times 10^{-102}$. In summary, our experimental results show that the proposed blind watermarking scheme based on pattern recognition for biomedical data has good imperceptibility and strong robustness against several different attacks.

5 Conclusion

A ownership protection based on optimized watermarking scheme for EEG data in data mining has been developed. The watermarking scheme based on DWT with the exploration of SVD properties and DM quantization. The attractive properties of SVD, DWT and DM quantization techniques make our scheme very robust to various common signal processing attacks. In our optimization process, PSO was used to make an optimal trade-off between imperceptibility and robustness through effective selection of quantization steps to locate the best parameters to insert the watermarks. The quantization steps were optimally adapted to achieve the most suitable performance for various EEG data with different characteristics for medical application. In addition, our watermark scheme possesses the characteristic of blind extraction which does not require the original EEG signal in extraction, thus reducing a lot of space for storing the original EEG data and watermark.The experimental results have revealed that the pro-

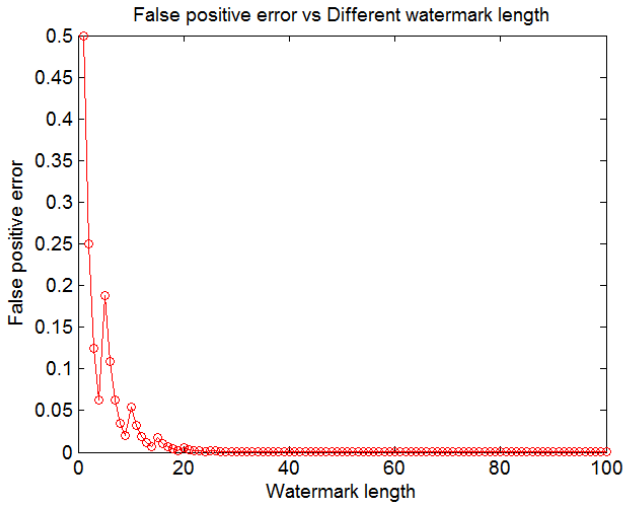


Figure 8: False Positive Error rate under different watermark length

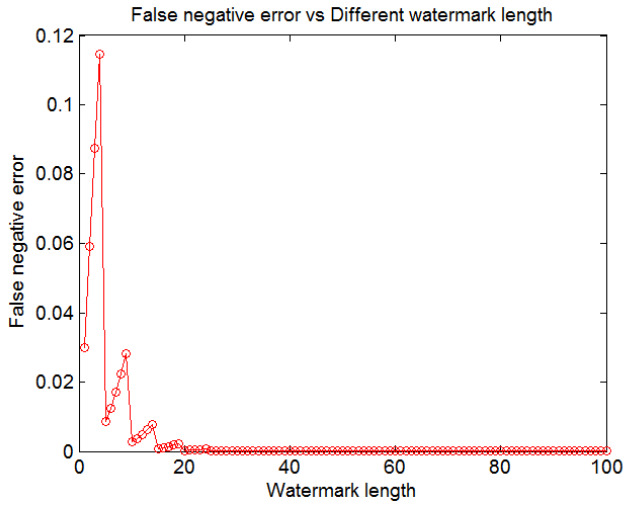


Figure 9: False Negative Error rate under different watermark length

posed watermarking scheme achieves good imperceptibility and strong robustness against common signal processing.

In the future work we will consider the following problems:

1. Enhancing privacy of EEG data in data mining by tracing the source of an authorized release of EEG data in networks.
2. Implementing the error coding code in watermark extraction.
3. Investigation of other evolutionary algorithms for the performance with respect to the existing algorithms.

References

Alhaqbani, B. & Fidge, C. (2008), Privacy-preserving electronic health record linkage using pseudonym identifiers, *in* 'e-health Networking, Applications and Services, 2008. HealthCom 2008. 10th International Conference on', IEEE, pp. 108–117.

Amin, H. U., Malik, A. S., Ahmad, R. F., Badruddin, N., Kamel, N., Hussain, M. & Chooi, W.-T. (2015), 'Feature extraction and classification for eeg signals using wavelet transform and machine learning techniques', *Australasian Physical & Engineering Sciences in Medicine* **38**(1), 139–149.

Bender, W., Gruhl, D., Morimoto, N. & Lu, A. (1996), 'Techniques for data hiding', *IBM systems journal* **35**(3.4), 313–336.

Bertino, E., Ooi, B. C., Yang, Y. & Deng, R. H. (2005), Privacy and ownership preserving of outsourced medical data, *in* '21st International Conference on Data Engineering (ICDE'05)', IEEE, pp. 521–532.

Bhat, V., Sengupta, I. & Das, A. (2010), 'An adaptive audio watermarking based on the singular value decomposition in the wavelet domain', *Digital Signal Processing* **20**(6), 1547–1558.

Bhatnagar, G. & Wu, Q. J. (2013), 'Biometrics inspired watermarking based on a fractional dual tree complex wavelet transform', *Future Generation Computer Systems* **29**(1), 182–195.

Chen, T.-S., Chang, C.-C. & Hwang, M.-S. (1998), 'A virtual image cryptosystem based upon vector quantization', *Image Processing, IEEE Transactions on* **7**(10), 1485–1488.

Fan, M. & Wang, H. (2009), 'Chaos-based discrete fractional sine transform domain audio watermarking scheme', *Computers & Electrical Engineering* **35**(3), 506–516.

Gupta, A. K. & Raval, M. S. (2012), 'A robust and secure watermarking scheme based on singular values replacement', *Sadhana* **37**(4), 425–440.

Han, J. (2001), 'Kamber., m.: Data mining: Concepts and techniques'.

Hassan, R., Cohanim, B., De Weck, O. & Venter, G. (2005), A comparison of particle swarm optimization and the genetic algorithm, *in* 'Proceedings of the 1st AIAA multidisciplinary design optimization specialist conference', pp. 18–21.

Jahankhani, P., Kodogiannis, V. & Revett, K. (2006), Eeg signal classification using wavelet feature extraction and neural networks, *in* 'Modern Computing, 2006. JVA'06. IEEE John Vincent Atanasoff 2006 International Symposium on', IEEE, pp. 120–124.

Kamran, M. & Farooq, M. (2012), 'An information-preserving watermarking scheme for right protection of emr systems', *IEEE Transactions on Knowledge and Data Engineering* **24**(11), 1950–1962.

Kamran, M. & Farooq, M. (2013), 'A formal usability constraints model for watermarking of outsourced datasets', *IEEE transactions on information forensics and security* **8**(6), 1061–1072.

Kennedy, J. (2011), Particle swarm optimization, *in* 'Encyclopedia of machine learning', Springer, pp. 760–766.

Kincade, K. (1998), 'Data mining: digging for healthcare gold', *Insurance & Technology* **23**(2), 2–7.

Koelstra, S., Mühl, C., Soleymani, M., Lee, J.-S., Yazdani, A., Ebrahimi, T., Pun, T., Nijholt, A. & Patras, I. (2012), 'Deap: A database for emotion analysis; using physiological signals', *Affective Computing, IEEE Transactions on* **3**(1), 18–31.

Koh, H. C., Tan, G. et al. (2011), 'Data mining applications in healthcare', *Journal of healthcare information management* **19**(2), 65.

Kumsawat, P. (2010), 'A genetic algorithm optimization technique for multiwavelet-based digital audio watermarking', *EURASIP Journal on Advances in Signal Processing* **2010**(1), 1.

Orhan, U., Hekim, M. & Ozer, M. (2011), 'Eeg signals classification using the k-means clustering and a multilayer perceptron neural network model', *Expert Systems with Applications* **38**(10), 13475–13481.

Percival, D. B. & Walden, A. T. (2006), *Wavelet methods for time series analysis*, Vol. 4, Cambridge university press.

Sriyingyong, N. & Attakitmongcol, K. (2006), Wavelet-based audio watermarking using adaptive tabu search, *in* '2006 1st International Symposium on Wireless Pervasive Computing', IEEE, pp. 1–5.

Trelea, I. C. (2003), 'The particle swarm optimization algorithm: convergence analysis and parameter selection', *Information processing letters* **85**(6), 317–325.

Wicker, S. B. (1995), *Error control systems for digital communication and storage*, Vol. 1, Prentice hall Englewood Cliffs.

Yoo, I., Alafaireet, P., Marinov, M., Pena-Hernandez, K., Gopidi, R., Chang, J.-F. & Hua, L. (2012), 'Data mining in healthcare and biomedicine: a survey of the literature', *Journal of medical systems* **36**(4), 2431–2448.

Segregator ant colony optimization with application to text clustering

Alireza Moayedikia[a, b], Kok-Leong Ong[b], Yee Ling Boo[c]
[a] Department of Information Systems and Business Analytics, Deakin University, Victoria 3125, Australia.
[b] SAS Analytics Innovation Lab, ASSC, La Trobe University, Victoria 3086, Australia.
[c] School of Business IT and Logistics, RMIT University, Victoria 3000, Australia.

amoayedi@deakin.edu.au, kok-leong.ong@latrobe.edu.au, yeeling.boo@rmit.edu.au

Abstract

We present a new variant of ACO that primarily focuses on the paths on the solution space that lead to high rate of *segregation* between dissimilar data, while declining the inner cluster dissimilarity. This feature enables ACO to intelligently balance the *exploration* and *exploitation* steps in the optimization process. This objective is fulfilled through considering a specific quality function that increases the pheromone laid on the edges leading to tours with high document separation. The modified algorithm that is referred to as Segregator Ant Colony (SACO) is then applied to solve the document clustering problem. To further improve the efficiency of the proposed clustering algorithm, we hybridized it with the *k*–means clustering algorithm to take advantage of fine-tuning power of the *k*-means in exploring the solution in the proximity of global optima obtained by SACO based clustering method. To demonstrate the effectiveness of the proposed algorithms, we conduct experiments on standard datasets and compare the proposed algorithms with other baseline and state-of-the-art ant colony optimization algorithms. Experimental results indicate that the proposed algorithms outperform the competitors.

Keywords: Ant Colony Optimization, Hybrid method, *k*-means, Text Clustering

1 Introduction

Text mining is a broad area of research that has received much attention in recent decades. Much of the work in this area is concerned with text clustering and text categorization tasks. Text clustering is the task of partitioning a set of documents to a given number of classes (unlabeled categories) that fulfills some objective on the partitions. In text classification, documents are categorized along a series of pre-specified and labeled groups or categories. In text clustering, documents are divided into categories that are not labeled by a human expert, where such unlabeled categories are called clusters. Therefore categorization and clustering are called supervised and unsupervised methods of partitioning, respectively. In recent years many evolutionary and swarm based algorithms such as ant colony (Dziwiński, et al., 2012) (Azzag, et al., 2007) (Niknam & Amiri, 2010), harmony search, and particle swarm have been utilized to solve the clustering problem.

The dominating methodology in applying evolutionary methods to clustering problem is to cast the clustering problem as an optimization problem and then explore the solution space to find the best solution by adapting the searching process of evolutionary method. Although most of these algorithms were successful, but they are good at locating near optimal solutions in the solution space and rarely focus on highlighting paths leading to high separation of dissimilar data. In other words, in spite of their relative effectiveness, these methods mostly suffer from entrapping in a near optimal solution. This problem is known as *local entrapment*. As a result, it is an interesting challenge to adopt the searching process in a way that solves the optimization problem and simultaneously yields the maximum separation of dissimilar data.

In this paper we consider the Ant Colony Optimization (ACO) method and adopt it for clustering problem. The ACO as a swarm intelligence application has shown its significance and applicability in various areas of science (Mora, et al., 2013) (Kim, et al., 2008) (Min-Thai, et al., 2012), To tackle the deficiencies discussed above when applying ACO to clustering problem, we propose a new ant colony algorithm suitable for clustering of data that is rooted in its quality function to highlight paths in the solution space leading to clusters with high rate of intra similarity. Then, the algorithm is applied to text clustering by modeling the clustering problem as an optimization problem. The proposed algorithm is a swarm application suffering from local entrapment. We believe that its integration with a local procedure like *k*-means can help the algorithm to better fine-tune the final solution at the proximity of optima and can guide the algorithm to boost its performance. As a result, the new text clustering algorithm is a hybridization of the *k*-means and the proposed ant colony optimization method.

The ant colony algorithm proposed herein is named Segregator Ant Colony Optimization (SACO), as it searches the solution space and magnifies paths that lead to low similarity among clusters whilst contributing to high similarity within each cluster. Based on SACO, the first version of the text clustering algorithm that is referred to as SACOClust is introduced. Here, ants move across the solution space trying to find the optimal parts of the space that minimizes the quality function which its optimal value is near zero. In fact a suitable quality function is the one having characteristic of Equation 1. This equation indicates that a quality function suitable for SACO is the

one that its objective approaches zero as the algorithm proceeds:

$$\lim_{x\to\infty} f(x) = 0 \qquad (1)$$

The problem associated with SACOClust is its weakness in fine-tuning the obtained solution in the proximity of final near-optimal solution. Although, finding the proximity of near-optimal solution and escaping from a local optimal solution is an appealing feature of SACOClust, but lack of fine-tuning of final solution may deteriorate the effectiveness of the algorithm. To resolve this issue, the SACOClust is integrated with the *k*-means algorithm to form the hybrid method named *k*SACOClust. The hybrid method is able to locate the best solution in the proximity of the global solution and fine-tune it enough to attain the desired clustering performance, i.e., combining the explorative power of SACOClust with fine-tuning power of *k*-means. In summary, the present work makes the following contributions:

A new variation of ant colony optimization algorithm that focuses on the paths in the solution space that lead to high rates of separation between dissimilar data, while maximizing the inner cluster similarity. These modifications are aimed at increasing the explorative power of the ACO algorithm and propagation of knowledge in optimization process, respectively. This algorithm is called Segregator Ant Colony (SACO).

A novel document clustering algorithm based on SACO is proposed that is referred to as Segregator ant clustering (SACOClust). This algorithm aims at highlighting the paths in the solution space that leads to high separation of the dissimilar documents while clustering the most similar ones. This objective is fulfilled through considering a specific quality function that increases the pheromone laid on the edges leading to tours with high document separation.

A hybrid clustering algorithm using *k*-means and the SACOClust algorithm is proposed. Although the problem of getting stuck in the local optimum has been solved in the SACOClust method, the algorithm still suffers from locating the best solution in the proximity of the found global solution. The hybrid technique alleviate this problem by combining the fine tuning capability of the *k*-means in the proximity of global solution and the searching power of the SACOClust in locating the global solution.

A comprehensive set of experiments to demonstrate the effectiveness of SACOClust and hybrid algorithm. We have applied these algorithms on various standard datasets and got very promising results compared to the *k*-means, traditional ant colonies (Dorigo, et al., 1996) (Gambardella & Dorigo, 1996) (Stutzle & Hoos, 2000), some recent state-of-the-art algorithms (Zhang & Feng, 2012) ant colony based text clustering algorithms (Dziwiński, et al., 2012) and harmony search (Forsati, et al., 2012) clustering algorithms.

The paper is organized as follows. We begin in Section 2 by briefly reviewing the related works in the areas of partitioning, heuristic and swarm based document clustering. In Section 3, the improved ant colony algorithm is described. In Section 4 the improved ant colony based text clustering algorithm will be discussed. Section 5 presents the data sets used in our experiments, empirical study of parameters on convergence on the behavior of proposed algorithms, and comparison of different algorithm. It also contains the performance evaluation of the proposed algorithms compared to well-known algorithms. Finally Section 6 concludes the paper.

2 Literature review

Many text clustering algorithms have been proposed including *k*-means, evolutionary algorithms and swarm based algorithms. These algorithms are well suited for clustering a large document sets due to their relatively low computational requirements, which is of interest to this work. Data clustering algorithms or specifically text clustering algorithms can be seen from two different viewpoints. In view mainly data (e.g. document) clustering algorithms fall into two categories of discriminative and generative algorithms. In discriminate algorithms the data are clustered based on their similarity and closeness. In fact an optimization measure is used to be maximized or minimized in order to refine the clustering and find the best clustering of data (text). In model-based algorithms a proportion of textual data is used to extract a model from the dataset. Then the future incoming data are clustered based on the produced model.

Besides that there is another view that, considers text clustering into two other classes of partition and hierarchical. Hierarchical algorithms try to partition the data into many levels and a tree-like structure while partitioning methods try to partition a collection of documents into a set of groups, so as to maximize a pre-defined fitness value, in which the clusters can be overlapped or not. Although hierarchical methods are often said to have better quality clustering results, usually they do not provide the reallocation of documents, which may have been poorly classified in the early stages of the text analysis (Mahdavi & Abolhassani, 2009).

The best known partitioning algorithm is *k*-means. In its simplest form, *k*-means selects *K* documents as cluster centers and assigns each document to the nearest cluster. Next, the algorithm repeatedly evaluates allocations and reassigns documents until a convergence criterion is met. Under certain configurations, this algorithm can have almost linear time complexity (Xu & Wunsch, 2005). This algorithm has some drawbacks. Specifically, its performance is sensitive to the initial selection of appropriate centroids, the number of clusters (*K*), convergence to local optima, and noisy and outlying data (Xu & Wunsch, 2005).

2.1 Partitioning based clustering algorithms

(Huang, et al., 2005) have proposed, a new *k*-means type algorithm, called W-*k*-means that can automatically weigh variables based on the importance of the variables in clustering. W-*k*-means associates a weight with each variable and adds an additional step to the basic *k*-means algorithm to update the variable weights based on the current partition of data. Specifically, based on the current partition in the iterative *k*-means clustering process, the algorithm calculates a new weight for each variable based on the variance of the within cluster distances. The new

weights are used in deciding the cluster memberships of objects in the next iteration.

(Arthur & Vassilvitskii, 2007) have introduced *k*-means++ as a stochasticity algorithm. It starts with the uniform random selection of one object as the first centroid. Then, each next centroid is determined using a weighted probability distribution. Specifically, the probability for a candidate object to be selected as the next new centroid is proportional to the squared distance between the object and its nearest centroid previously selected.

Clustering refinement refers to post-processing procedures that aim to improve the clusters produced by a clustering method. The refinement algorithm may be a specialized algorithm that proceeds to make small changes in the clusters, such as single object reassignment or swapping cluster memberships of pairs of objects (Kanungo, et al., 2004).

(Kanungo, et al., 2004) claimed that *k*-means is not always an effective document clustering method. They argue that this deficiency stems from sparsity, low quality of text data and high dimensionality and propose an approach that first selects a series of objects as cluster members and then try to compute the representative of the clusters. To address the problem of *k*-means many optimization algorithms such as swarm and evolutionary algorithms have been used and adapted the algorithm for web content mining applications.

2.2 Swarm-based clustering algorithms

Swarm-based algorithms are generally referred to optimization methods that rely on a swarm to conduct the search process. These algorithms have proven to be very effective in solving different problems of computer science. These algorithms are bee colony optimization (BCO) (Forsati, et al., 2012), particle swarm optimization (PSO) (Kennedy & Eberhart, 1995), and ant colony algorithms (Forsati, et al., 2014). The most important improvements in the area of optimization, seek to exploit optimization approaches with a view exploring the search space more deeply, in simpler term it means that searching most of the possible solutions for the problem in the solution space. Ant clustering and PSO (Kennedy & Eberhart, 1995) have also been used to improve the clustering results (Labroche, et al., 2003).

(Yang, et al., 2005) proposed a new ant colony based method for text clustering using a validity index is introduced. In this method the walking of the ants is mapped to the picking or dropping of projected document vectors with different probabilities. In (Yang, et al., 2005) a clustering validity index is used for algorithm evaluation, to find the optimal number of clusters and to reduce outliers. Furthermore, the experiments reveal that the proposed model has superiority, in terms of the proposed clustering validity index over LF algorithm and ART neural networks.

In another work (Zhang, et al., 2008) suggests that the random movements of ants in the solution space leads to slow convergence. They provide a method for faster text clustering, called AFTC. The approach employs the pheromone laid by the ants to avoid randomness of movement, which lead the ants to move towards a direction with high pheromone concentration at each step. The direction of movement is the orientation where the text vectors are relatively more concentrated. The ant-based fast text clustering approach does not need the number of clusters to be initialized before execution. The author also considers the amount of time needed to perform experiments and shows that the average time of ant-based fast text clustering is faster than that of traditional ant-based text clustering.

(Hui, et al., 2009) presented a new text clustering approach named elite Ant Colony Optimization Clustering (EACOC), based on suitable retention of the elites. The mechanism is to retain the elites that the algorithm works, in a way that in each iteration it retains a certain number of valuable solutions into the next cycle, with the purpose of improving algorithm performance. For assessment the author used a text data set and two external evaluation measures to show that the algorithm improved the performance in compare to ant colony optimization clustering (ACOC).

(Dziwiński, et al., 2012) presented a new Fully Controllable Ant Colony Algorithm (FCACA) for documents clustering, which is an enhanced version of the Lumer and Faieta Ant Colony Algorithm (LF-ACA). This introduces a new version of the basic heuristic decision function that significantly improves the convergence and provides greater control over the process of the grouping data.

(Azzag, et al., 2007) presented a new model a new model for data clustering, which is inspired from the self-assembly behavior of real ants. Real ants can build complex structures by connecting themselves to each other's. It is shown in the paper that this behavior can be used to build a hierarchical tree-structured partitioning of the data according to the similarities between those data.

Also PSO algorithm has shown a significant contribution in data clustering. (Das, et al., 2008) present a novel, modified PSO-based strategy for hard clustering of complex data. An important feature of the proposed technique is that it is able to find the optimal number of clusters automatically (that is, the number of clusters does not have to be known in advance) for complex and linearly non-separable datasets.

Another PSO-based algorithm was proposed by (He-Nian, et al., 2009), as an effective and unsupervised text clustering method (OK-PSO). In this approach, *k*-means is used to calculate the distance from each term to the cluster centers, and then Otsu is used in its two-dimensional form to evaluate the optimization of the clustering distance threshold. The process of 2D Otsu is taken by PSO algorithm.

(Kang & Zhang, 2012) proposed a fuzzy c-means (FCM), as a new method for document clustering. This is employed as part of a hybrid approach (PSO-FCM) through combining fuzzy c-means and PSO. The purpose of this combination is to make the most of the advantages of the both methods, through helping the FCM to escape from local optima by using PSO. Besides that its slow convergence speed is another problem can be solved through hybridization with PSO.

Meta-information, such as usage of external resources and materials, and user tagging can be used to inform text clustering procedures. This approach is utilized in (Li, et al., 2008), which showed that data on the number of user tags is not sufficient to enhance the clustering. Therefore, the paper proposes a method to expand on user tagging data. More precisely, user tagging is used as background knowledge to add more useful tags to the original tag document. However, adding tags may give rise to the phenomenon of topic drift, which means that the dominant topic(s) of the original document are changed. A novel generative model, called Folk-LDA, is designed, to solve topic drift. Folk-LDA, jointly models original and expanded tags as independent observations. A summary of parameters used in the rest of this paper are outlined in Table 1.

Table 1 – Symbols used and their meaning in this paper

Symbol	Meaning
K	The number of clusters.
M	The number of ants.
γ	The number of iterations.
τ_{ij}	The pheromone value of the edge connecting two nodes of *i* and *j*.
s_p	Tour taken by the *p*th ant.
c_{il}	A path within the tour s_p.
θ ~ U (0, 1)	A balancing coefficient.
N_{ij}	The number of documents belong to category *i* placed in cluster *j*.
N_j	The number of documents in cluster *j*.
P_{ij}	The probability that a member of cluster *j* belongs to class *i*
C_i	The *i*th cluster
D_{ij}	The Euclidean distance between the *j*th document in center *i* and the cluster center C_i

3 Segregator Ant Colony Optimization

In this section we introduce our novel ant colony optimization method that is referred to as Segregator Ant Colony Optimization (SACO). In ant colony optimization, the algorithm starts from a randomly chosen node in the solution space. Then the ants search across the solution space to search for an optimal solution for a desired number of iterations. The ants can be diversified to search different solutions in the space, through applying local pheromone updates.

For further and detailed explanations of the algorithm the interested readers can refer to (Dorigo, et al., 1996). The main idea behind the SACO algorithm is to utilize a local pheromone with the purpose of prevention from premature convergence and making all the ants to select the same path repeatedly. In the modified algorithm, the paths that minimize the fitness function will be selected. Roughly speaking, the SACO algorithm is strong in finding the paths that force the clusters include most similar data points, while forcing the dissimilarity between clusters to be as large as possible.

Unlike the traditional ACO method where a pheromone update is done at the end of each iteration by all the ants, in the SACO the pheromone update is adaptive and is based on the fitness function used to guide the search process. The novelty of this variation is that the pheromone update principle will lay higher values on the edges that are more likely to lead to creation of solutions that separate the data as accurate as possible, through only identifying their fitness function. While in other variations of ant colony algorithms the concentration is less on identifying potential paths that lead to more data separations, and the authors have to model the work through changing the ant colony variations based on their algorithm. The elements of the algorithm are explained in what follows.

3.1 Initialization

In this stage, number of ants' iterations and some other random parameters are initialized. Each ant is assigned a colony randomly, and the ants are ready to select the next colony based on a criterion.

3.2 Selection

To explain the selection probability, each ant at each node chooses the next possible movement according to the below selection probability function.

$$p_{ij}^{k} = \frac{\tau_{ij}}{\sum_{cil \in N(s_p)} \tau_{ij}} \qquad (2)$$

where $cil \in N(s_p)$ is the total number of possible and available paths that the ant can choose. p_{ij}^{k} is the probability that the ant K moves from node i, to node j. It is generally believed that pheromone value can be a good indication as the level of goodness of choosing the next possible movement, therefore all the pheromones are summed and the probability of choosing each path is generated by dividing the pheromone value of the edge by the sum value. The outcome along with a random number generator will help to select the next path.

3.3 Local Pheromone Update

The selection of the edges and creating paths is based on the pheromone laid by the previous ants. Therefore it can play a significant role in diversifying the solution created by the ants population. The local pheromone update is done as each ant passes a selected edge.

$$\tau_{ij} = (1 - \theta) \cdot \tau_{ij} + \theta \times \tau_0 \qquad (3)$$

where, τ_{ij} is the pheromone laid on the edge *(i, j)*, $\theta \in (0, 1]$, is a constant number known as pheromone decay coefficient, and τ_0 is the initial number of the pheromone on the edge *(i, j)*.

3.4 Assessment and pheromone update

Final updating of the pheromones is one of the key characteristics of the proposed ant colony in which according to some policies the pheromones laid on the edges are updated, to reflect the evolutionary nature of ant colony. Furthermore, one of the key characteristics of the proposed algorithm is that after each iteration is finished, all of the ants in the iteration are allowed to update the global pheromones, through the equations indicated as:

$$\tau_{ij} = \tau_{ij} + \sum_{j=1}^{m} \Delta\tau_{ij} \qquad (4)$$

$$\Delta\tau_i = \begin{cases} P(x)_{mi} & \text{if } (i, j) \in S_{mi} \\ 0 & Otherwise \end{cases} \qquad (5)$$

where S_{mi} is the tour taken by the *i*-th ant, $P(x)_{mi}$ is the quality function for the ant *i*-th, in which for $x_1 > x_2$, $P(x_1) < P(x_2)$ and F_k are the set of edges traversed and the fitness value of the *k*-th ant. In fact quality function can be $\frac{1}{G(x)}$ in which *G(x)* is the value of fitness function, with optimal value near to zero. Therefore assessment can affect the ants' ability in distinguishing worthy and non-worthy paths leading to high dissimilar data separation. In fact the purpose is to use any assessment method to provide separation among unrelated objects while groups the similar object. The name "segregator" is stemmed from this fact. Hence the assessment function is the one which satisfy Equation 1.

4 Segregator Ant Colony Optimization Clustering (SACOClust)

In this section the proposed ant colony algorithm SACO, is applied to text clustering and the algorithms are described in details. As a brief explanation, after the offline process of removing stop words and stemming (Porter, 1980), the documents are represented as vectors of words that should be clustered in *K* clusters. The ants randomly select a path that the path indicates a series of cluster centers for *K* number of clusters. Then the remaining documents are clustered according to their distance to the centers.

4.1 Initialization

During this stage the number of clusters and iterations and ants are identified by the users. Furthermore other parameter settings which require random initialization are randomly-initialized, such as the initial values of pheromones on the edges. Furthermore the ants are assigned randomly a different document, through generating an integer random number between zero and the number of documents in the collection, representing the centroid of a new cluster each.

4.2 Solution Representation

In the proposed algorithms, each ant generates a solution, and each solution includes a series of numbers indicating the documents selected as the cluster centers. The number of clusters, *K*, is identified by the user as explained above. Solutions are represented as in the example of Figure 1. Here, *K* is five that for instance A_1 and A_2 refer to clusters that have the 26th and 87th documents from the collection as their initial cluster centers.

A_1	A_2	A_3	A_4	A_5
26	87	2	90	103

Figure 1 - An example of initial cluster centroids

4.3 Selection

Solution creation stems from the chosen selection process. In this algorithm (Shown in Algorithm 1) the selection of the next document as the center for the next cluster is based on the pheromone values laid on the edge and a random number generator. The random number generator is used to guide the search process toward the next colony through comparing the randomly generated number with the output of the selection function explained in Equation 2. This random number is generated between the lowest and highest values of pheromones laid on the edges.

Algorithm 1: Random Selection Procedure

Input:
A set of ants have selected their first centroid
Output:
A set of ants with fully created solutions
Algorithm:

while the solution has not been created completely
Generate a random number $R \sim U(0, 1)$.
Select an edge with highest value of pheromone P_R where $P_R < R$
end while

4.4 Local Pheromone Update

Since the ants are moving across the search space based on the pheromones laid on the edges finding a good solution might lead to laying higher values of pheromone on the edges leading to the worthy solution. At this stage the ant colony is said to have converged. Furthermore it is likely that for this convergence to happen at the early stages of the algorithm and prevent the method from exploring the solution space more completely. In order to avoid premature convergence a local pheromone updating method is employed in which each ant after selecting a centroid updates the leading edge from the previous centroid to the current one. Pheromone values are updated according to Equation 2.

4.5 Assessment

When all the ants have completed their tours and created their solution each ant must be assessed. A tour is the path that each ant takes to generate a complete solution. In this paper to assess the quality of the clusters three alternative measures have been used: ADDC, F-measure and Entropy. The explanations regarding these evaluation functions are given in experimental section.

But according to explanation given in section 3, the purpose is to create clusters with high degree of inner-cluster similarity and low degree of intra-similarity between each pair of them. Therefore according to this explanation and Equation 4, the best measure is ADDC, to be considered as the main assessment method. Since by decreasing the amount of ADDC the overall pheromone value that should be laid on an edge will increase. Generally any fitness function can be replaced by ADDC that its optimal value is almost zero.

4.6 Final Pheromone Update

The last stage before finishing an iteration of the ant colony procedure is to update the pheromones laid on the edges according to Equation 4. The lower the fitness function output, the higher effect will be on the edge passed by the *j*th ant. In simpler terms, the amount of pheromone laid on the edges is influenced by the quality of the solution each ant creates. Solutions with low ADDC

value are considered as high quality. Therefore the ants with high lower fitness value must have more effect on a given edge E_i compared to the ants that passed the same edge but have lower value of ADDC. This fact is reflected in Equation 5 where $\frac{1}{G_k}$. Algorithm 2 is an overview of the SACOClust algorithm.

Algorithm 2: SACOClust

Input:
total number of iterations
total number of ants
desired Number of clusters
Output:
clustered documents
Algorithm:

while iterations are not completed {
Assign M ants to a node randomly
while all the solutions are not created
for total number of ants
Select the next possible node for ant m by selection function;
Update the pheromone of the traversed edge by the ant m;
end for
end while
for total number of ants
Perform the clustering of documents;
Calculate the fitness of ant m;
end for
Update the pheromones values of the edges that passed by each ant
end while

Document purification like removing stop words and porter stemming were done before the algorithm initialization as an offline step. Each document then is represented as a vector in a vector space as a TF-IDF matrix.

4.7 Hybridization

In this section we hybridize the ant colony with *k*-means to provide the opportunity of local search as well. *k*-means and ant colony are integrated when SACOClust is executed for a number of iterations and then *k*-means is applied to the best clustering produced by SACOClust. This variation is known as *k*-means Segregator Clustering (*k*SACOClust).

Ant colony is strong in finding global solutions. The global solution can locate the best possible solution globally, while *k*-means is a localized search, the procedure finds the proximity of the best possible solution (Mahdavi & Abolhassani, 2009). On one hand *k*-means is sensitive to its initial stage, (i.e. the initial centroids selected), which can affect the final result of the procedure. On the other hand *k*-means is good at local search but lacks a global perspective. Therefore it can be concluded that hybridization of these algorithms is good enough to outperform either one individually. The procedure of hybrid algorithm is shown in Algorithm 3. In the hybrid algorithm ant colony provides globally best centroids while *k*-means can search around the global solution to fine-tune the final result.

Algorithm 3: *k*SACOClust

Input:
Total number of iterations
Total number of ants
Desired Number of clusters
Output:
Clustered documents

Assign M ant to a node randomly
while all the solutions are not created
for total number of ants
Select the next possible node for ant m by selection probability function;
Update the pheromone of the traversed edge by the ant m;
end for
end while

while iterations are not completed
for *each* ant
Consider the generated centers as initial centroids of the clusters;
for Total number of documents
Select a document;
Assign it to the nearest cluster, C;
end for
Calculate the fitness of ant m;
end for
Update the pheromones values of the edges that passed by the each ant
Update the centroids.
end while

5 Experimental Results

In this section we conduct a series of experiments to demonstrate the merits of the proposed SACO algorithm and to investigate how it performs on document clustering (i.e., SACOClust and *k*SACOClust). To do so, we compare the proposed algorithms with other ant-based text clustering algorithms. Different evaluation measures and datasets are used, as separately discussed in following subsections.

5.1 Performance Evaluation

Here we introduce the three main measures we will use to assess the performance of different clustering algorithms. These measures are the mostly used measures in text mining.

5.1.1 F-measure

Precision and recall are two popular performance measures for text clustering to assess the quality of clustering algorithms. However, neither precision nor recall makes sense in isolation from each other. As it is well known from the IR practice, the higher level of precision may be obtained at the price of low values of recall. To combine precision and recall F-measure is proposed by other experts and researchers. F-measure function attributes equal importance to precision (P) and recall (R), and it is computed as indicated in Equations 6 to 8.

$$P = \text{Precision}(i, j) = \frac{N_{ij}}{N_j} \quad (6)$$

$$R = \text{Recall}(i, j) = \frac{N_{ij}}{N_i} \quad (7)$$

$$\text{F-measure} = \frac{2PR}{P+R} \quad (8)$$

where N_{ij} is the number of members of class *i* in cluster *j*, N_j is the number of members of cluster *j*, and N_i is the number of member of cluster *i*.

In F-measure calculation, each document belongs to a category when it is fed to the algorithm. In fact the categories and class of each document are identified by an expert but we want the algorithm to cluster the documents. Since clustering is an unsupervised procedure, just the number of required clusters is identified, and not the categories label. Therefore class i, is the category that a document. For example d_1, belongs to, according to the documents categories. With respect to class i, we consider the cluster with the Highest F-measure to be the cluster that maps to class i, and that F-measure becomes the score for class i. The overall F-measure for the clustering result C is computed as Equation 9.

$$F_C = \frac{\sum_i \left(|i| * F(i) \right)}{\sum_i |i|} \quad (9)$$

where |i| and F(i) are, the size and F-measure of the class i, respectively. The higher this measure is the better quality is gained as a result of clustering. The unit of this measure is percent (%) that we used it as well, in this work. In fact F-measure indicates how many of the documents that are supposed to be together, after the clustering are grouped or clustered together.

5.1.2 Entropy

The second measure is known as Entropy, which analyzes the distribution of categories in each cluster. The entropy measure looks at how various classes of documents are distributed within each clustering. First, the class distribution is calculated for each cluster and then this value is used to calculate the entropy of each cluster. The entropy of a cluster C_i is as Equations 10 and 11.

$$E(C_i) = -\sum_j P_{ij} \log P_{ij}, \quad (10)$$

where P_{ij} is the probability that a member of cluster j belongs to class i and then the summation of all classes is taken. After the entropy is calculated, the summation of entropy for each cluster is calculated using the size of each cluster as weight.

$$E = -\sum_{i=1}^{K} \frac{n_i}{n} E(C_i), \quad (11)$$

where n_i is the size of cluster *i*, *n* is the total documents, and *K* is the number of clusters. A good document clustering is one that reduces the general entropy as much as possible, while increases the F-measure value.

Entropy measures how diverse a set of documents in a cluster is. In other words, are most of them belonging to the same group, or are many documents that ought to belong to different groups included in a cluster? The lower the diversity of documents is the lower entropy value would be. Therefore higher diversity indicates that in a given cluster from documents from different groups and labels are available (e.g. Education, Politics). In fact the algorithm performed poorly in distinguishing the similar and non-similar documents and led to high amount of entropy. While an F-measure is to some extent an external measure and calculates the percentage of the documents that are supposed to be considered together, according to their initial categories. The higher this rate the more successful the algorithm is at recognizing similar documents, thereby placing them into the same cluster.

5.1.3 ADDC

The third measure is one which computes the average distance within each cluster of the algorithm, known as Average Distance of Documents to the Cluster (ADDC) centroid. This formula is represented as Equation 12.

$$ADDC = \frac{\sum_{i=1}^{k} \left(\frac{\sum_{j=0}^{n_i} D(c_i, d_{ij})}{n_i} \right)}{K} \quad (12)$$

where *K* is the number of clusters, n_i is the number of documents in cluster *i*, $D(.,.)$ is distance function, and d_{ij} is the *j*th document of cluster *i*. This measure is also introduced in some papers as intra cluster similarity; in which it refers to average of similarities inside each cluster. Each of these measures is considered as the fitness for ant colony algorithms in this paper, separately. In simpler term the solutions are created based on the main fitness function ADDC, and then the same solutions are evaluated against F-measure and entropy.

5.2 Datasets

In the conducted experiments a wide range of standard datasets are used to evaluate the effectiveness and performance of the algorithms; as summarized in Table 2. The first data set UWC is a collection of 314 Web documents manually collected and labeled from various University of Waterloo and Canadian Web sites. It is categorized manually based on topic description. This data set has a moderate degree of overlap between the different classes.

The second dataset 20NEWS is a subset of the full 20-newsgroups collection of Usenet news group articles. This data set is available from the UCI KDD Archive. Each news group constitutes a different category, with varying overlap between them; some news groups are much related (e.g. talk.politics.mideast & talk.politics.misc) while others are not related at all (e.g. comp. graphics & talk.religion.misc). YN is a collection of Reuters news posted on Yahoo! News site extracted from 20 categories.

Table 2 – Textual datasets for experimentation

	UW-CAN	20NEWS	Yahoo!	DMOZ	Politics	NEWS	Reuters
Acronym	UWC	20NEWS	YN	DM	PL	NWS	RS
#Docs	314	2000	2340	700	176	424	1313
#Cat	10	20	20	14	10	10	6

In RS pages are extracted from 6 different categories of coffee, sugar, interest, money-fx, trade and crude to compare our work with one proposed in (Zhang, et al., 2010). The datasets has been derived from Reuters-21578.

DM dataset, consists of 14 subcategories are constructed from subcategories of Art, Business,

Computer, Game, Health, Home, Recreation, Reference, Science, Shopping, Society, Sport, News, and Regional, to compare the proposed algorithm with other similar works in (Zhang, et al., 2010). PL dataset is collected from Politics area and contains 176 web documents that are selected randomly in some topics of Politics. Dataset NWS is collected from News sites and contains 424 different news texts this along with other datasets PL and NWS are collected in 2006 (Mahdavi, et al., 2008).

5.3 Comparisons

In this subsection, conventional ant colony algorithms including AS, MMAS and ACS are implemented as text clustering methods and compared with our proposed methods. We note that these algorithms are not originally proposed for text clustering purpose and we treat them as document clustering algorithm. As mentioned earlier, we assess the performance of algorithms in terms of three measures of Entropy, ADDC and F-measure.

In Table 3, ADDC values of SACOClust and *k*SACOClust are compared to other three conventional ant based algorithms of AS, MMAS and ACS. In datasets of 20NEWS, PL, UWC and RS, SACOClust had the lowest rates of ADDC that makes the algorithm as a distinct one, while in other datasets such as YN, SACOClust had a similar or superior performance comparing to other competitors. In some other datasets such as DM, SACOClust was inferior to ACS and MMAS. Using F-measure as the other assessment measure, *k*SACOClust in datasets of 20NEWS, RS, UWC and YN has the highest value while in DM, SACOClust has the best performance. Besides that, SACOClust in PL and NWS datasets has a similar performance comparing to its other competitors.

Entropy as the other evaluation measure, in *k*SACOClust for dataset of 20NEWS, YN, UWC and RS has the lowest values, while in DM dataset the best performance was shown by SACOClust. The behaviour of the proposed algorithms is varied in different datasets, but what is very common among most of the datasets is the superiority of the either SACOClust or *k*SACOClust, over the conventional variations, in most datasets. We continue our experiments by comparing the proposed methods with the state-of-the-art ant colony algorithms of (Niknam & Amiri, 2010) and (Zhang & Feng, 2012) that are implemented as text clustering algorithms.

Table 4, illustrates the comparisons between different ant colony algorithms and the proposed ant colony based algorithms, using seven datasets. In 20NEWS and NWS datasets SACOClust has outperformed most of the other ant-based variations, while were inferior in compare to TSIACO1 and TSIACO2. In the other dataset of DM SACOClust had the lowest value of ADDC that showed a significant superiority most of the other algorithms, except in TSIACO2, that TSIACO could outperform all other ant-based algorithms including the ones proposed in this paper. In other datasets of PL, RS, UWC and YN, SACOClust also have superiority, over its competitors.

In Table 4, the proposed algorithms are also compared in terms of entropy. *k*SACOClust could out perform its competitors in datasets including NWS, 20NEWS, YN, UWC and RS. Therefore integration of SACOClust and *k*-means had a significant improvement in the performance of SACOClust in terms of entropy measure. Besides that, in PL dataset the lowest rate of entropy belongs to FAPSO-ACO-K that makes it as a distinctive algorithm. In DM dataset the proposed algorithms had the same performance with TSIACO1 and TSIACO2.

The other popular measure of assessment is F-measure that was used to compare the performance of the proposed algorithms with other ant algorithms. In three datasets of UWC, YN and RS the superiority of *k*SACOClust over the other algorithms is significant, while in two other datasets of PL and NWS, FAPSO-ACO could outperform all other competitors.

It can be concluded that the proposed algorithms (either SACOClust or *k*SACOClust) mainly have superiority in compare to other algorithms (both traditional and state-of-the-art algorithms), that this success would be stemmed from the quality function that is used in these variations. In the traditional variations of the ant colony algorithms (i.e. AS, ACS, MMAS) and the state of the art ones (i.e. TSIACO1, TSIACO2, FAPSO-ACO and FAPSO-ACO-K) there is no emphasis on highlighting paths of the solution space leading to high separation of the irrelevant data. This conclusion can be drawn according to two measures of ADDC and entropy.

6 Conclusion and Future Works

In this paper we have proposed a novel variant of ant colony optimization method that is capable of segregating dissimilar solution paths which makes it more suitable for document clustering. The new method is called the Segregator Ant Colony Optimization (SACO). The main feature of SACO algorithm is its capability in identifying and magnifying paths in the solution space that potentially lead to higher separation of data in the final partitioning.

In the proposed clustering algorithm based on SACO that is referred to as SACOClust, a group of artificial ants select K documents randomly as cluster centers of each cluster and then the rest of the documents are assigned to the clusters.

Then, ants search across the solution space to look for the best possible solution that most satisfies the optimization measure. Although the SACOClust is good at locating the proximity of optimal solution, but it behaves poorly in fine-tuning the solution to boost the performance around that point. Therefore, we combined the *k*-means and SACOClust methods to take advantage of both methods, i.e., the explorative power of former and fine-tuning power of latter. Based on the experiments we have conducted on different datasets, it can be inferred that the proposed method has superiority over the existing baseline and state-of-the-art algorithms.

Table 3 - Comparisons with baseline ant colony optimization methods adopted for clustering

Dataset	F-measure					Entropy					ADDC				
	SACOClust	*k*SACOClust	AS	ACS	MMAS	SACOClust	*k*SACOClust	AS	ACS	MMAS	SACOClust	*k*SACOClust	AS	ACS	MMAS
20NEWS	0.13	0.44	0.13	0.12	0.14	0.2	0.08	0.2	0.2	0.42	0.29	0.32	0.32	0.32	0.3
DM	0.18	0.15	0.161	0.169	0.17	0.26	0.27	0.267	0.268	0.266	0.04	0.44	0.16	0.35	0.05
PL	0.25	0.21	0.25	0.25	0.24	0.38	0.3	0.27	0.27	0.27	0.02	0.22	0.07	0.06	0.04
NWS	0.19	0.18	0.19	0.19	0.19	0.45	0.34	0.31	0.31	0.31	0.04	0.24	0.03	0.03	0.14
RS	0.52	0.62	0.26	0.26	0.26	0.32	0.3	0.32	0.32	0.32	0.1	0.43	0.17	0.16	0.16
UWC	0.24	0.6	0.29	0.23	0.21	0.44	0.13	0.44	0.45	0.46	0.07	0.26	0.107	0.106	0.108
YN	0.21	0.49	0.27	0.29	0.18	0.16	0.07	0.15	0.15	0.16	0.29	0.46	0.29	0.31	0.3

Table 4 - Comparisons with state-of-the-art ant colony algorithms

Measure	Algorithms	20NEWS	DM	PL	NWS	RS	UWC	YN
ADDC	SACOClust	0.29	0.04	0.02	0.04	0.1	0.07	0.24
	*k*SACOClust	0.32	0.44	0.22	0.24	0.43	0.26	0.45
	FAPSO-ACO-K	0. 48	0.17	0.31	0.17	0.35	0.08	0.29
	FAPSO-ACO	0.45	0.23	0.04	0.23	0.21	0.27	0.28
	TSIACO1	0.23	0.07	0.04	0.03	0.17	0.1	0.29
	TSIACO2	0.26	0.02	0.06	0.055	0.15	0.11	0.3
Entropy	SACOClust	0.2	0.26	0.38	0.45	0.32	0.42	0.16
	*k*SACOClust	0.08	0.27	0.3	0.42	0.3	0.13	0.07
	FAPSO-ACO-K	0.12	0.27	0.24	0.46	0.31	0.28	0.16
	FAPSO-ACO	0.2	0.28	0.27	0.47	0.33	0.14	0.17
	TSIACO1	0.18	0.26	0.4	0.45	0.4	0.45	0.23
	TSIACO2	0.22	0.26	0.38	0.45	0.4	0.44	0.16
F-measure	SACOClust	0.13	0.18	0.25	0.19	0.52	0.24	0.21
	*k*SACOClust	0.44	0.15	0.21	0.18	0.62	0.6	0.49
	FAPSO-ACO-K	0.12	0.15	0.41	0.18	0.27	0.26	0.21
	FAPSO-ACO	0.17	0.25	0.43	0.25	0.32	0.34	0.34
	TSIACO1	0.19	0.18	0.3	0.22	0.32	0.21	0.28
	TSIACO2	0.11	0.18	025	0.2	0.32	0.26	0.28

7. Bibliography

Arthur, D. & Vassilvitskii, S., 2007. *K-means++: the advantages of careful seeding.* s.l., s.n., pp. 1027-1035.

Azzag, H., Venturini, G., Oliver, A. & Guinot, C., 2007. A hierarchical ant based clustering algorithm and its use in three real-world applications. *European Journal of Operational Research,* 179(3), pp. 906-922.

Chehreghani, M., Abolhassani, H. & Chehreghani, M., 2008. Improving density-based methods for hierarchical clustering of web pages. *Data and Knowledge engineering,* 67(1), pp. 30-50.

Das, S., Abraham, A. & Konar, A., 2008. Automatic clustering with a multi-elitist particle swarm optimization algorithm. *Pattern Recognition Letters,* 29(1), pp. 688-699.

Dorigo, M., Maniezzo, V. & Colorni, A., 1996. Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics—Part B,* 26(1), pp. 29-41.

Dziwiński, P., Bartczuk, Ł. & Starczewski, J., 2012. Fully Controllable Ant Colony System for Text Data Clustering. In: *Lecture Notes in Computer Science.* s.l.:s.n., pp. 199-205.

Forsati, R., Mahdavi, M., Shamsfard, M. & Meybod, M., 2012. Efficient stochastic algorithms for document clustering. *Information Science,* Volume 220, pp. 269-291.

Forsati, R. et al., 2014. Enriched ant colony optimization and its application in feature selection. *Neurocomputing.*

Forsati, R., Moayedikia, A. & Keikhah, A., 2012. A Novel Approach for Feature Selection based on the Bee Colony Optimization. *International Journal of Computer Applications,* 43(8), pp. 30-34..

Gambardella, L. & Dorigo, M., 1996. *Solving Symmetric and Asymmetric TSPs by Ant Colonies.* s.l., s.n., p. 622–627..

Hammouda, K. & Kamel, M., 2004. Effiecient Phrase-Based Document Indexing for Web Document Clustering. *IEEE Transaction on Knowledge and data Engineering,* 16(10), pp. 1279-1296.

He-Nian, C. et al., 2009. *A Text Clustering Method Based on Two-Dimensional OTSU and PSO Algorithm.* s.l., s.n., pp. 1-4.

Huang, J., Ng, . M. K., Rong, H. & Li, Z., 2005. Automated variable weighting in k-means type clustering. *IEEE Transaction on Pattern Analysis and Machine Intelligence,* 27(5), pp. 657-668.

Hui, F., Chao-tian, C. & Xiao-yong, L., 2009. *An Improvement Text Clustering Algorithm Based on Ant Colony.* s.l., s.n., pp. 782-785.

Kang, J. & Zhang, W., 2012. Combination of Fuzzy C-Means and Particle Swarm Optimization for Text Document Clustering. *Advances in Intelligent and Soft Computing,* Volume 139, pp. 247-252.

Kanungo, T., Mount, D., Netanyahu, N. & Piatko, C., 2004. A local search approximation algorithm for k-means clustering. *Computational Geometry,* 28(2-3), pp. 89-112.

Kennedy, J. & Eberhart, R., 1995. *Particle Swarm Optimization.* s.l., s.n., pp. 1942-1948..

Kim, L. et al., 2008. Hybrid ant colony algorithms for path planning in sparse graphs. *soft computing ,* 12(10), pp. 981-994.

Labroche, N., Monmarche, N. & Venturini, G., 2003. *AntClust: ant clustering and web usage mining.* s.l., s.n., pp. 25-36.

Li, P., Wang, B. & Jin, W., 2012. Improving Web Document Clustering through Employing User-Related Tag Expansion Techniques. *Journal of Computer Science and Technology,* 27(3), pp. 554-566.

Li, Y., Chung, S. & Holt, J., 2008. Text document clustering based on frequent word meaning sequences. *Data and Knowledge Engineering,* Volume 64, pp. 381-404.

Mahdavi, M. & Abolhassani, H., 2009. Harmony k-means algorithm for document clustering. *Data and Knowledge Discovery,* Volume 18, pp. 370-391.

Mahdavi, M., Chehreghani, M., Abolhassani, H. & Forsati, R., 2008. Novel meta-heuristic algorithms for clustering web documents. *Applied Mathematics and Computation,* 201(1-2), pp. 441-451.

Min-Thai, W., Hong, T. & Lee, C., 2012. A continuous ant colony system framework for fuzzy data mining. *soft computing ,* 16(12), pp. 2071-2082.

Mora, A., Garcia-Sunchez, P., Merelo, J. & Castillo, P., 2013. Pareto-based multi-colony multi-objective ant colony optimization algorithms: an island model proposal. *soft computing,* 17(7), pp. 1175-1207.

Niknam, T. & Amiri, B., 2010. An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis. *Applied Soft Computing,* Volume 10, pp. 183-197.

Porter, M., 1980. *An Algorithm for Suffix Stripping.* s.l., Morgan Kaufmann Publishers Inc, pp. 130-137.

Stutzle, T. & Hoos, H., 2000. Max-min Ant System. *Future Generation Computer Systems,* 16(8), p. 889–914..

Xu, R. & Wunsch, D., 2005. Survey of clustering algorithms. *IEEE Transaction on Neural Networks,* 16(3), pp. 645 - 678.

Yang, Y., Kamel, M. & Jin, F., 2005. *A model of document clustering using ant colony algorithm and validity index.* s.l., s.n., pp. 2730-2735.

Zhang, F., Ma, Y., Hou, N. & Liu, H., 2008. *An Ant-Based Fast Text Clustering Approach Using Pheromone.* s.l., s.n., pp. 385-389.

Zhang, W., Yoshida, T., Tang, X. & Wang, Q., 2010. Text clustering using frequent itemsets. *Knowledge-Based Systems,* 23(5), pp. 379-388.

Zhang, Z. & Feng, Z., 2012. Two-stage updating pheromone for invariant ant colony optimization algorithm. *Expert Systems with Applications,* Volume 39, pp. 706-712.

Spectral Methods for Immunization of Large Networks

Muhmmad Ahmad[1] **Juvaria Tariq**[1] **Muhammad Farhan**[1]
Mudassir Shabbir[2] **Imdadullah Khan**[1]

[1] Dept. of Computer Science, Lahore University of Management Sciences,
[2] Dept. of Computer Science, Information Technology University, Lahore

Abstract

Given a network of nodes, minimizing the spread of a contagion using a limited budget is a well studied problem with applications in network security, viral marketing, social networks, and public health. In real graphs, virus may infect a node which in turn infects its neighbor nodes and this may trigger an epidemic in the whole graph. The goal thus is to select the best k nodes (budget constraint) that are immunized (vaccinated, screened, filtered) so as the remaining graph is less prone to the epidemic. It is known that the problem is, in all practical models, computationally intractable even for moderate sized graphs. In this paper we employ ideas from spectral graph theory to define relevance and importance of nodes. Using novel graph theoretic techniques, we then design an efficient approximation algorithm to immunize the graph. Theoretical guarantees on the running time of our algorithm show that it is more efficient than any other known solution in the literature. We test the performance of our algorithm on several real world graphs. Experiments show that our algorithm scales well for large graphs and outperforms state of the art algorithms both in quality (containment of epidemic) and efficiency (runtime and space complexity).

Keywords: Graph Immunization, eigendrop, closed walks

1 Introduction

Consider a *large* network of hospitals distributed, geographically, over a region. Due to presence of an active pathogen, there is a danger of pandemic in the region. We assume that pathogen can easily travel between *linked* hospitals. We are concerned with keeping the maximum of the hospital network clean of pandemic. Based on their geographic vicinity (or some other criteria) , it is known for every pair of hospitals X, Y whether contamination at X forebodes certain contamination at Y. If this is true about some pair of hospitals X, Y, we call X and Y as linked or *connected*. Now a team of scientists have developed a vaccine for the pathogen. Unfortunately, due to cost or production constraints the vaccine is only available in limited quantity and only a small fraction of hospitals can receive it. It is assumed that once a hospital receives the vaccine it can not be contaminated nor can it contaminate any other hospital. Given the scarcity of the vaccine resource it is a very serious question to ask whether a given hospital should or should not receive the vaccine so as to minimize the overall spread of the pathogen in the region. This question, in essence, is the topic of this work, which appears in various other scenarios. An efficient solution to this problem can be applied to diverse array of high-impact applications in public health. It will also be an important tool in cyber-security solutions. Finding important players in social networks is a fundamental problem in viral marketing, online advertisement and social networks monitoring. In its essence, the problem of finding important nodes in a network can be reduced to the main problem of this paper.

This is known, in the literature, as the *Network Immunization Problem*. The problem is studied in terms of graphs where each node represents a hospital (or any other resource-hungry entity) and an edge between a pair of nodes represents the *link* between the pair of hospitals. We use the SIS model of infection spread in the network that is detailed in section3. We will start with the following formulation of the problem that appears in (Chen & Chau 2016).

Problem 1. *Given a simple undirected graph $G = (V, E)$ and an integer k find a set S of k nodes such that if we " immunize" S, renders G least "vulnerable" to an attack over all choices of S.*

Clearly, the problem is ill-defined, unless a precise and quantifiable definition of the *vulnerability* of a network is provided. To this end, it turns out that the largest eigenvalue of the adjacency matrix of the graph is a good measure of vulnerability of the graph (Chakrabarti & Faloutsos 2008).

Largest Eigenvalue Problem For a simple undirected graph G on n nodes, adjacency matrix A_G (or just A whenever the graph G is obvious from the context) is an $n \times n$ binary matrix with columns and rows representing the vertices of G and cell entries representing the edges i.e. $A_G(i, j) = 1$ if and only if there is an edge from vertex i to vertex j. Eigen spectrum, $\{\lambda_i\}_{i=1}^{n}$, of adjacency and Laplacian matrices has been studied a lot for certain graph properties (Chung 1997). Of particular interest to us is the largest eigenvalue (denoted by $\lambda_1(A)$ or λ_1 when G is obvious from the context) of the the adjacency matrix also referred to as the first eigenvalue in literature. It is known that λ_1 is related to the connectivity measures of the graph. For example we know that

$$\Delta \geq \lambda_1 \geq d_{avg} \tag{1}$$

where Δ and d_{avg} are the maximum and average degrees in the graph, respectively [(West 2001, p. 459)].

In the context of the discussion above it seems only logical to delete/vaccinate the vertices in G such that the remaining graph has minimum possible largest eigenvalue.

Let G be a graph of order n, with adjacency matrix A. For a subset of vertices $S \subseteq V(G)$, we denote by G_{-S} the subgraph induced on vertices in $V \setminus S$. Similarly, for $S \subseteq V(G)$, we define A_{-S} to be the submatrix of A obtained by deleting rows and columns in A at corresponding indices in S. By definition of induced subgraph, it is clear that A_{-S} is the adjacency matrix of G_{-S}. We will denote by G_S the subgraph of G induced by vertices in S and A_S denotes the adjacency matrix of G_S. The precise formulation of the above problem is given as follows:

Problem 2. *Given a simple undirected graph $G = (V, E)$ find a set S of k vertices so that $\lambda_1(A_{-S})$ is the minimum possible over all choices of $S \subseteq V$.*

While computing any eigenvalue of a graph can be accomplished in time polynomial in n, it turns out that solving Problem 2 is NP-Hard.

In this work we propose a novel algorithm to approximately solve Problem 2. Our algorithm is quite intuitive, promises a better approximation than all known approaches, and is quite easy to implement. Moreover, theoretical and practical time complexity of our algorithm beats state of the art. Along with the algorithm we propose, ideas developed in the course of its development provide a good insight into the graph connectivity structures. We hope that this approach is proved to be useful in design of practical graph algorithms in the future. In the end we perform extensive experiments to test and compare the approximation to current best known algorithms and the benchmark.

Rest of the paper is organized as follows. In Section 2 we provide a detailed background to Problem 2 and discuss its computational intractability and approaches to approximate it. An extensive literature review of the immunization problem is provided in 3 . We propose our algorithm in Section 4 along with its approximation guarantee complexity analysis. Section 5 contains results of tests of our algorithms on real world graphs. This section also provides comparison of performance of our algorithm with other known algorithms. We conclude with a discussion on future directions in section 6.

2 Background

The problem definition suggests a brute-force solution that checks the eigendrop achieved by each possible subset of size k and selects the subset achieving the largest eigendrop. Since computing the largest eigenvalue of a graph can be accomplished in $O(m)$ (Chen & Chau 2016), hence runtime of this method is $O(\binom{n}{k} \cdot m)$, which is exponential in size of input (k).

Indeed, it turns out that solving Problem 2 optimally is NP-Hard. A simple reduction from Minimum Vertex Cover Problem goes as follows: If there exist k vertices that cover all the edges in the graph, then deleting those vertices will imply that $\lambda_1(A_{-S}) = 0$. Similarly if there exists a k-set S such that $\lambda_1(A_{-S}) = 0$, then S must be a vertex cover; since otherwise it contradicts the following implication of the Perron-Frobenius theorem.

Fact 1. *(Serre 2002) Deleting any edge from a simple connected graph G strictly decreases the largest eigenvalue of the corresponding adjacency matrix.*

For a reduction from Max Independent Set problem that doesn't use Perron-Frobenius theorem see appendix of (Chen & Chau 2016).

Although Problem 1 is NP-Hard, the following greedy algorithm guarantees a $(1 - 1/e)$-approximation to the optimal solution to Problem 2. Approximation guarantee of GREEDY-1 follows from

Algorithm 1 : GREEDY-1(G,k)

$S \leftarrow \emptyset$
while $|S| < k$ **do**
  $v \leftarrow \arg\max_{x \in V \setminus S} (\lambda_1(A_{-\{S \cup \{x\}\}}))$
  $S \leftarrow S \cup \{v\}$
return S

Theorem 1.

Theorem 1. *(Nemhauser & Fisher 1978) Let f be a non-negative, monotone and submodular function, $f : 2^\Omega \to \mathbb{R}$. Suppose $\mathcal{A}$ is an algorithm, that choose a k elements set S by adding an element u at each step such that $u = \arg\max_{x \in \Omega \setminus S} f(S \cup \{x\})$. Then $\mathcal{A}$ is a $(1 - 1/e)$-approximate algorithm.*

For sparse graphs largest eigenvalue can be computed in $O(m)$, running time of GREEDY-1 amounts to $O(knm)$. This runtime is impractical for any reasonably large real world graph.

In (Chen & Chau 2016) each subset was assigned a score, called *shield-value* that was meant to approximate eigendrop achieved by that set. For a set S of size k, shield value , $Sv(S)$ can be computed in $O(k^2)$ when the eigenvector corresponding to largest eigenvalue is known. Hence, the straightforward method of finding the set with largest shield value takes $O(\binom{n}{k} \cdot k^2)$ time. Using the fact that the objective function based on shield value is submodular, (Chen & Chau 2016) gave a greedy algorithm with runtime $O(nk^2 + m)$. As for GREEDY-1, by Theorem 1, their algorithm guarantees a $(1 - 1/e)$-approximation to the optimal shield value.

3 Related work

While the focus of this paper is to target node immunization problem using spectral graph theoretic techniques, there is vast amount of literature on this problem with approaches from diverse areas of the subject. Initially in 2003 Brieseneister, Lincoln and Porras (Briesemeister & Porras 2003) studied the propagation styles of viruses in communication networks. Along with this, the effects of graph topology in the spread of an epidemic are described by Ganesh, Massoulié and Towsley in (Ganesh & Towsley 2005) and they discuss the conditions under which an epidemic will eventually die out. Similarly Chakrabarti et. al in (Chakrabarti & Faloutsos 2008) devise a non linear dynamical system (NLDS) to model virus propagation in communication networks. They use the idea of *birth rate, β, death rate,δ, and epidemic threshold,τ,* for a virus attack where birth rate is the rate with which infection propagates, death rate is the node curing rate and epidemic threshold is a value such that if $\beta/\delta < \tau$, infection will die out quickly else if $\beta/\delta > \tau$ infection will survive and will result in an epidemic. Virus propagation is studied in both directed and undirected graphs. For undirected graphs, they prove that epidemic threshold τ equals $1/\lambda$ where λ is largest eigenvalue of adjacency matrix A of the graph.

Thus for a given undirected graph, if $\beta/\delta<1/\lambda$, then the epidemic will die out eventually. But none of these specifically discuss the graph immunization problem.

Afterwards the problem is studied using the edge manipulation scheme. In (Kuhlman & Ravi 2013) dynamical systems are used to delete appropriate edges to minimize contagion spread. While Tong et al. in (Tong & Faloutsos 2012) remove k edges from the graph in a manner that eigendrop (difference in largest eigenvalues of original and resultant graphs) is maximized. For this edges are selected on the basis of left and right eigenvectors of leading eigenvalue of the graph such that for each edge e_x, score(e_x) is the dot product of the left and right eigenvectors of leading eigenvalue of adjacency matrix of A.

Graph vulnerability is defined as measure of how much a graph is likely to be affected by a virus attack. As in (Tong & Faloutsos 2012), the largest eigenvalue is selected as a measure of graph vulnerability , in (Chen & Chau 2016) they also use largest eigenvalue for the purpose but instead of removing edges, nodes are deleted to maximize the eigendrop. Undirected, unweighted graphs are considered and nodes are selected by an approximation scheme using the eigenvector corresponding to largest eigenvalue which cause the maximum drop.

Probabilistic methods are also used for node immunization problem. Zhang et al. and Song et al. adapt the non-preemptive strategy i.e selection of nodes for immunization is done after the virus starts propagating across the graph. For this they use discrete time model to obtain additional information of infected and healthy nodes at each time stamp. In (Song & Lee 2015) directed and weighted graphs are used in which weights represent the probability of a healthy node being infected by its neighbors and node selection is done on the basis of these probabilities. Then results are evaluated on the basis of save ratio (SR) which is the ratio between the number of infected nodes when k nodes are immunized over the number of infected nodes when no node is immunized. The work in (Zhang & Prakash 2014*a*) and (Zhang & Prakash 2014*b*) considers undirected graphs and incorporates dominator trees for selecting nodes. Results are evaluated in terms of expected number of remaining infected nodes in the graph after the process of immunization.

Other important and closely related problem is k facility location and a lot of work is done on this. In filter placement (Erdös & Bestavros 2012), those nodes are identified which are cause of maximum information multiplicity. Moreover some reverse engineering techniques are also used for similar problems. Prakash, Vreeken and Faloutsos (Prakash & Faloutsos 2012) study the graphs in which virus has already spread for some time and they point out those nodes from where the spread started. From this they find out the likelihood of other nodes being affected.

Another direction to look at the problem is to consider graphs in which some nodes are already infected and these nodes can spread virus among other reachable nodes or graphs in which all nodes are contaminated and the goal is to decontaminate the graph by using some agent nodes which traverse along the edges of the graph and clean the nodes. The problem is usually referred to as decontamination of graph or graph searching problem. Different models are studied to solve the problem and most of them assume the monotonicity in decontamination i.e once a node is decontaminated then it can not get contaminated again (Bienstock & Seymour 1991),(Flocchini & Luccio 2008),(Flocchini & Luccio 2007),(Fraigniaud & Nisse 2008). But non-monotonic strategies are also studied (Daadaa & Shabbir 2016).

Other work that is related to node immunization is the selection of most influential nodes in a given network to maximize the diffusion of new information in a network. Kempe et al. provided the provably efficient approximation algorithm for the problem (Kempe & Tardos 2003). Seeman and Singer (Seeman & Singer 2013) use stochastic optimization models to maximize the information diffusion in social networks. Influence maximization problem is slightly different from immunization problem as in influence maximization problem the goal is to select nodes for seeding which will maximize the spread on new idea while in node immunization problem goal is to select nodes which will help in minimal spread of virus.

4 Our Proposed Algorithm

In (Chen & Chau 2016) they defined shield value that was an approximation to the eigendrop. We define our shield value to be the eigendrop and approximate the eigenvalue computation instead. We achieve better theoretical guarantees with our shield value definition. Furthermore, it is easy and computationally efficient to approximate our shield value.

4.1 Our shield value and its justification

We use the following well known facts from linear algebra and graph theory. Given a $n \times n$ matrix A,

Fact 2. *[c.f. (Strang 1988)]*

$$tr(A) = \sum_{i=1}^{n} a_{ii} = \sum_{i=1}^{n} \lambda_i(A)$$

Fact 3. *[c.f. (West 2001) p. 455]*

$$tr(A^p) = \sum_{i=1}^{n} \lambda_i(A^p) = \sum_{i=1}^{n} \lambda_i(A)^p$$

Clearly, for even powers p, $tr(A^p)$ is an upper bound on λ_1^p and as p grows $tr(A^p)$ approaches $\lambda_{max}^p(A)$ where $\lambda_{max} = max\{|\lambda_i(A)|\}$. We therefore, try to remove a set S of k vertices from the graph such that $tr((A_{-S})^p)$ is minimized. The goodness of a set S in this setting is defined as

$$f_p(S) = tr((A_{-S})^p) \tag{2}$$

Define

$$g_p(S) = tr(A^p) - tr((A_{-S})^p) \tag{3}$$

Note that minimizing $f_p(S)$ is same as maximizing $g_p(S)$.

Given a graph $G = (V, E)$, a walk W of length l in G is a sequence of vertex $v_0, v_1, \ldots, v_l$ such that for $0 \leq i \leq l-1$ $(v_i, v_{i+1}) \in E$. We say that W is a walk from v_0 to v_l. If $v_0 = v_l$, then W is called a closed walk. For $v \in V$, let $\mathcal{CW}_i(G, v)$ be the set of closed walks of length i in G containing the vertex v. Suppose $CW_i(G, v) = |\mathcal{CW}_i(G, v)|$. When $X \subset V$, $\mathcal{CW}_i(G, X) = \bigcup_{x \in X} \mathcal{CW}_i(G, x)$. We similarly denote by $CW_i(G, X)$ to be the cardinality of the set $\mathcal{CW}_i(G, X)$. When $X = V$, we refer to $\mathcal{CW}_i(G, X)$ as $\mathcal{CW}_i(G)$. So $\mathcal{CW}_i(G)$ is the set of all closed walks of length i in G.

We use the following well-known fact from graph theory

Fact 4. *[c.f. (West 2001) p.455] Given a graph G with adjacency matrix A,*

$$CW_p(G) = tr(A^p) = \sum_{i=1}^{n} \lambda_i^p.$$

It is clear from the definitions and fact 4 that for $S \subset V$,

$$CW_p(G) = CW_p(G_{-S}, V \setminus S) + CW_p(G, S)$$

This identity is same as the one already mentioned in (3). This similarly suggests a strategy namely one should delete a set S of k vertices such that $CW_p(G_{-S}, V \setminus S)$ is minimized for a large even integer p (equivalently $CW_p(G, S)$ is maximized). The goodness of a set of vertices $S \subseteq V$, in terms of number of closed walks is given by

$$g_p(S) = CW_p(G, S) \tag{4}$$

which as noted above we would like to maximize. Hence we reduced the problem of finding a set maximizing the eigendrop to the problem of finding a set of vertices that are part of many closed walks. We note that this latter problem is of a combinatorial nature and more amenable to techniques from graph theory.

Algorithm 2 : GREEDY-2(G,k,p)

$S \leftarrow \emptyset$
while $|S| < k$ **do**
  $v \leftarrow \arg\max_{v \in V \setminus S} \mathcal{CW}_p(G_S, v)$
  $S \leftarrow S \cup \{v\}$
return S

An algorithm based on this intuition is given in Algorithm GREEDY-2(G,k,p). Quality guarantee of Algorithm GREEDY-2(G,k,p) follows from submodularity of the optimization function (4) and Theorem 1. Next we show that the objective function given in (4) is non-negative, monotone submodular, i.e. we show that $g_p(S)$ has the property of diminishing return.

Lemma 1. *The function $g_p(S)$ given in (4) is non-negative, monotonically non-decreasing and submodular.*

Proof. Since $g_p(S)$ counts the number of closed walks of length p, it is clearly non-negative. By definition of $g_p(S)$ for $X \subseteq Y$ we have

$$\begin{aligned}
&g_p(Y) - g_p(X) \\
&= CW_p(G, Y) - CW_p(G, X) \\
&= CW_p(G, (Y \setminus X) \cup X) - CW_p(G, X) \\
&= CW_p(G, X) + CW_p(G_{-X}, Y \setminus X) - CW_p(G, X) \\
&= CW_p(G_{-X}, Y \setminus X) \geq 0,
\end{aligned}$$

where the last inequality follows from non-negativity of $g_p(S)$. Hence $g_p(S)$ is monotonically non-decreasing function.

For submodularity of $g_p(S)$, let $X, Y, Z \subset V$ such that $X \subset Y$ and $Z \cap Y = \emptyset$. Let $L = X \cup Z$ and $R = Y \cup Z$. We have

$$\begin{aligned}
& (g_p(R) - g_p(Y)) - (g_p(L) - g_p(X)) \\
= \;& CW_p(G, R) - CW_p(G, Y) - CW_p(G, L) \\
& + CW_p(G, X) \\
= \;& CW_p(G, Y \cup Z) - CW_p(G, Y) \\
& - CW_p(G, X \cup Z) + CW_p(G, X) \\
= \;& CW_p(G, Y) + CW_p(G_{-Y}, Z) - CW_p(G, Y) \\
& - CW_p(G, X) - CW_p(G_{-X}, Z) + CW_p(G, X) \\
= \;& CW_p(G_{-Y}, Z) - CW_p(G_{-X}, Z) \leq 0
\end{aligned}$$

where the last inequality follows from the fact that by definition $\mathcal{CW}_p(G_{-Y}, Z) \subseteq \mathcal{CW}_p(G_{-X}, Z)$, hence $CW_p(G_{-Y}, Z) \leq CW_p(G_{-X}, Z)$ □

For a given vertex v, $CW_p(G, v)$ can be computed by a running a breadth first search from v and it takes time proportional to $n + m$.

The Algorithm GREEDY-2(G,k,p) is expensive in terms of runtime for large graphs and it is practically impossible to compute $CW_p(G, v)$ for every vertex v of the graph when p is large and order of G is large enough as well. So we set $p = 4$ to approximate the result and our practical experimentation shows that $p = 4$ gives good enough quality guarantee.

First we give a closed form expression for $CW_4(G, v)$ in terms of degrees and codegrees. For a given graph G, $N_G(x) = \{y \in V(G) : A_G(x, y) = 1\}$ and $d_G(x) = |N_G(x)|$. Define $N_G(x, y) = \{z \in V(G) : A_G(x, z) = 1 \wedge A_G(y, z) = 1\}$ to be the common neighborhood of x and y in G. Let $d_G(x, y) = |N_G(x, y)|$, note that $N_G(x, x) = N_G(x)$ and $d_G(x, x) = d_G(x)$. When G is clear in the context, we refer to $N_G(x, y)$ as $N(x, y)$ and similarly to $d_G(x, y)$ as $d(x, y)$.

Lemma 2. *For any vertex $v \in V$,*

$$CW_4(G, v) = 2d(v)^2 + 4 \sum_{u \in V, u \neq v} d(u, v)^2.$$

Proof. A closed walk of length 4 can be interpreted as the concatenation of two walks of length 2 with same end points u and v. Also, the number of walks of length 2 with the end points u, v are $d(u, v)$.

So for a fixed vertex v, number of closed walks of length 4 starting at v are

$$\sum_{u \in V(G)} d(v, u)^2$$

Let such a walk be $v - a - b - c - v$. Note that this is also a closed walk of length 4 starting at a as $a - b - c - v - a$. So for each closed walk starting at v, there are 3 other closed walks containing v.

We get number of closed walks of length 4 containing v as $4 \sum_{u \in V(G)} d(v, u)^2$. But these may not be distinct walks and it can easily be seen that when walks starting from b are considered, b can not be equal to v. And number of these walks are $d(v)^2$. Similarly for the walks starting at c, hence $d(v)^2$ walks are counted twice.

As required, for any vertex v, we get

$$\mathcal{CW}_4(G, v) = 4 \sum_{u \in V(G)} d(u, v)^2 - 2d(v)^2$$

□

We incorporate the above formula in the following algorithm. For a given vertex v,

$$score_G(v) = 4 \sum_{u \in V(G)} d(u,v)^2 - 2d(v)^2$$

then $\sum_{u \neq v} d(u,v)^2$ can be computed by taking the characteristic vector χ_v of $N(v)$ (a bit vector of length n where the $\chi_v[i] = 1 \leftrightarrow (v, v_i) \in E$). Then for each vertex $u \in V \setminus \{v\}$ we go through each neighbor x of u in its adjacency list and check if $\chi_v[x] = 1$ to increment $d(v,u)$.

It takes $O(m)$ time to compute the $score_G(x)$ for a vertex x, to get the vertex with maximum score it takes $O(nm)$ time. Note that after removing k vertices (for constant k) the graph still has $O(m)$ edges. Now we give an efficient approximation to $score_G(x)$ that not only can be computed in linear time but also can be updated after removing a vertex y in time proportional to $d(x)$.

We have

$$\left(\frac{\sum\limits_{u \neq v} d(u,v)}{n}\right)^2 \leq \left(\frac{\sum\limits_{u \neq v} d(u,v)^2}{n}\right) \leq \left(\frac{(\sum\limits_{u \neq v} d(u,v))^2}{n}\right). \quad (5)$$

The first inequality is the Cauchy-Schwarz inequality, [c.f (Kreyszig 1978)], while the second follows from the fact that $d(u,v)$ is non-negative for each u, v.

In view of the above inequality, we approximate the $score_G(v)$ by $score'_G(v)$ given as

$$score'_G(v) = 2d_G^2(v) + 4\left(\sum_{u \neq v} d_G(u,v)\right)^2. \quad (6)$$

Our motivation to use $score'_G(x)$ is that not only it is easy to compute but also after a vertex is deleted it is easy to update the scores of all vertices in the remaining subgraph.

4.2 Algorithm

Now we give algorithm to compute the results. First we show procedure to find $score'_G(v)$ for graph G in Algorithm COMPUTE-SCORE(G) , then in Algorithm UPDATE-SCORE(G, v_i)we give algorithm to update the scores of vertices when a certain vertex v_i is deleted from the graph and finally we discuss the greedy approach to approximate that which vertices should be deleted in order to immunize the graph in Algorithm GREEDY-3(G,k)and along with these we discuss the time and space complexity in this section.

Now we give algorithm to compute $score'_G(v)$;

Algorithm 3 : COMPUTE-SCORE(G)

```
1: deg ← ZEROS(n)   ▷ Initialize the degree array to n zeros
2: codegSum ← ZEROS(n) ▷ Initialize the co-degree sum array to n zeros
3: score'_G ← ZEROS(n)   ▷ Initialize all scores to zeros
4: for each vertex v_i do
5:     for each neighbor v_j of v_i do
6:         deg_G[v_i] ← deg_G[v_i] + 1
7: for each vertex v_i do
8:     for each neighbor v_j of v_i do
9:         codegSum[v_j] ← codegSum[v_j] + deg_G[v_i] − 1
10: for each vertex v_i do
11:     score'_G[v_i] ← 2 * deg_G[v_i]^2 + 4 * codegSum[v_i]^2
```

Algorithm 4 : UPDATE-SCORE(G,v_i)

```
1: for each neighbor v_j of v_i do
2:     deg_G[v_j] ← deg_G[v_j] − 1
3:     codegSum[v_j] ← codegSum[v_j] − (deg_G[v_i] − 1)
4: deg_G[v_i] ← 0
5: codegSum[v_i] ← 0
6: score'_G[v_i] ← 0
7: for each neighbor v_j of v_i do
8:     score'_G[v_j] ← 2 * deg_G[v_j]^2 + 4 * codegSum[v_j]^2
```

Lemma 3. *runtime of Algorithm COMPUTE-SCORE(G) is $O(m)$.*

Proof. It is clear that line 6 of Algorithm COMPUTE-SCORE(G) takes $O(1)$ time and it is executed $O(m)$ times. Since loop at line 5 is iterated over all neighbors of a fixed vertex, v_i. Hence for v_i, line 6 is executed $d_G(v_i)$ times. Thus for all $v_i \in G$, line 6 runs for $\sum_{v_i \in V(G)} d_G(v_i) = 2m$ (Bondy & Murty 2008). Same is true for line 9.

Line 11 has constant time computation while it is computed for every vertex, thus it takes $O(n)$ time.

So total time taken to compute score of every vertex is $O(m)$. □

For a given vertex v of G, we update the score of vertices after removing v in the following way;

Lemma 4. *Algorithm UPDATE-SCORE(G,v_i) takes $O(d_G(v_i))$ time to update score with respect to parameter v_i. It takes $O(m)$ time to run Algorithm UPDATE-SCORE(G,v_i) for all $v_i \in V(G)$.*

Proof. In algorithm, line 2 and 8 takes constant time steps and both are computed $d_G(v_i)$ times. Hence for a fixed vertex v_i runtime of algorithm is $O(d_G(v_i))$. Now as mentioned in lemma 3, sum of degrees of vertices is $2m$, so for all $v_i \in V(G)$ it takes only $O(m)$ time to update score in $G_{-\{v_i\}}$. □

We here give the proof of correctness of the Algorithms COMPUTE-SCORE(G) and UPDATE-SCORE(G,v_i).

Lemma 5. *1. For each vertex $v \in V(G)$ Algorithm COMPUTE-SCORE(G) computes the $score'_G(v)$ as defined in (6).*

2. For all vertices $u, v \in V(G)$ Algorithm UPDATE-SCORE(G,v_i) computes the $score'_{G-u}(v)$ as defined in (6).

Proof. 1. It is clear that the Algorithm COMPUTE-SCORE(G) computes the first term correctly, as each neighbor v_i of v contributes 1 to the degree of v. To see why the second term is computed correctly, note that since $v_i \in N(v)$, v_i appears in $N(v,x)$ for each neighbor $x \neq v$ of v_i exactly once. Hence iterating over all v_i correctly computes $\sum\limits_{u \neq v} d_G(u,v)$.

2. This statement follows from the argument in the above part, since contribution of any vertex u to the first term of $score'(v)$ in (6) is exactly 1 if

and only if $v \in N(u)$. While that to the second term as argued above is exactly equal to $d(u)-1$. □

Here is the algorithm to select k vertices in the given graph G such that approximated eigendrop is maximized after deleting those vertices.

Algorithm 5 : GREEDY-3(G,k)

1: $S \leftarrow \emptyset$
2: COMPUTE-SCORE(G)
3: **while** $|S| < k$ **do**
4: $\quad v \leftarrow \arg\max_{u \in V \setminus S} score'_G[u]$
5: $\quad S \leftarrow S \cup \{v\}$
6: $\quad$ UPDATE-SCORE(G,v)
7: **return** S

Theorem 2. *The computational complexity of the Algorithm GREEDY-3(G,k) is $O(m + n + k \log n)$, while the space complexity is $O(m + n + k)$.*

Proof. By Lemma 3 line 2 takes $O(n + m)$ time. The scores for all vertices can be stored in a MAX-HEAP, which can be built in $O(n)$ time while each UPDATE-KEY and EXTRACT-MAX takes $O(\log n)$ time (Thomas H.. Cormen & Stein 2001). We extract max from the the heap k times, hence its total runtime is $O(\log n)$. As argued by Lemma 4, total time consumed in all calls to UPDATE-SCORE takes $O(m)$ time.

For the space complexity, in addition to storing the graph that takes $O(n + m)$ space, we need $O(n)$ space to store the three additional arrays. □

5 Experiments

We implemented our proposed algorithm in Matlab and our implementation along with source code and documentation is available online on the given link [1]. For benchmark comparison we also implemented MAX-DEGREE and UPDATED-MAX-DEGREE. MAX-DEGREE picks the top k maximum degree vertices - UPDATED-MAX-DEGREE selects the vertex with the maximum degree and deletes that vertex and repeats k times. We use the NET-SHIELD implementation which is available online[2].

Name	Nodes #	Edges #
Karate	34	78
Oregon	10,670	22,002
AA	418,236	2,753,798

Table 1: Summary of Datasets

The data sets used in experimentation are described in Table 1. The first data set is of a local karate club and is named as Karate graph[3]. Nodes of the graph represent members of the club and an edge between nodes show that corresponding members are friends with each other. Graph consists of 34 nodes and 78 edges. The graph is undirected and unweighted.

The second data set is from Oregon AS (Autonomous System)[4] router graphs, which are AS level connectivity networks inferred from Oregon route views. There are a number of Oregon AS graphs available and each node represents a router and an edge between two routers represents a direct peering relationship between two routers. We have selected one set from Oregon router graphs having 10,670 nodes and 22,002 edges. The graph is undirected and unweighted. Nodes selected by greedy algorithm are those routers whose immunization will maximally reduce the spread of virus.

The third data set (AA) is from DBLP[5] dataset. In graph a node represents an author and presence of an edge between two nodes shows that two authors have a co-authorship. In DBLP there is total node count of 418,236 and the number of edges among nodes is 2,753,798. We extracted smaller graphs by selecting co-authorship graph of only one journal (e.g Displays, International Journal of Computational Intelligence and Applications, International Journal of Internet and Enterprise Management, etc.). We ran our experiments on 20 different smaller co-authorship networks based on co-authorship graphs of 20 different journals. For the smaller sub graphs that we have extracted from DBLP dataset, node count goes up to few thousands and edge count goes up to few ten thousands. Detail of sub graphs of DBLP data set is given in Table 2. These subgraphs are also undirected and unweighted.

Name	Nodes	Edges
AI Communication	1,203	2,204
APJOR	1,132	1,145
Computer In Industry	2,844	4,466
Computing And Informatics	1,598	2,324
Display	1,374	3,204
Ecological Informatics	1,990	4,913
Engineering Application of AI	4,164	6,733
IJCIA	848	975

Table 2: Summary of DBLP subgraphs

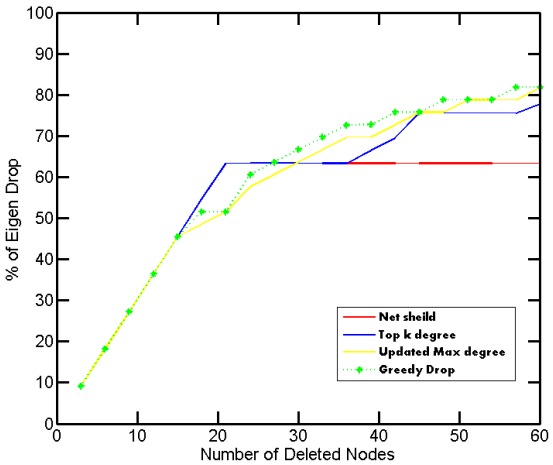


Figure 1: eigendrop of AI Communication Graph

We performed the experiments using varying value of k. In the results shown, x-axis shows the value of k which is the deletion count of vertices and y-axis shows the percentage of eigendrop which is the achieved benefit after immunizing k nodes in graph. Results are evaluated on the basis of percentage of eigendrop. Eigendrop is difference of largest eigenvalues of original graph and perturbed version of graph

[1] www.dropbox.com
[2] https: //www.dropbox.com/s/aaq5ly4mcxhijmg/Netshieldplus.tar.
[3] http://konect.uni-koblenz.de/networks/ucidata-zachary
[4] http://snap.stanford.edu/data/oregon1.html
[5] http://dblp.uni-trier.de/xml/

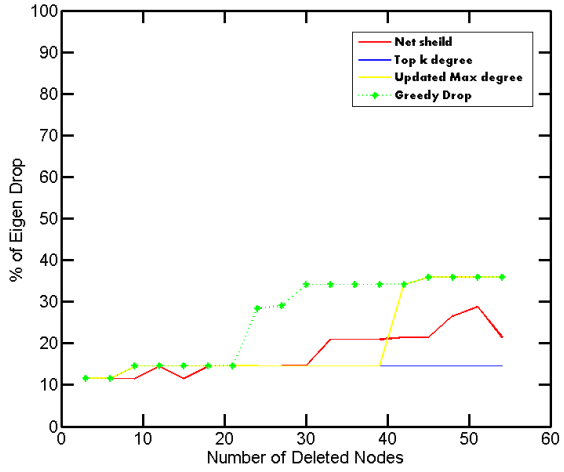


Figure 2: eigendrop of APJOR Graph

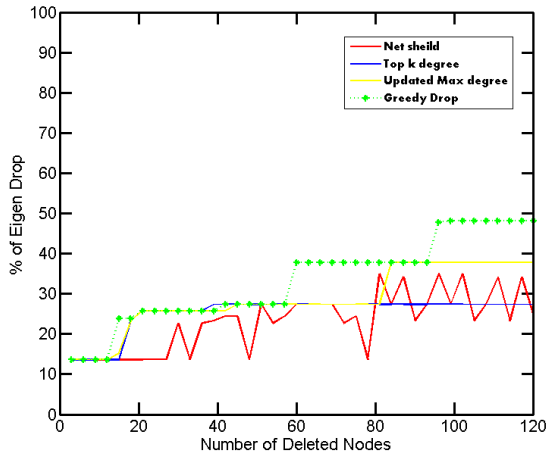


Figure 3: eigendrop of Computer In Industry Graph

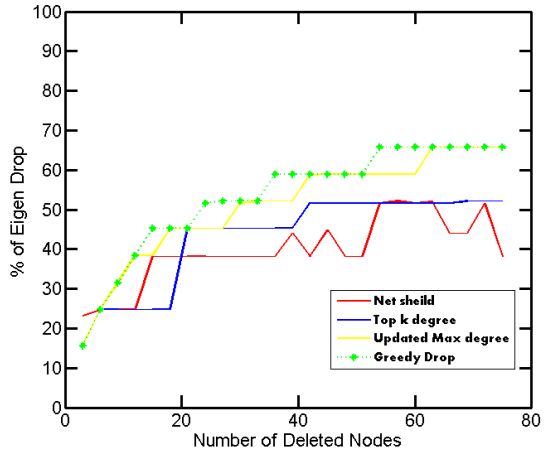


Figure 4: eigendrop of Computing And Informatics Graph

after immunization of k nodes.

$$\Delta\lambda = \lambda - \lambda(S) \tag{7}$$

where S is the set containing nodes to be immunized having cardinality k. It is clear from the results

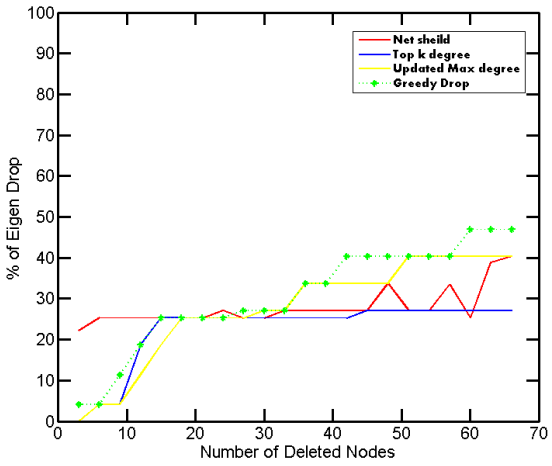


Figure 5: eigendrop of Displays Graph

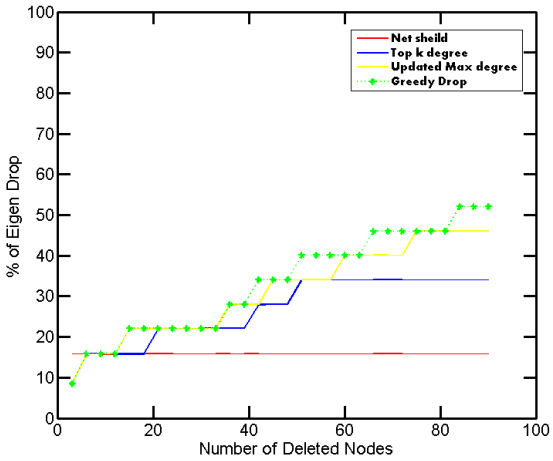


Figure 6: eigendrop of Ecological Informatics Graph

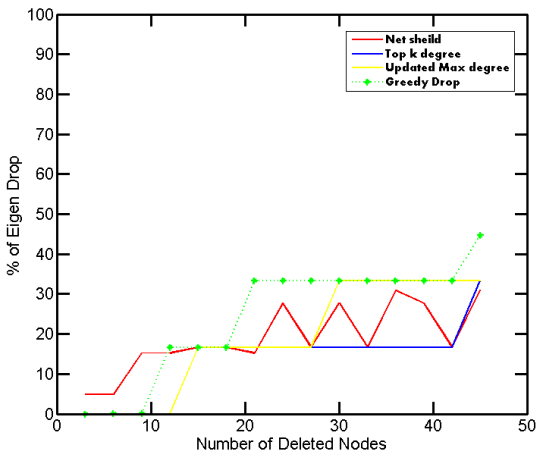


Figure 7: eigendrop of IJCIA Graph

that our greedy algorithm outperforms NetSheild approach and other approaches like top k degree and updated maximum degree approach. Our greedy algorithm is scalable for larger values of k as well as for larger graphs as our algorithm has less running time complexity than NetSheild, top k degree and updated maximum degree.

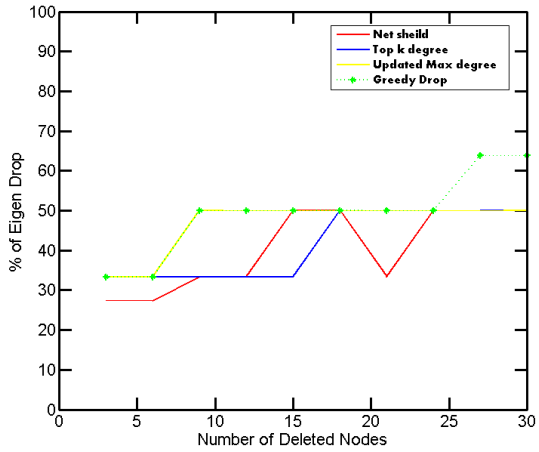


Figure 8: eigendrop of IJIEM Graph

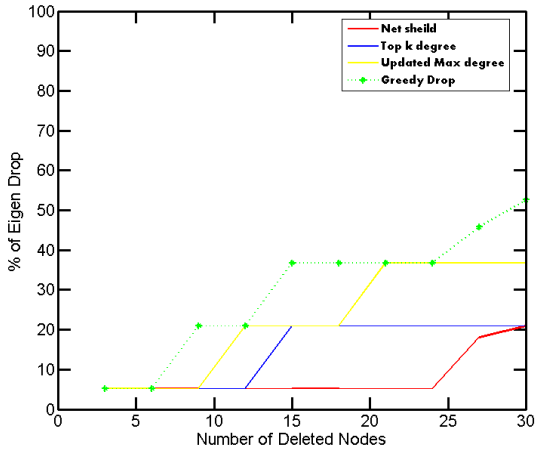


Figure 9: eigendrop of MASA Graph

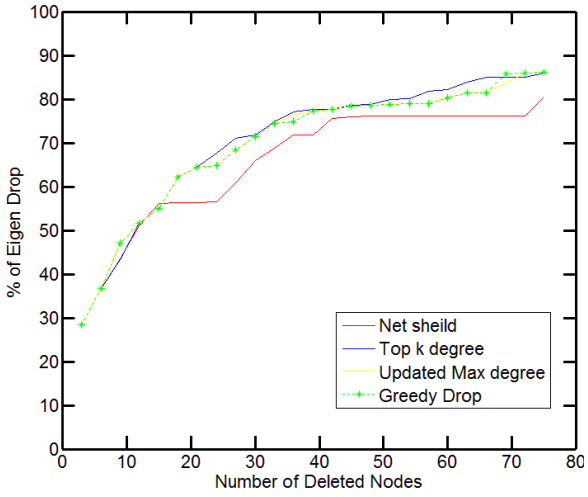


Figure 10: EigenDrop of Oregon Graph

6 Conclusion

In this work, we explored some links between established graph vulnerability measure and other spectral properties of even powers of adjacency matrix of the graph. We define shield value in terms of trace of the adjacency matrix of the graph. Based on these insights we present a greedy algorithm that iteratively selects k nodes such that the impact of each node is maximum in the graph, in the respective iteration, and thus we maximally reduce the spread of a potential infection in the graph by removing those vertices. Our algorithm is scalable to large graphs since it has linear running time in the size of the graph.

We have conducted experiments on different real world communication graphs to confirm the accuracy and efficiency of our algorithm. Our algorithm outperforms the state of the art algorithm in performance as well as in quality.

For the future work, we consider the generalized even parameter k, used for our shield value. Hence, we will be focusing on effectively improving the quality of the estimates. Techniques like locality sensitive hashing can be incorporated for the efficient approximation of the shield value for general k.

References

Bienstock, D. & Seymour, P. (1991), 'Monotonicity in graph searching', *Journal of Algorithms* **12**(2), 239–245.

Bondy, J. & Murty, U. (2008), *Graph theory*, Springer.

Briesemeister, L., L. P. & Porras, P. (2003), Epidemic profiles and defense of scale-free networks, *in* 'Proceedings of the 2003 ACM workshop on Rapid malcode', ACM, pp. 67–75.

Chakrabarti, D., W. Y. W.-C. L. J. & Faloutsos, C. (2008), 'Epidemic thresholds in real networks', *ACM Transactions on Information and System Security (TISSEC)* **10**(4), 1.

Chen, C., T. H. P.-B. T. C. E.-R. T. F. C. & Chau, D. (2016), 'Node immunization on large graphs: Theory and algorithms', *IEEE Transactions on Knowledge and Data Engineering* **28**(1), 113–126.

Chung, F. (1997), *Spectral graph theory*, American Mathematical Society.

Daadaa, Y., J. A. & Shabbir, M. (2016), 'Network decontamination with a single agent', *Graphs and Combinatorics* **32**(2), 559–581.

Erdös, D., I. V. L.-A. T. E. & Bestavros, A. (2012), 'The filter-placement problem and its application to minimizing information multiplicity', *Proceedings of the VLDB Endowment* **5**(5), 418–429.

Flocchini, P., H. M. & Luccio, F. (2007), 'Decontaminating chordal rings and tori using mobile agents', *Int. J. Found. Comput. Sci.* **18**(03), 547–563.

Flocchini, P., H. M. & Luccio, F. (2008), 'Decontamination of hypercubes by mobile agents', *Networks* **52**(3), 167–178.

Fraigniaud, P. & Nisse, N. (2008), 'Monotony properties of connected visible graph searching', *Information and Computation* **206**(12), 1383–1393.

Ganesh, A., M. L. & Towsley, D. (2005), The effect of network topology on the spread of epidemics, *in* 'Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.', Vol. 2, IEEE, pp. 1455–1466.

Kempe, D., K. J. & Tardos, É. (2003), Maximizing the spread of influence through a social network, *in* 'Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 137–146.

Kreyszig, E. (1978), *Introductory functional analysis with applications*, Wiley.

Kuhlman, C.J., T. G. S.-S. M. M. & Ravi, S. (2013), Blocking simple and complex contagion by edge removal, *in* '2013 IEEE 13th International Conference on Data Mining', IEEE, pp. 399–408.

Nemhauser, G.L., W. L. & Fisher, M. (1978), 'An analysis of the approximations for maximizing submodular set functions', *Mathematical Programming* **14**, 265–294.

Prakash, B.A., V. J. & Faloutsos, C. (2012), Spotting culprits in epidemics: How many and which ones?, *in* '2012 IEEE 12th International Conference on Data Mining', IEEE, pp. 11–20.

Seeman, L. & Singer, Y. (2013), Adaptive seeding in social networks, *in* 'Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on', IEEE, pp. 459–468.

Serre, D. (2002), *Matrices*, Springer.

Song, C., H. W. & Lee, M. (2015), Node immunization over infectious period, *in* 'Proceedings of the 24th ACM International on Conference on Information and Knowledge Management', ACM, pp. 831–840.

Strang, G. (1988), 'Linear algebra and its applications brooks', *Cole Thomson Learning Inc* .

Thomas H.. Cormen, Leiserson, C.-R. R. & Stein, C. (2001), *Introduction to algorithms*, Vol. 6, MIT press Cambridge.

Tong, H., P. B. E.-R. T. F. M. & Faloutsos, C. (2012), Gelling, and melting, large graphs by edge manipulation, *in* 'Proceedings of the 21st ACM international conference on Information and knowledge management', ACM, pp. 245–254.

West, D. (2001), *Introduction to graph theory*, Prentice Hall.

Zhang, Y. & Prakash, B. (2014*a*), Dava: Distributing vaccines over networks under prior information., *in* 'SDM', SIAM, pp. 46–54.

Zhang, Y. & Prakash, B. (2014*b*), Scalable vaccine distribution in large graphs given uncertain data, *in* 'Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management', ACM, pp. 1719–1728.

Identification of Interesting Rules by Pruning Redundant Specialisations and Generalisations

Henry Petersen[1] **Josiah Poon**[1] **Simon Poon**[1] **Clement Loy**[2]

[1] School of Information Technologies
University of Sydney
NSW, Australia
Email: {h.petersen, josiah.poon, simon.poon}@sydney.edu.au

[2] School of Public Health
University of Sydney
NSW, Australia
Email: clement.loy@sydney.edu.au

Abstract

Association rule mining algorithms can generate potentially useful rules from raw data, but it is well known that they can create an impractically large number of associations. Generating too many associations presents a barrier to user analysis and interpretation. This paper proposes a novel approach for identifying the most interesting associations based on excluding redundant rules. We refer to this approach as robust redundancy. Given an interesting rule, most comparable approaches consider additional rules to be redundant when terms are added without increasing some measure of rule quality. We note that such approaches can be overly aggressive, and complex interactions between variables can lead to interesting rules being incorrectly excluded.

Our approach also removes associations that are always worse than those rules with additional terms. This avoids producing general rules that are simply artefacts of interesting specialisations.

Our proposed approach is shown to identify interesting associations missed by other approaches on multiple datasets (including standard data). Additional uninteresting associations are also pruned by our algorithm. The total number of rules identified is not only often lower overall, but average rule quality is generally improved or unaffected.
Keywords Association Rules, Redundancy, Data Mining

1 Introduction

Association rule mining is an important task in knowledge extraction and data analysis. Formally, let $\mathcal{A} = \{a_1, a_2, \ldots, a_M\}$ be a set of M attributes. We then define N data $\mathcal{D} = \{d_1, d_2, \ldots, d_N\}$ such that $d_i \subseteq \mathcal{A} \forall d_i \in \mathcal{D}$. The association rule mining task seeks to find all *interesting* rules of the form $X \Rightarrow Y$, where X and Y are disjoint subsets of $\mathcal{A}$.

It is well known that the number of rules generated can be so large as to obscure their interpretation, presenting a barrier to practical use [18]. Ideally, we wish to find only the most interesting rules. This can be broken into two tasks; identification of truly interesting rules, and the removal of those which are simply redundant artefacts of other rules. Existing approaches for both are introduced in section 2.

Within the literature, there are several subproblems that have been studied. Examples include rules with fixed [14], or single attribute [6] consequents, numerical data [12], or negative associations [6, 8] (e.g. rules of the form $X \Rightarrow \neg Z$). In this paper we focus on the problem of positive rules from binary data with single attribute consequents.

In this paper we refer to the concepts of rule *generalisations* and *specialisations*. Rule $Y \Rightarrow Z$ is a *generalisation* of $X \Rightarrow Z$ if Y is a proper subset of X. Similarly, rule $Y \Rightarrow Z$ is a *specialisation* of $X \Rightarrow Z$ if Y is a proper superset of X.

When comparing two rules $X \Rightarrow Z$ and $XQ \Rightarrow Z$ to evaluate redundancy, existing approaches ignore data containing only part of the antecedent. As a result, non-interesting rules can be retained, and potentially interesting rules discarded. This paper proposes an alternate definition of redundancy (which we call robust redundancy) that utilises such information to improve the quality of the discovered rules. An algorithm for generating rules with robust redundancy is also proposed and evaluated.

2 Background

A key problem for association rule mining is measuring how interesting a rule is. Traditionally this is done using *support* and *confidence* (analogous to the sample probability of a rule and the conditional probability of Y given X respectively). Many alternate approaches for measuring interestingness have been proposed [4, 11], several of which are presented in Table 1. The interested reader is directed to the literature for further information [13].

The size of the search space is a major concern when mining association rules. Prior work often employs heuristics such as maximum rule lengths, fixed consequents, or frequency thresholds in order to control this [5]. To our knowledge, only Hämäläinen's Kingfisher algorithm is able to identify all significant, non-redundant rules. We extend the Kingfisher approach for rule generation in section 3.

It is well established that the number of rules identified can often be so large as to hamper their interpretation [1]. The concept of *redundancy* can be used to control the number of rules. Consider a hypothetical study of supermarket transactions which identifies that people who buy a soft drink will also buy chips.

Measure	Formula
Support [2]	$\frac{m(X)}{M}$
Confidence [2]	$\frac{m(XY)}{m(Y)}$
Interest [4]	$\frac{M \times m(XY)}{m(X)m(Y)}$
Leverage [11]	$\frac{m(XY)}{M} - \frac{m(X)m(Y)}{M^2}$
χ^2	$\frac{M^5 \times Leverage(X \Rightarrow Y)^2}{m(X)m(\neg X)m(Y)m(\neg Y)}$
Fisher's P	$\sum_{i=0}^{min(m(X\neg Y),m(Y\neg X))} \frac{\binom{m(X)}{m(XY)+i}\binom{x(\neg X)}{m(\neg X \neg Y)+i}}{\binom{M}{m(Y)}}$

Table 1: Several common interestingness measures for a rule $X \Rightarrow Y$ expressed in terms of partial frequency counts.

Further analysis may also identify that people who buy soft drink on Tuesday will buy chips. However the condition that it be Tuesday does not improve the quality of the association. The rule likely exists because the association between people buying soft drink and chips holds regardless of whether it is Tuesday or not. The association between people buying soft drink on Tuesday and buying chips is *redundant*.

Several authors have proposed formal means for defining redundant rules [1, 18, 3, 15, 16]. A definition of redundancy suitable for use with a general goodness measure (assuming single attribute consequents) was first proposed in 2010 by Hämäläinen [5]. Hämäläinen defines a rule R to be redundant if some more general rule (i.e. a rule whose antecedent is a subset of the antecedent of R) has equal or better utility with respect to some goodness measure. We repeat this more formally in definition 1.

Definition 1 *CLASSICAL REDUNDANCY*

Consider two rules $X \Rightarrow Z$ and $XQ \Rightarrow Z$ where X and Q are disjoint sets of items, and Z is a single item of value a. Let $M(\cdot)$ be an increasing measure of rule interestingness. Rule $XQ \Rightarrow Z$ is redundant with respect to rule $X \Rightarrow Z$ if $M(XQ \Rightarrow Z) \leq M(X \Rightarrow Z)$.

We note that when comparing rules based on some arbitrary goodness measure, complications can arise due to complex interactions between constituent attributes. That such interactions can give rise to spurious associations has been studied [10, 17], however less attention has been paid to how it might obscure useful relationships.

Models of redundancy that remove spurious generalisations is a problem that has seen limited interest within the community. Removing spurious generalisations was first looked at in 2001 by Liu et al. [9] in their work on non-actionable rules. For a rule r_0 and the set of its decedents $R = \{r_1, r_2, \cdots, r_N\}$, they define a r_0 to be *non-actionable* if it is not interesting over the domain where instances matching at least one antecedent in R are removed. Essentially, they claim a rule must cover some unique set of instances (with respect to the set of its specialisations) in which the relationship described still holds. A rule has no utility with respect to the set of its specialisations if it does not cover such a set of instances.

A similar concept to non-actionable rules was proposed by Webb in his work on self sufficient itemsets [17]. This work builds upon the concept of an *exclusive domain* for a given itemset. Formally, given an itemset s and its specialisations S, the exclusive domain of s is defined to the domain of s minus the union of the domains of all itemsets in S. After generalising the concepts of redundancy and productivity [15, 16] for use with itemsets, an itemset is defined to be self sufficient if it is productive and non-redundant both with respect to the entire data and its exclusive domain.

In many respects, self sufficient itemsets can be considered an extension of non-actionable rules for use in an itemset context. However, it is noteworthy that itemsets must also be productive and non-redundant. This pruning of both specialised and general itemsets is similar to our work with robust redundancy described in this paper, although it is performed in an itemset context. We also examine how to avoid pruning specialisations where redundancy is likely to be an artefact of interactions between constituent attributes.

3 Robust Redundancy

A rule is considered redundant when it adds no information over another rule. We claim that classical redundancy makes such a comparison using incomplete information.

Depending on the interestingness measure $M(\cdot)$ in use, $M(X \Rightarrow Z)$ is computed using the frequencies XZ, $\neg XZ$, $X\neg Z$, and $\neg X\neg Z$. Note that directly comparing rules $M(X \Rightarrow Z)$ and $M(XQ \Rightarrow Z)$ does not consider transactions including only part of the rule antecedent (i.e. frequencies of $X\neg QZ$, $\neg XQZ$, $X\neg Q\neg Z$, and $\neg XQ\neg Z$).

Association rule mining can be confounded by noise and complex relationships between variables. Potential lack of control over the data collection process can further complicate matters. Such noise could artificially raise or lower the measured interestingness value of a rule, which could lead to interesting rules being incorrectly excluded.

We propose using additional information in an attempt to avoid excluding interesting rules. We also seek to identify seemingly interesting rules that are simply artefacts of groups of their specialisations. We refer to these approaches as specialisation and generalisation redundancy respectively.

3.1 Specialisation Redundancy

We propose an alternate approach to redundancy in definition 2. We augment the classical approach given in definition 1 by not eliminating a rule $XQ \Rightarrow Z$ if the partial frequencies can be used to demonstrate the attributes in Q add value. This is accomplished by computing the strength of the association between X and Z conditioned on Q, and comparing it against the strength of the marginal association. If the conditional association between X and Z improves over the strength of the previous association, we obtain evidence that the addition of Q adds value to the existing rule.

Definition 2 *SPECIALISATION REDUNDANCY*

Consider two rules $X \Rightarrow Z$ and $XQ \Rightarrow Z$ where X and Q are disjoint sets of attributes, and Z is a single attribute. Let $M(\cdot)$ be an increasing measure of rule interestingness. Rule $XQ \Rightarrow Z$ is specialisation redundant with respect to rule $X \Rightarrow Z$ if $M(XQ \Rightarrow Z) \leq M(X \Rightarrow Z)$, and $M(X \Rightarrow Z|Q) \leq M(X \Rightarrow Z)$.

Computing the conditional association requires frequencies for $\neg XQZ$ and $\neg XQ\neg Z$ (in addition to the frequencies XQZ and $XQ\neg Z$). We do not consider the association between X and Z conditioned on $\neg Q$, as the rule we are seeking to obtain evidence for is $XQ \Rightarrow Z$, which contains Q.

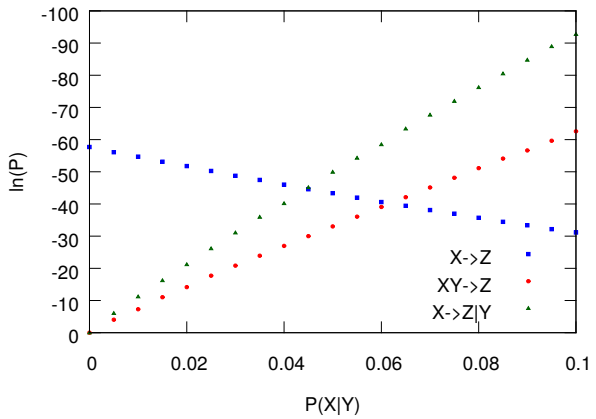


Figure 1: ln(P-values) for rules $X \Rightarrow Z$, $Y \Rightarrow Z$, and $XY \Rightarrow Z$ versus conditional probability of X and Y.

$P(X)$	0.3
$P(Y)$	0.3
$P(Z\|X,Y)$	0.8
$P(Z\|X,\neg Y)$	0.4
$P(Z\|\neg X,Y)$	0.4
$P(Z\|\neg X,\neg Y)$	0.6

(a) Specialisation

$P(X)$	0.5
$P(Y)$	0.5
$P(Z\|X,Y)$	0.5
$P(Z\|X,\neg Y)$	0.1
$P(Z\|\neg X,Y)$	0.1
$P(Z\|\neg X,\neg Y)$	0.1

(b) Generalisation

Table 2: Marginal and conditional probabilities for several combinations of variables used in the motivating example for robust generalisation redundancy.

3.1.1 Example

Consider hypothetical data sampling 3 binary variables X, Y, and Z. Assume 1000 data points with the probabilities expressed in Table 2a. From these probabilities, observe that a strong dependency exists between Z and the itemset XY. We now examine the rules $X \Rightarrow Z$, $Y \Rightarrow Z$, and $XY \Rightarrow Z$ as we vary the joint probability of variables X and Y.

Figure 1 plots rule quality against the conditional probability of X given Y. For larger conditional probabilities we can observe that the quality of the rule $XY \Rightarrow Z$ is superior to that of rule $X \Rightarrow Z$. As the overlap between data containing X and data containing Y decreases, the quality of the more general rule $X \Rightarrow Z$ surpasses its specialisation. Consequently the rule $XY \Rightarrow Z$ is removed as redundant, obscuring the true underlying structure of the data.

Holding the marginal probabilities constant, the frequencies of both X and Y decrease along with the conditional probability of X given Y. In order to support the rule $XY \Rightarrow Z$, we require data containing both (or neither) XY and Z. Hence, as the number of data with XY and Z decreases, so too does the evidence available to evaluate it. That the general rule $X \Rightarrow Z$ surpasses the true rule $XY \Rightarrow Z$ in quality as the conditional probability decreases is a reflection of this.

By comparing rules $X \Rightarrow Z$ and $XY \Rightarrow Z$ using robust redundancy (i.e. including the conditional dependencies) we make more effective use of available data to evaluate the rules. In the example given in Figure 1, rule $XY \Rightarrow Z$ is retained as non-redundant for conditional probabilities greater than $\sim$0.045. This is in contrast to classical redundancy, where the threshold for retaining $XY \Rightarrow Z$ is $\sim$0.062. Although in both cases the conditional probability of X given Y eventually reaches a point where insufficient evidence for the specialised rule exists, the range of values for which robust redundancy can still retain $XY \Rightarrow Z$ is increased.

3.2 Generalisation Redundancy

It is possible for general rules to exist that only appear interesting due to the presence of one or more interesting specialisations. Definition 3 outlines a concept we call *Robust Generalisation Redundancy*. A rule $X \Rightarrow Z$ is generalisation redundant if for all non-redundant specialisations $XQ \Rightarrow Z$, the rule $X \Rightarrow Z|\neg Q$ is uninteresting (its goodness is less than the required threshold).

If a rule $X\neg Q \Rightarrow Z$ is interesting, we obtain evidence that the generalised rule is interesting even in the absence of the terms in Q. If after identifying all other interesting rules we cannot find evidence that $X \Rightarrow Z$ is interesting in the absence of the additional terms in its specialisations, we consider it redundant. Computing the conditional association on $\neg Q$ uses the frequencies $X\neg QZ$ and $X\neg Q\neg Z$. Hence, by applying both specialisation and generalisation redundancy we consider all frequencies in the sample data.

Definition 3 *GENERALISATION REDUNDANCY*

Consider a rule $X \Rightarrow Z$ and the complete set of its non-redundant specialisations $\mathcal{R}$. Let $M(\cdot)$ be an increasing measure of rule interestingness, and α be the corresponding goodness threshold. Rule $X \Rightarrow Z$ is generalisation redundant with respect to $\mathcal{R}$ if $M(X \Rightarrow Z|\neg Q) \leq \alpha$ for all rules $XQ \Rightarrow Z$ in $\mathcal{R}$.

3.2.1 Example

Consider hypothetical data containing drug prescriptions and a corresponding binary patient outcome. Assume there are two drugs (X and Y) which work in combination to produce a positive outcome. Neither drug will produce a positive outcome on its own (a baseline probability for positive outcome of 0.1 is used). The exact probabilities used can be found in Table 2b.

When the conditional probability of X given Y is 1, the measured quality of the rules $X \Rightarrow Z$, $Y \Rightarrow Z$, and $XY \Rightarrow Z$ will be identical and maximal. The quality of these rules decreases with this conditional probability, with the quality of the general rules decreasing at the greatest rate. However despite the underlying structure of the data indicating that neither X or Y alone support a positive outcome, the strength of these associations will likely remain quite high.

When comparing rules $X \Rightarrow Z$ and $XY \Rightarrow Z$, examining the rule $X \Rightarrow Z|\neg Y$ (i.e. conditioned on the absence of the additional terms Y) indicates that there is no evidence to support the rule $X \Rightarrow Z$ without also including the features Y. As $XY \Rightarrow Z$ is the only identified specialisation of $X \Rightarrow Z$, and we have no evidence to indicate $X \Rightarrow Z$ is valid without the additional features, we consider it redundant.

Finally, we acknowledge that such an approach could potentially over-fit and remove valid general rules. We address this concern in the following section on redundancy chaining.

3.3 Redundancy Chaining

Classical redundancy as defined in definition 1 is transitive. If a rule $XQY \Rightarrow Z$ is redundant with respect to a generalisation $XQ \Rightarrow Z$, and $XQ \Rightarrow Z$ is redundant with respect to $X \Rightarrow Z$, then $XQY \Rightarrow Z$ will

Attr.	fr	Rule	ln(P)
ABCD	10	$A \Rightarrow D$	-19.33
ABD	10	$AB \Rightarrow D$	-8.10
ACD	10	$ABC \Rightarrow D$	-3.78
AD	10	$A \Rightarrow D\|B$	-18.75
BC	30	$A \Rightarrow D\|BC$	-20.56
BD	10	$AB \Rightarrow D\|C$	-7.83
CD	10		
D	10		

Table 3: Sample data and rules for lemma 1.

Attr.	fr	Rule	Conf
ABCD	60	$A \Rightarrow D$	0.90
AB	20	$AB \Rightarrow D$	0.75
ACD	10	$ABC \Rightarrow D$	1.00
AD	10	$A \Rightarrow D\|\neg B$	1.00
		$A \Rightarrow D\|\neg(BC)$	0.50
		$AB \Rightarrow D\|\neg C$	0.00

Table 4: Sample data and rules for lemma 2.

be redundant with respect to $X \Rightarrow Z$. This result is straightforward to prove.

Unfortunately the same relation does not hold for the proposed robust redundancy. A proof that specialisation redundancy is nontransitive is given in lemma 1. That specialisation redundancy is nontransitive creates an interesting possibility. Assume a rule r exists that is specialisation redundant with respect to one or more generalisations $r_0, \ldots r_i$. Let $r_0, \ldots r_i$ be redundant with respect to rules $r_{i+1}, \ldots r_n$. Despite being a redundant specialisation of other rules, r is non-redundant with respect to all non-redundant generalisations. We take the view that in such a situation, the rule r should be considered non-redundant.

Lemma 1 *Specialisation redundancy is nontransitive.*

Let $A \Rightarrow D$, $AB \Rightarrow D$, and $ABC \Rightarrow D$ be three rules generated from data in Table 3 using the log of Fishers P.

As the interestingness of the rules $AB \Rightarrow D$ and $A \Rightarrow D|B$ is worse than that of the rule $A \Rightarrow D$, $AB \Rightarrow D$ is redundant w.r.t. $A \Rightarrow D$.

As the interestingness of the rules $ABC \Rightarrow D$ and $AB \Rightarrow D|C$ is worse than that of the rule $AB \Rightarrow D$, $ABC \Rightarrow D$ is redundant w.r.t. $AB \Rightarrow D$.

As the interestingness of the rule $A \Rightarrow D|BC$ is better than that of the rule $A \Rightarrow D$, the rule $ABC \Rightarrow D$ is non-redundant w.r.t. $A \Rightarrow D$. ■

Lemma 2 *Using redundant rules when evaluating generalisation redundancy allows for additional rules to be included.*

Consider three rules $A \Rightarrow D$, $AB \Rightarrow D$, and $ABC \Rightarrow D$ generated from the data in Table 4 using the confidence measure with a threshold of 0.6.

According to definition 3 and the confidence scores for the above rules, $AB \Rightarrow D|\neg C$ is uninteresting so $AB \Rightarrow D$ is redundant w.r.t. $ABC \Rightarrow D$. When evaluating generalisation redundancy without redundant rules, the uninteresting rule $A \Rightarrow D|\neg(BC)$ implied $A \Rightarrow D$ is redundant. When evaluating generalisation redundancy with redundant rules, as $A \Rightarrow D|\neg B$ is interesting $A \Rightarrow D$ is non-redundant. ■

We also prove that whether or not redundant attributes are counted effects generalisation redundancy in lemma 2. As above, we elect not to allow redundant rules to influence the redundancy of another rule. In contrast to specialisation redundancy, this will lead to the exclusion of additional rules (comparing against additional rules raises the chance of inclusion as generalisation redundancy requires a rule be uninteresting with respect to ALL its specialisations).

Not using redundant rules to support retaining otherwise redundant generalisations can produce the following interesting situation. Assume a rule $r_1 : Y \Rightarrow A$ where $conf(Y \Rightarrow A) = 1$ and $supp(Y) = supp(A)$. Then for all rules of the form r_1 $X \Rightarrow A$ where $Y = XQ$ (i.e. generalisations of $Y \Rightarrow A$), the frequency of $X\neg QA$ will be 0, and the rule $X\neg Q \Rightarrow A$ will be uninteresting. By definition 3, r_1 is the only possible non-redundant rule with consequent A.

While it may in fact be desirable to keep such a rule, care must be taken to avoid confounding caused by the addition of independent attributes. We demonstrate how such confounding might occur by providing an extension of the above example. Consider the rule $YZ \Rightarrow A$ for some variable Z where $supp(ZA) = 1$. It is simple to see that $conf(YZ \Rightarrow A) = 1$, $supp(YZ) = supp(A)$, and $freq(Y\neg ZA) = 0$. The rule $Y \Rightarrow A$ will be considered redundant.

By Occam's Razor we prefer a more general rule over its specialisations unless evidence can be obtained to suggest otherwise. Using only generalisation redundancy can violate this principle as no evidence is ever considered to support $YZ \Rightarrow A$ over $Y \Rightarrow A$. In the worst case, for a given consequent only one highly specific rule will be selected with all others being made redundant. Therefore specialisation (or classical) redundancy should usually be employed before generalisation redundancy. We note however that in some cases (such as those where we prefer to generate more specific rules), generalisation redundancy may be applied first.

4 Rule Mining Algorithm

Pseudocode for our algorithm (an extension of the Kingfisher algorithm [5, 6]) is given in Algorithm 1. We find non-redundant rules using the natural log of Fisher's P value (a decreasing measure) in a three stage process:

1. All potentially non-redundant rules with some minimum log P-value are identified.
2. Rules identified in stage 1 are examined and specialisation redundant rules are pruned.
3. Remaining rules are examined and generalisation redundant rules are pruned.

Stage 1 is a BFS over itemsets, which is described in Algorithms 1 and 2. This begins by creating level 1 nodes for each individual attribute. Any attribute whose frequency is too low to produce an interestingness value greater than α is removed at this stage.

For a level k node X corresponding to attributes $\{x_1, x_2, \ldots, x_k\}$, the P-values of the k rules $X\backslash\{x_i\} \Rightarrow x_i$ are computed and compared against the minimum threshold α. This process is described in Algorithm 3. The frequency of set X is calculated and stored, with P-values for rules being computed using frequencies of parent nodes. The number of iterations over the dataset is therefore limited to the number of nodes considered.

Each node maintains a length $|\mathcal{A}|$ bit vector of possible consequents (attributes A where the rule $XQ \Rightarrow A$ is possible). These vectors are initialised using bitwise and of vectors for parent nodes. As a node is processed, lower bounds on the log Fisher's P value

Algorithm 1: Search($\mathcal{D}$, $\mathcal{A}$, M, α)
Search algorithm for non-robust redundant association rules.

input : A set of data $\mathcal{D}$ over attributes $\mathcal{A}$, an increasing interestingness measure $M(\cdot)$, and a corresponding threshold α
output: A set of rules $\mathcal{R}$

```
   // Step 1: Find potentially interesting
      rules
 1 determine minf
 2 r ← Level1nodes(D,A,M,α,minf)
 3 l ← 2
 4 nls ← |r|
 5 while nls ≥ l do
 6     nls ← 0
 7     for i ← 1 to |A| do
 8         nls ← nls + Bfs(r.children[i],l,0)
 9     end
10     l ← l + 1
11 end
   // Steps 2 and 3: Prune redundant rules
12 R ← PruneSpecialisations(R)
13 R ← PruneGeneralisations(R, α)
14 return R
```

Algorithm 2: BFS(n, l, t)
BFS search for potentially non-redundant rules.

input : Root node n, target search level l, current level t
output: The number of level l nodes remaining in the tree

```
 1 nls ← 0
 2 if t = l − 2 then
 3     for i ← 1 to |n.children| − 1 do
 4         Y ← n.children[i].set
 5         for j ← i + 1 to |n.children| do
 6             Z ← n.children[j].set
 7             X ← Y ∪ Z
 8             Create new node child = Node(X)
 9             nls ← nls + 1
10             n.children[i].insert(child)
11             if Checknode(child) = false then
12                 delete child
13                 nls ← nls − 1
14                 for ∀ nodes v = Node(Y_m) where
                    X = Y_m A_m do
15                     v.possible[m] = false
16                 end
17             end
18             delete n.children[|n.children|]
19         end
20     end
21 else
22     for i ← 1 to |n.children| do
23         nls ← nls + Bfs(n.children[i],l,t + 1)
24     end
25 end
26 if |n.children| = 0 then
27     delete node n
28 end
29 return nls
```

are computed for all rules of the form $XQ \setminus \{A\} \Rightarrow A$ for all A in $\mathcal{A}$. If bounds for all attributes exceed the relevance threshold (the vector of possible consequents is 0), the node is pruned from the search. The bounds used were first reported by Hämäläinen [5], and are reproduced in Table 5. This process is described in Algorithm 4.

Each node X contains a vector with the best previous P-value for rules with consequent $x_i \in X$. Similar to the possible bit vector, these vectors are merged from parents when the node X is created (see line 8 of Algorithm 3). Using classical redundancy, if the bound on P-values for rules with a given consequent exceeds the corresponding value in this vector that consequent can also be considered impossible. Using robust redundancy (see definition 2), we also test that the bound on rule $XQ \Rightarrow A|Q$ is worse than the previous best value.

The Fishers P-value for rule $XQ \Rightarrow A|Q$ takes its smallest value when the number of instances containing sets $QA\neg X$ and $QX\neg A$ are 0 and QXA and $Q\neg X\neg A$ are as large as possible. This occurs when $freq(QXA) = freq(XA)$ and $freq(Q\neg X\neg A) = freq(\neg X\neg A)$. We therefore compute the bound for $XQ \Rightarrow A|Q$ using bnd3 from Table 5 with parameters $f(XA) = freq(XA)$, $f(X) = freq(XA)$, $f(A) = freq(XA)$, and $\mathcal{N} = freq(XA) + freq(\neg X\neg A)$.

The running time for stages 2 and 3 are quadratic in the number of rules tested (in general this is dwarfed by the initial search in stage 1). When comparing two rules $X \Rightarrow A$ and $XQ \Rightarrow A$, specialisation redundancy requires the computation of $M(X \Rightarrow A|Q)$, and generalisation redundancy requires $M(X \Rightarrow A|\neg Q)$. For Fisher's P, this requires us to obtain the frequencies for Q, $\neg Q$, AQ, and $A\neg Q$.

4.1 Pruning the Search Space

The Kingfisher algorithm employs two additional pruning steps to control the size of the search space. The first, referred to as the *lapis philosophorum* principle, deals with the case where all rules of the form $XQ \Rightarrow A$ become impossible at a given node $X\{A\}$. In such a case, A is also an impossible consequence for children of the parent node X, and its possible consequents vector can be updated. This principle is also applied in our approach.

The latter pruning step is pruning based on *minimality*. A rule $X \Rightarrow A$ is considered minimal iff $P(A|X) = 1$. For a given minimal rule $X \Rightarrow A$, any rule of the form $XQ \Rightarrow A$ or $XQA \Rightarrow B$ will be either classically redundant or not significant (Hämäläinen [6]).

Pruning based on minimality cannot be employed when searching for rules with robust redundancy. We now prove that with robust redundancy it is possible for a specialisation of a minimal rule to be both significant and non-redundant.

Lemma 3 *Given data $\mathcal{D}$, an increasing statistical goodness measure $M(\cdot)$, and rule $X \Rightarrow A$ such that $P(A|X) = 1$, $XQ \Rightarrow A$ may exist such that $M(X \Rightarrow A) < M(X \Rightarrow A|Q)$.*

$bnd1(\lvert A\rvert, \mathcal{N}) =$	$\frac{f(A)!f(\neg A)!}{\mathcal{N}!}$
$bnd2(\lvert X\rvert, \lvert A\rvert, \mathcal{N}) =$	$\frac{f(\neg X)!f(A)!}{\mathcal{N}!(f(A) - f(X))!}$
$bnd3(\lvert XA\rvert, \lvert X\rvert, \lvert A\rvert, \mathcal{N}) =$	$\frac{f(A)!f(\neg A)!(\mathcal{N} - f(XA))!}{\mathcal{N}!f(\neg A)!f(A\neg X)!}$

Table 5: Lower bounds for Fishers P. The function $f(\cdot)$ returns the frequency of its argument in $\mathcal{D}$

Algorithm 3: Checknode(v_X)
Generate rules from a node and check if it can have valid descendants.

input : Node v_X to check
output: Boolean value indicating whether or not children of the v_X can produce interesting, non-redundant rules.

```
1  for ∀ Y ⊂ X where |Y| = |X| − 1 do
2      Par_Y ← searchTree(Y)
3      if Par_Y not found then
4          return false
5      end
6      for i ← 1 to |A| do
7          v_X.possible[i] ←
             v_X.possible[i] & Par_Y.possible[i]
8          v_X.pbest[i] ←
             max(v_X.pbest[i],Par_Y.pbest[i])
9      end
10     if v_X.possible = ∅ then
11         return false
12     end
13 end
14 setfreq(X) = calcFreq(X)
15 for ∀ A_i ∈ A do
16     v_X.possible[i] ←
         v_X.possible[i] & possible(v_X,X,i)
17     if ((A_i ∈ X) and (v_X.possible[i])) then
18         val ← M(X\{A_i} ⇒ A_i)
19         if val ≤ α then
20             add rule X\{A_i} ⇒ A_i to R
21             v_X.pbest[i] ← max(val,v_X.pbest[i])
22         end
23     end
24 end
25 if v_X.possible = ∅ then
26     return false
27 end
28 return true
```

$X \Rightarrow A$ is minimal implies the frequency of set $X\neg A$ is 0. The frequencies of sets XA, $\neg XA$, and $\neg X\neg A$ are unknown.

Let Q be a set of attributes whose corresponding rows in $\mathcal{D}$ exactly match the sets XA and $\neg X\neg A$. $M(X \Rightarrow A)$ increases with each occurrence of XA and $\neg X\neg A$, and decreases with each occurrence of $\neg XA$. It is easy to observe that $freq(XA) = freq(XQA)$, $freq(\neg X\neg A) = freq(\neg XQ\neg A)$, and $freq(\neg XA) \geq freq(\neg XQA)$. Assuming $\mathcal{D}$ contains at least one occurrence of $\neg XA$, $M(X \Rightarrow A|Q)$ will therefore be greater than $M(X \Rightarrow A)$. ■

Lemma 4 *Given data $\mathcal{D}$, an increasing statistical goodness measure $M(\cdot)$, and a rule $X \Rightarrow A$ such that $P(A|X) = 1$, there may exist a rule $XQA \Rightarrow B$ such that $M(XA \Rightarrow B) < M(XA \Rightarrow B|Q)$.*

$X \Rightarrow A$ is minimal implies that the frequency of the set $X\neg A$ is 0. Hence $freq(XA) \geq freq(XQA)$. The frequencies of the sets XA, $\neg XA$, and $\neg X\neg A$ are unknown.

Let Q be a set of attributes whose rows in $\mathcal{D}$ exactly match the sets CB and $\neg C\neg B$ where $C = XA$. Observe that $freq(C) = freq(CQ)$, $freq(\neg C\neg B) = freq(\neg CQ\neg B)$, and $freq(\neg C) \geq freq(\neg CQ)$. Assuming $\mathcal{D}$ contains at least one occurrence of $\neg CB$, $M(C \Rightarrow B|Q)$ will therefore be greater than $M(C \Rightarrow B)$ (or $M(XA \Rightarrow B|Q) > M(XA \Rightarrow B)$). ■

Algorithm 4: Possible(v_X,X,A)
Check if rules generated from a node or its descendants with a given consequent can be non-redundant

input : Node v_X being checked, set X associated with that node, and consequent attribute A_j
output: True if rules with consequent A generated using v_X or its descendants can be interesting and non-redundant

```
1  if |X| < minf then
2      return false
3  end
4  if A ∉ X then
5      if |X| > |A| then
6          bnd = LB1(|A|, |D|)
7      else
8          bnd = LB2(|X|, |A|, |D|)
9      end
10 else
11     bnd ← LB3(|X|, |X\{A}|, |A|, |D|)
12     bndonq ←
         LB3(|X|, |X|, |X|, |D| − |X¬A| − |A¬X|)
13 end
14 if bnd > α then
15     return false
16 end
17 if A ∈ X then
18     if bnd ≥ v_X.pbest[j] and bndonq ≥
         v_X.pbest[j] then
19         return false
20     end
21 end
22 return true
```

5 Evaluation

Performance is evaluated with respect to three characteristics: total number of rules, overall rule quality, and efficiency. All experiments were run on a PC running Ubuntu Linux, with an Intel I7-4500 processor and 8gb RAM. Performance is also reported for rules generated with the classical definition of redundancy (definition 1), as well as a baseline with no redundancy based pruning.

5.1 Data

Our evaluation uses the following data covering several domains. Descriptive statistics are also given in Table 6.

- **Mushroom** Mushroom descriptions from the 1981 Audobon Society Field Guide to North American Mushrooms. This data is available in the UCI Machine Learning Repository [1].
- **T10I4D100K** An artificial dataset representing market basket data, obtained from the Frequent Itemset Mining Dataset Repository [2].
- **T40I10D100K** An artificial dataset representing market basket data, obtained from the Frequent Itemset Mining Dataset Repository [2].
- **Diabetes** Collection of real world data reporting traditional Chinese medical herbal prescriptions

[1] https://archive.ics.uci.edu/ml/datasets/Mushroom
[2] http://fimi.ua.ac.be/data/

Algorithm 5: PruneSpecialisations($\mathcal{R}$)
Prune redundant specialisations in $\mathcal{R}$

input : Set of rules $\mathcal{R}$
output: Set of non (specialisation) redundant rules $\mathcal{R}$

for $\forall$ A $\in \mathcal{A}$ **do**
  $\mathcal{R}_{\mathsf{A}} = \{\mathsf{R} \in \mathcal{R}$ s.t. $\mathsf{R} = \mathsf{X} \Rightarrow \mathsf{A}\}$
  Sort $\mathcal{R}_{\mathsf{A}}$ in increasing order on length of the antecedent
  for $i \leftarrow 1 \rightarrow |\mathcal{R}_{\mathsf{A}}| - 1$ **do**
    Let $\mathcal{R}_{\mathsf{A}}[i] = \mathsf{X} \Rightarrow \mathsf{A}$
    for $j \leftarrow i + 1 \rightarrow |\mathcal{R}_{\mathsf{A}}|$ **do**
      Let $\mathcal{R}_{\mathsf{A}}[j] = \mathsf{Y} \Rightarrow \mathsf{A}$
      $\mathsf{Q} = \mathsf{Y} \backslash \mathsf{X}$
      CondM $\leftarrow$ M(setFreq$[XQA]$, setFreq$[XQ]$, setFreq$[QA]$, setFreq$[Q]$)
      if X $\subset$ Y **then**
        if M(R_2) $\leq$ M(R_1) ***and*** CondM $\leq$ M(R_1) **then**
          delete R_2
        end
      end
    end
  end
end
return $\mathcal{R}$

Name	# Instances	# Attributes	Agv. Instances Length	Avg. Attribute Freq.
Aspergillosis	4377	101	15.93 ± 0.26	680.51 ± 66.05
Mushroom	8124	119	23.00 ± 0.00	1624.80 ± 358.73
Diabetes	1915	204	10.26 ± 0.11	105.21 ± 30.09
Fertility	766	215	15.73 ± 0.32	59.62 ± 14.21
Insomnia	460	112	13.48 ± 0.25	55.38 ± 11.10
T10I4D100K	100000	870	10.10 ± 0.02	1161.18 ± 74.73
T40I10D100K	100000	942	39.61 ± 0.05	4204.36 ± 249.57

Table 6: Summary of datasets used in the evaluation.

for diabetes. Includes both the herbs prescribed and a binary classification of the patient outcome as 'good' or 'bad'.

- **Fertility** Collection of real world data reporting traditional Chinese medical herbal prescriptions for fertility. Includes both the herbs prescribed and a binary classification of the patient outcome as 'good' or 'bad'.
- **Insomnia** Collection reporting traditional Chinese medical herbal prescriptions for insomnia. Includes both the herbs prescribed and a binary classification of the patient outcome as 'good' or 'bad'.
- **Aspergillosis** Text documents (titles and abstracts) for articles considered for inclusion in a systematic review on Aspergillosis [7]. Each document is converted to a binary vector indicating the presence or absence of each of 100 words, as well as a binary variable indicating whether the title and abstract was potentially relevant to the review. The words selected were those with the greatest discriminative power when identifying articles relevant to the review.

All values were obtained as the average of 10 independent experiments using a random 50/50 test/training split. Results are reported with their 95% confidence interval. Statistical significance tests are performed using a P-value of .05. In line with similar work, we measure interestingness using Fishers exact test (we report the natural log of P-values) [8, 6]. Thresholds for interesting rules were chosen to strike a balance between permissiveness and execution time, and differ between data and experiments.

Algorithm 6: PruneGeneralisations($\mathcal{R}$, α)
Prune redundant generalisations in $\mathcal{R}$

input : Set of rules $\mathcal{R}$ and an interestingness threshold α
output: Set of non (generalisation) redundant rules $\mathcal{R}$

for $\forall$ R $\in \mathcal{R}$ **do**
  Keep(R) $= false$
  HasSpec(R) $= false$
end
for $\forall$ A $\in \mathcal{A}$ **do**
  $\mathcal{R}_{\mathsf{A}} = \{\mathsf{R} \in \mathcal{R}$ s.t. $\mathsf{R} = \mathsf{X} \Rightarrow \mathsf{A}\}$
  Sort $\mathcal{R}_{\mathsf{A}}$ in decreasing order on length of the antecedent
  for $i \leftarrow 1 \rightarrow |\mathcal{R}_{\mathsf{A}}| - 1$ **do**
    if Keep($\mathcal{R}_{\mathsf{A}}[i]$) $= false$ ***and*** HasSpec($\mathcal{R}$(A)$[i]$) $=true$ **then**
      continue
    end
    Let $\mathcal{R}_{\mathsf{A}}[i] = \mathsf{X} \Rightarrow \mathsf{A}$
    for $j \leftarrow i + 1 \rightarrow |\mathcal{R}_{\mathsf{A}}|$ **do**
      Let $\mathcal{R}_{\mathsf{A}}[j] = \mathsf{Y} \Rightarrow \mathsf{A}$
      $\mathsf{Q} = \mathsf{Y} \backslash \mathsf{X}$
      CondM $\leftarrow$ M(setFreq$[XA\neg Q]$, setFreq$[X\neg Q]$, setFreq$[A\neg Q]$, setFreq$[\neg Q]$)
      if X $\subset$ Y **then**
        HasSpec(X) $= true$
        if CondM $\leq \alpha$ **then**
          Keep(X) $= true$
        end
      end
    end
  end
end
for $\forall$ R $\in \mathcal{R}$ **do**
  if Keep(R) $= false$ **then**
    delete R
  end
end
return $\mathcal{R}$

5.2 Size of the Rule Set

Tables 7 and 8 show the number of rules generated for each dataset, pruning approach, and threshold. Observe each pruning approach consistently eliminates a substantial percentage of rules (compared against no pruning). The number of rules generated with robust redundancy is also significantly (P=.05) lower than for classical redundancy with most tested data. For the three exceptions (Insomnia data with thresholds -30 and -35, and Mushroom with threshold -2000), no significant difference in means is observed.

The cases where no significant difference in the number of non-redundant rules is observed occur using the strictest thresholds. In addition, the difference between the mean number of rules generated appears to increase as the interestingness threshold is relaxed. The number of rules appears to converge as the bound

Dataset	α	No Prune	Classic	Robust Specialisations	Robust (Both)
Aspergillosis	-50	29548.60 ± 4461.94	389.40 ± 31.39	391.60 ± 31.48	268.30 ± 21.10
	-75	4207.80 ± 352.07	119.20 ± 6.36	120.10 ± 6.41	88.90 ± 2.82
	-100	1073.60 ± 124.63	55.90 ± 5.90	56.20 ± 5.87	41.70 ± 3.45
	-125	383.10 ± 33.97	26.90 ± 2.55	27.00 ± 2.57	22.20 ± 1.77
	-150	163.90 ± 16.29	19.50 ± 1.55	19.50 ± 1.55	16.50 ± 1.28
	-175	89.90 ± 6.74	14.60 ± 1.08	14.70 ± 1.04	12.90 ± 1.02
	-200	49.40 ± 5.33	10.00 ± 1.04	10.20 ± 1.10	9.20 ± 0.77
Diabetes	-15	34543.20 ± 12832.45	1327.80 ± 99.71	1676.30 ± 235.23	823.60 ± 83.15
	-20	11816.40 ± 2574.97	613.00 ± 52.78	699.90 ± 73.42	394.40 ± 38.01
	-25	4152.80 ± 227.20	343.40 ± 18.07	365.00 ± 21.57	224.90 ± 10.38
	-30	2731.10 ± 229.94	244.50 ± 10.73	248.60 ± 11.75	162.40 ± 9.20
	-35	1656.20 ± 102.51	180.80 ± 7.58	183.40 ± 7.73	126.30 ± 5.47
	-40	1198.80 ± 86.62	143.60 ± 7.21	145.10 ± 7.38	102.50 ± 4.95
	-45	782.30 ± 78.51	107.10 ± 11.24	107.20 ± 11.29	78.80 ± 7.13
Fertility	-15	362405.60 ± 86789.45	595.30 ± 45.95	618.30 ± 48.06	352.80 ± 25.40
	-20	141740.50 ± 41787.17	278.00 ± 15.74	283.80 ± 15.80	176.80 ± 12.03
	-25	42389.60 ± 11074.22	176.10 ± 5.92	178.20 ± 6.07	111.20 ± 5.71
	-30	20686.90 ± 4609.10	132.50 ± 14.70	133.20 ± 15.04	85.00 ± 9.83
	-35	12736.90 ± 2347.48	113.60 ± 6.68	114.00 ± 7.06	72.80 ± 4.27
	-40	7713.60 ± 1830.05	90.50 ± 7.11	90.80 ± 7.11	62.40 ± 6.12
	-45	4718.80 ± 687.50	74.50 ± 10.29	74.60 ± 10.33	48.70 ± 7.68
Insomnia	-15	12812.70 ± 2399.55	624.00 ± 48.35	747.80 ± 64.69	402.10 ± 33.42
	-20	3180.50 ± 998.68	243.60 ± 37.06	269.00 ± 40.89	161.20 ± 24.88
	-25	876.70 ± 362.33	104.50 ± 15.75	112.50 ± 14.98	72.30 ± 12.27
	-30	271.10 ± 36.97	43.60 ± 6.41	46.80 ± 6.78	32.00 ± 4.42
	-35	136.40 ± 23.64	24.30 ± 5.93	25.40 ± 5.86	17.40 ± 3.59
	-40	59.40 ± 8.13	10.60 ± 1.25	12.40 ± 2.19	8.20 ± 1.20
	-45	31.60 ± 7.02	5.20 ± 1.70	6.10 ± 1.93	4.60 ± 1.47

Table 7: Number of rules generated for text and herbal prescription data.

on interesting rules is tightened, and diverge as it is relaxed. This supports the conclusion that our proposed approach is able to produce a practical number of rules from a larger number of potentially interesting associations. This quality is desirable as it allows the use of relaxed interestingness thresholds, lowering the risk of missing potentially useful associations.

We now examine performance when exclusively removing redundant specialisations. Given rule $X \Rightarrow Z$, specialisation redundancy (definition 2) uses the conditional association $X \Rightarrow Z|Q$ to provide an additional chance to obtain evidence for keeping rule $XQ \Rightarrow Z$ (with respect to the classical approach defined in section 2). All specialisations that are not classically redundant will also not be robust redundant. Robust specialisation redundancy will always return at least as many rules as the classical approach.

5.3 Rule Quality

Next we examine at the quality of the generated rules. As evidence has been given that we should not prune generalisations without first pruning specialisations, no results are reported for pruning generalisations exclusively. Tables 9 and 10 show the mean log P-values for each of the tested redundancy methods and thresholds. Despite the smaller generated rule set, it can be seen that in all cases the performance of robust pruning is equivalent or slightly better than for rules generated with classical redundancy.

5.4 Efficiency

The expanded search for robust redundancy increases the amount of time and space required. The main factor that effects both computational time and memory requirements is the number of nodes generated during the search. This can be seen by observing the similarity of the trends for the number of nodes generated (Figure 2) against time (Figure 3) and memory (Figure 4) (like Figure 2 values for robust and specialisation redundancy can appear quite similar).

Two factors contribute to the increased search space size. As robust specialisation redundancy is more permissive than classical redundancy, the pruning employed during the search must be less aggressive. Additionally, we do not prune based on minimality with robust redundancy. We note however that as our initial search differs from the existing Kingfisher algorithm only in the pruning strategies employed, it maintains the same worst case time and space complexity.

Figure 2 shows the number of nodes generated when searching with each data (values for robust and specialisation redundancy appear quite similar). Observe the difference in number of nodes generated when searching with robust and classical redundancy varies substantially. Some data (e.g. Aspergillosis and T10I4D100K) differ very little, while the greatest difference is observed for the Mushroom and T40I10D100K data.

In addition to comparing the number of nodes when searching with classical and robust redundancy, it is interesting to examine the number of nodes when

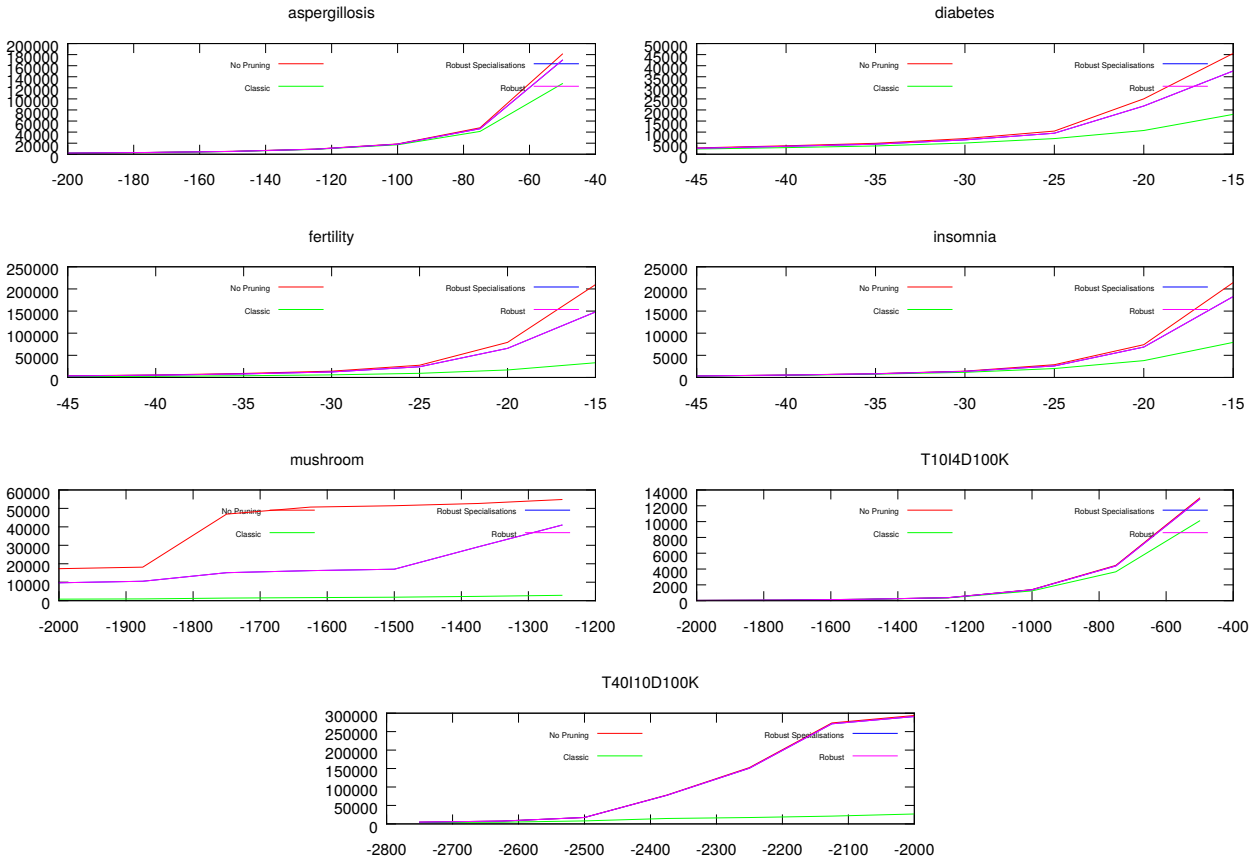


Figure 2: Number of nodes generated during mining vs. goodness threshold for each data and redundancy approach.

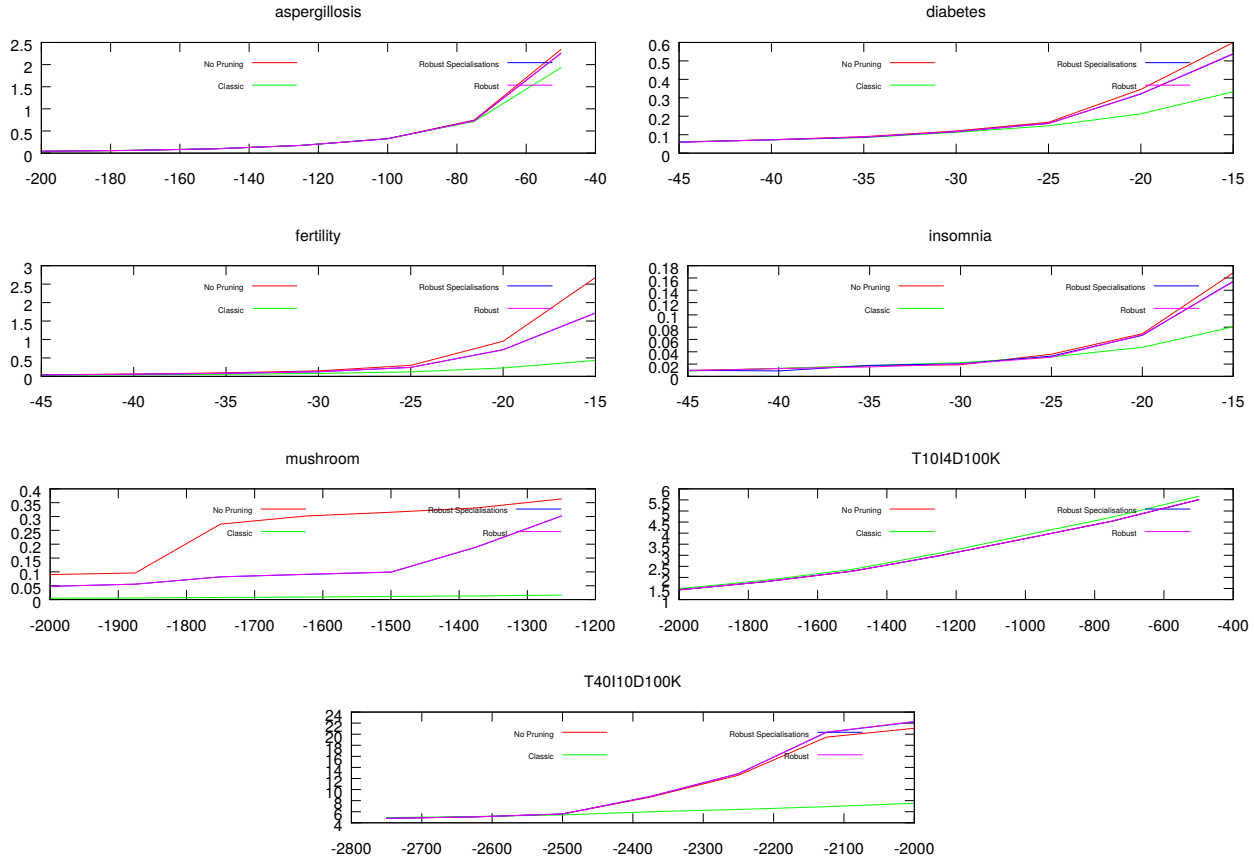


Figure 3: Average search time vs. goodness threshold for each data and redundancy approach.

Dataset	α	No Prune	Classic	Robust Specialisations	Robust (Both)
Mushroom	-1250	61767.80 ± 158.39	409.70 ± 5.32	568.70 ± 7.42	227.40 ± 2.95
	-1375	37501.10 ± 4080.51	308.10 ± 7.70	342.40 ± 10.14	166.00 ± 6.38
	-1500	22634.50 ± 92.39	229.70 ± 5.67	239.40 ± 5.97	125.50 ± 1.67
	-1625	22049.80 ± 39.14	191.30 ± 2.54	196.30 ± 2.85	114.80 ± 1.77
	-1750	19980.00 ± 2498.40	140.70 ± 5.51	141.70 ± 5.51	93.30 ± 6.17
	-1875	7507.80 ± 78.31	88.60 ± 3.75	89.60 ± 3.75	56.70 ± 1.08
	-2000	6430.80 ± 522.10	38.90 ± 5.39	39.90 ± 5.39	34.70 ± 4.15
T10I4D100K	-500	17287.60 ± 191.08	6114.80 ± 53.65	6114.80 ± 53.65	4302.40 ± 43.14
	-750	3484.70 ± 73.63	1568.00 ± 28.03	1568.00 ± 28.03	1217.30 ± 21.15
	-1000	750.40 ± 31.17	411.90 ± 12.19	411.90 ± 12.19	353.50 ± 9.51
	-1250	169.70 ± 3.67	99.80 ± 2.51	99.80 ± 2.51	85.30 ± 2.30
	-1500	76.70 ± 4.39	41.70 ± 2.73	41.70 ± 2.73	36.50 ± 2.47
	-1750	28.50 ± 3.82	16.90 ± 1.55	16.90 ± 1.55	15.60 ± 1.31
	-2000	2.90 ± 1.53	2.90 ± 1.53	2.90 ± 1.53	2.90 ± 1.53
T40I10D100K	-2000	297056.60 ± 16747.78	5477.00 ± 211.44	5477.00 ± 211.44	3675.70 ± 133.93
	-2125	227409.30 ± 30167.75	4165.60 ± 181.86	4165.60 ± 181.86	2874.70 ± 116.69
	-2250	80480.40 ± 31503.32	3001.90 ± 159.61	3001.90 ± 159.61	2195.80 ± 93.08
	-2375	32533.60 ± 24486.95	1746.40 ± 372.79	1746.40 ± 372.79	1323.10 ± 264.55
	-2500	5693.70 ± 611.02	660.20 ± 78.37	660.20 ± 78.37	528.70 ± 69.71
	-2625	1933.20 ± 631.18	341.90 ± 58.24	341.90 ± 58.24	282.20 ± 47.46
	-2750	615.10 ± 272.55	193.50 ± 57.87	193.50 ± 57.87	172.50 ± 51.47

Table 8: Number of rules generated for traditional data.

no redundancy based pruning is used. There appears to be a substantial difference between the number of nodes generated without pruning when compared against using robust redundancy. This implies that the bounds computed in algorithm 4 have a reasonable effect on the size of the search space.

Three exceptions to the above observation occur with the Aspergillosis, T10I4D100K, and T40I10D100K data. For Aspergillosis and T10I4D100K we note there is also little difference between the number of nodes generated using robust and classical redundancy. The implication here is that the majority of the pruning is being done by lapis philosophorum. However, for T40I10D100K there is a substantial difference in performance with robust and classical redundancy.

Figure 3 reports the time for all searches (including pruning in algorithms 5 and 6). In all cases the required search time was quite manageable. Even for T40I10D100K, all searches completed in less than 30 seconds (all other data completed much quicker). In practice memory appears to become an issue long before time required for the search.

6 Conclusion

This paper concerns the problem of excessively large rule sets in association mining. This impacts user interpretation as the presence of spurious or redundant associations can obscure interesting relationships. A novel approach for identifying and removing redundant rules is presented, which we call robust redundancy. We demonstrate for multiple datasets that robust redundancy produces smaller overall rule sets compared to classical redundancy approaches. These rule sets hold as well or better in future data than those generated using classical redundancy.

Prior work compared rules using only the interestingness computed with their respective contingency tables. Such a comparison fails to take into account information included in instances containing only part of the antecedent. Robust redundancy is able to use this information to discover interesting specialisations that would be incorrectly removed with a classical approach.

We also remove rules which are redundant artefacts of their non-redundant specialisations. Unlike previous work [9, 17] that evaluate generalisations based on their exclusive domain with respect to the set of all specialisations, we base our method on comparisons to individual rules. We also present the first work using both specialisation and generalisation redundancy in a rule based context (as opposed to the work of Webb [17] with itemsets).

References

[1] C. C. Aggarwal and P. S. Yu. A new approach to online generation of association rules. *Knowledge and Data Engineering, IEEE Transactions on*, 13(4):527–540, 2001.

[2] R. Agrawal, T. Imieli, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM International Conference on Management of Data (SIGMOD)*, pages 207–216. ACM, 1993.

[3] M. Ashrafi, D. Taniar, and K. Smith. A new approach of eliminating redundant association rules. *Database and Expert Systems Applications*, 3180:465–474, 2004.

[4] S. Brin, R. Motwani, J. D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket data. In *Proceedings of the 1997 ACM International Conference on Management of Data (SIGMOD)*, pages 255–264. ACM, 1997.

[5] W. Hämäläinen. Efficient discovery of the top-k optimal dependency rules with fisher's exact test of significance. In *Proceedings of the Tenth IEEE International Conference on Data Mining (ICDM)*, pages 196–205, Dec 2010.

Dataset	α	No Prune	Classic	Robust Specialisations	Robust (Both)
Aspergillosis	-50	-57.44 ± 3.85	-73.19 ± 2.67	-73.25 ± 2.64	-77.29 ± 2.86
	-75	-91.80 ± 3.04	-115.51 ± 2.75	-115.36 ± 2.73	-121.00 ± 2.09
	-100	-125.78 ± 5.52	-155.54 ± 9.36	-155.37 ± 9.22	-165.06 ± 7.93
	-125	-157.39 ± 5.65	-206.80 ± 10.15	-206.60 ± 10.25	-215.07 ± 9.12
	-150	-197.29 ± 8.45	-240.32 ± 7.31	-240.32 ± 7.31	-249.13 ± 7.18
	-175	-224.52 ± 6.51	-265.57 ± 7.95	-265.10 ± 7.75	-273.72 ± 8.71
	-200	-259.87 ± 11.65	-297.70 ± 13.49	-296.33 ± 14.14	-304.60 ± 11.22
Diabetes	-15	-12.17 ± 1.72	-16.78 ± 0.90	-15.30 ± 1.28	-17.67 ± 1.39
	-20	-18.97 ± 2.38	-26.30 ± 1.74	-24.48 ± 1.91	-27.65 ± 2.27
	-25	-30.79 ± 1.34	-36.92 ± 1.58	-35.67 ± 1.53	-39.42 ± 1.54
	-30	-35.59 ± 2.38	-43.34 ± 2.05	-43.11 ± 2.05	-47.00 ± 2.27
	-35	-44.02 ± 1.78	-50.70 ± 1.22	-50.44 ± 1.18	-54.14 ± 1.27
	-40	-48.68 ± 2.62	-55.41 ± 2.06	-55.15 ± 2.07	-58.88 ± 1.94
	-45	-56.41 ± 3.62	-62.17 ± 4.00	-62.14 ± 4.01	-66.32 ± 4.32
Fertility	-15	-14.63 ± 1.86	-20.65 ± 1.35	-20.18 ± 1.29	-21.54 ± 1.23
	-20	-19.21 ± 3.37	-33.41 ± 3.24	-33.00 ± 3.06	-34.13 ± 3.22
	-25	-29.80 ± 3.75	-43.14 ± 3.31	-42.84 ± 3.30	-42.72 ± 3.14
	-30	-34.94 ± 3.51	-54.64 ± 3.28	-54.49 ± 3.30	-55.67 ± 2.92
	-35	-38.45 ± 4.19	-56.66 ± 4.09	-56.55 ± 4.10	-57.15 ± 4.08
	-40	-44.74 ± 5.33	-62.70 ± 6.00	-62.55 ± 5.92	-63.21 ± 5.56
	-45	-50.15 ± 3.88	-66.96 ± 5.25	-66.92 ± 5.28	-66.96 ± 5.36
Insomnia	-15	-10.67 ± 1.41	-13.27 ± 0.74	-12.83 ± 0.80	-13.05 ± 0.72
	-20	-16.50 ± 2.50	-18.77 ± 1.76	-18.68 ± 1.75	-18.61 ± 1.78
	-25	-24.19 ± 3.16	-24.32 ± 1.88	-24.58 ± 1.97	-24.39 ± 2.06
	-30	-31.51 ± 1.70	-30.45 ± 1.52	-30.45 ± 1.64	-30.64 ± 1.62
	-35	-35.02 ± 3.08	-33.93 ± 2.99	-34.37 ± 3.04	-35.13 ± 3.09
	-40	-44.25 ± 4.01	-41.28 ± 2.06	-40.81 ± 2.88	-42.70 ± 2.77
	-45	-57.66 ± 5.25	-55.98 ± 6.25	-55.34 ± 6.45	-58.16 ± 7.72

Table 9: Rule performance using average log P-values approaches for text and herbal prescription data.

[6] W. Hämäläinen. Kingfisher: an efficient algorithm for searching for both positive and negative dependency rules with statistical significance measures. *Knowledge and Information Systems*, 32(2):383–414, 2012.

[7] M. M. Leeflang, J. J. Deeks, C. Gatsonis, P. M. Bossuyt, and Group Cochrane Diagnostic Test Accuracy Working. Systematic reviews of diagnostic test accuracy. *Ann Intern Med*, 149(12):889–97, 2008.

[8] J. Li and O. Zaiane. Associative classification with statistically significant positive and negative rules. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM)*, pages 633–642. ACM, 2015.

[9] B. Liu, W. Hsu, and Y. Ma. Identifying non-actionable association rules. In *Proceedings of the Seventh ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 329–334, 2001.

[10] M. McGrane and S. K. Poon. Interaction as an interestingness measure. In *2010 IEEE International Conference on Data Mining Workshops*, pages 726–731, 2010.

[11] G. Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. *Knowledge discovery in databases*, pages 229–238, 1991.

[12] C. Song and T. Ge. Discovering and managing quantitative association rules. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management (CIKM)*, pages 2429–2434. ACM, 2013.

[13] P. Tan, V. Kumar, and J. Srivastava. Selecting the right objective measure for association analysis. *Inf. Syst.*, 29(4):293–313, 2004.

[14] F. Verhein and S. Chawla. Using significant, positively associated and relatively class correlated rules for associative classification of imbalanced datasets. In *Proceedings of the Seventh IEEE International Conference on Data Mining (ICDM)*, pages 679–684, 2007.

[15] G. I. Webb. Discovering significant rules. In *Proceedings of the Twelfth ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 434–443. ACM, 2006.

[16] G. I. Webb. Discovering significant patterns. *Machine Learning*, 68(1):1–33, 2007.

[17] G. I. Webb. Self-sufficient itemsets: An approach to screening potentially interesting associations between items. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 4(1):3, 2010.

[18] M. J. Zaki. Generating non-redundant association rules. In *Proceedings of the Sixth International Conference on Knowledge discovery and Data Mining (KDD)*, pages 34–43, 2000.

Dataset	α	No Prune	Classic	Robust Specialisations	Robust (Both)
Mushroom	-1250	-1542.40 ± 8.59	-1604.50 ± 9.79	-1535.21 ± 8.13	-1622.80 ± 9.53
	-1375	-1674.28 ± 39.12	-1700.65 ± 14.68	-1676.63 ± 14.71	-1733.22 ± 20.36
	-1500	-1853.86 ± 17.29	-1793.91 ± 19.56	-1787.94 ± 19.16	-1842.73 ± 17.71
	-1625	-1860.36 ± 11.36	-1843.53 ± 10.95	-1840.13 ± 11.21	-1869.33 ± 10.59
	-1750	-1886.27 ± 24.84	-1903.90 ± 8.54	-1904.92 ± 8.53	-1915.04 ± 9.57
	-1875	-2058.56 ± 15.00	-1996.28 ± 15.97	-1997.05 ± 15.92	-2016.88 ± 13.73
	-2000	-2064.61 ± 13.26	-2040.31 ± 15.01	-2040.44 ± 14.65	-2044.43 ± 13.54
T10I4D100K	-500	-654.97 ± 3.75	-677.78 ± 2.91	-677.78 ± 2.91	-691.13 ± 3.47
	-750	-901.33 ± 7.01	-923.40 ± 5.67	-923.40 ± 5.67	-934.32 ± 6.11
	-1000	-1155.35 ± 11.23	-1171.87 ± 9.15	-1171.87 ± 9.15	-1174.81 ± 8.79
	-1250	-1496.84 ± 15.98	-1489.76 ± 14.07	-1489.76 ± 14.07	-1497.53 ± 14.53
	-1500	-1679.61 ± 30.62	-1690.91 ± 30.69	-1690.91 ± 30.69	-1702.46 ± 30.36
	-1750	-1839.38 ± 44.39	-1865.57 ± 41.17	-1865.57 ± 41.17	-1869.08 ± 40.72
T40I10D100K	-2000	-2205.18 ± 32.30	-2282.00 ± 17.71	-2282.00 ± 17.71	-2302.10 ± 18.75
	-2125	-2201.06 ± 42.97	-2339.99 ± 33.51	-2339.99 ± 33.51	-2360.88 ± 33.30
	-2250	-2305.75 ± 57.95	-2408.56 ± 28.13	-2408.56 ± 28.13	-2423.45 ± 26.10
	-2375	-2410.26 ± 100.15	-2470.00 ± 58.98	-2470.00 ± 58.98	-2481.85 ± 56.55
	-2500	-2585.09 ± 42.48	-2621.87 ± 51.48	-2621.87 ± 51.48	-2631.06 ± 52.50
	-2625	-2691.38 ± 36.23	-2748.60 ± 26.65	-2748.60 ± 26.65	-2756.40 ± 25.11
	-2750	-2686.73 ± 40.39	-2706.56 ± 33.16	-2706.56 ± 33.16	-2710.19 ± 33.49

Table 10: Rule performance using average log P-values approaches for traditional data.

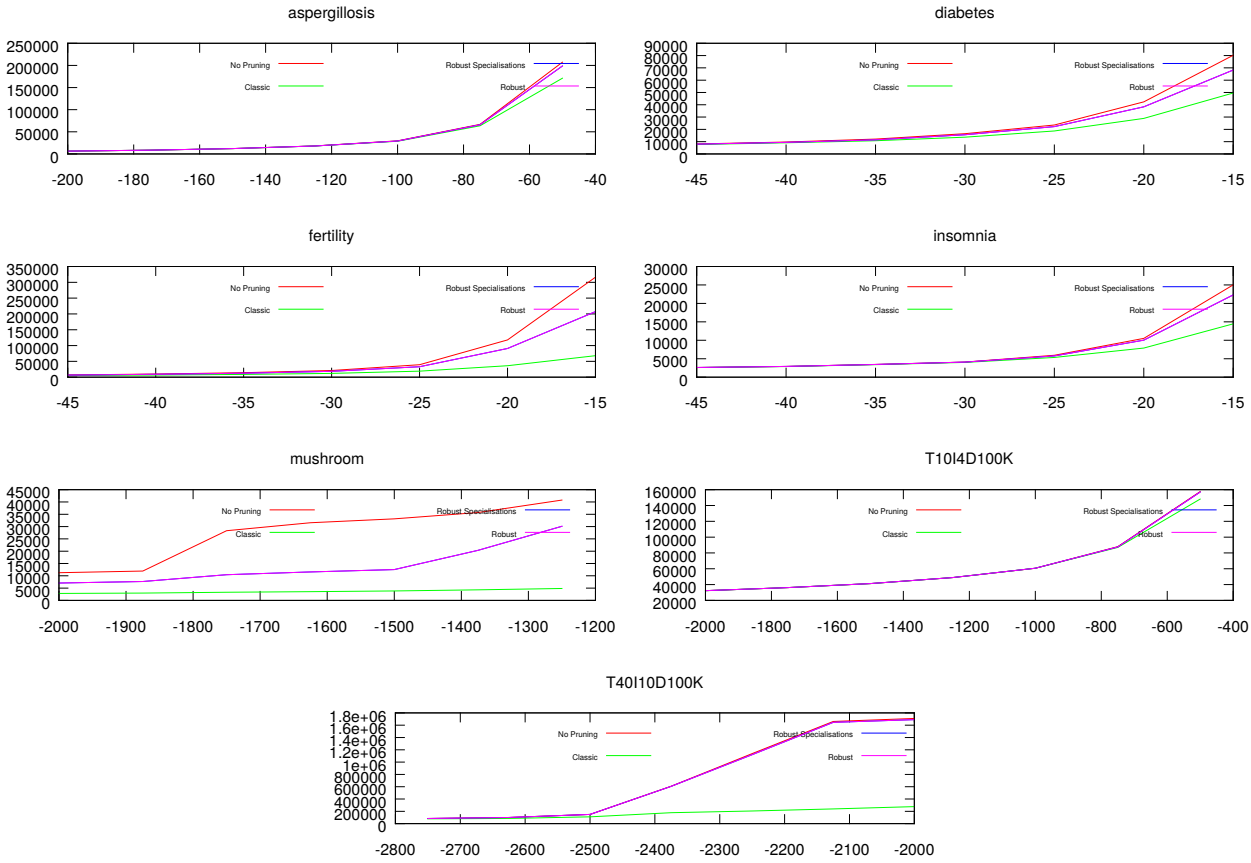


Figure 4: Memory usage (kb) vs. goodness threshold for each data and redundancy approach.

Measuring the Similarity between Rule Lists

Sam Fletcher **Md Zahidul Islam**

School of Computing and Mathematics,
Charles Sturt University,
Bathurst, New South Wales 2795, Australia,
Email: safletcher@csu.edu.au
Email: zislam@csu.edu.au

Abstract

The ability to extract knowledge from data has been the driving force of Data Mining since its inception, and of statistical modeling long before even that. Actionable knowledge often takes the form of patterns, otherwise known as rules, where a set of antecedents can be used to infer a consequent. In this paper we offer a solution to the problem of comparing different sets of rules. Our solution allows comparisons between rule lists that were derived from different techniques (such as different classification algorithms), or made from different samples of data (such as temporal data or data perturbed for privacy reasons). We propose using the Jaccard Index to measure the similarity between rule lists, by converting each rule into a single element within the set of rules. Our measure focuses on providing conceptual simplicity, computational simplicity, interpretability, and wide applicability. The results of this measure are compared to Prediction Accuracy in the context of a real-world data mining scenario.
Keywords: Data Mining, Patterns, Rules, Utility Measures, Quality Evaluation.

1 Introduction

The discovery of patterns in data is the cornerstone of Data Mining; sometimes for predicting the future, and other times for extracting meaning. It is the latter case that this paper will concern itself with. By extracting meaning, statisticians and data analysts are able to elevate otherwise meaningless data into usable information; information that can be acted on or used to discover knowledge (Han et al. 2006). The extraction of meaning from data is most often explicated as mining patterns from data, preferably in a form that is interpretable by humans. Many branches of Data Mining have concerned themselves over the years with this endeavor, not limited to Frequent Pattern (Itemset) Mining (Han et al. 2007), Decision Trees (Quinlan 1993) and Forests (Breiman 2001*a*), and Association Rule Mining (Ordonez & Zhao 2011). Constructing models from data in order to make accurate predictions is useful, but it is important to remember that the accuracy of a model does not guarantee the truthfulness of the patterns contained in the model. The history of science is littered with untruthful models that accurately predicted observations, from Bohr's model of the atom to the geocentric model of the Solar System. Assessing the patterns discovered by Data Mining algorithms is often overlooked in the current zeitgeist, instead favoring a dogged pursuit of higher Prediction Accuracy (Hand 2006, Wagstaff 2012). We argue that this is to the detriment of the discipline and provide a simple but powerful method for comparing sets of patterns, aiding a renewed focus in pattern assessment seen in recent years (Aodha et al. 2014, Letham et al. 2013).

Two terms warrant further explanation: data and patterns. By "data", we mean any two dimensional array of independent rows, each defined by the values ascribed to it from each column. The rows are most often called "records" and might each describe a single person, for example. The columns are most often called "attributes" and might each be a property possessed by people, such as education or age. By "patterns", we refer to the structured meaning (i.e. semantics) of a set of criteria that when met, allow an analyst to predict that additional information that was not previously known. We make no distinction between the words "pattern" and "rule" in this paper, and use them interchangeably.

More formally, rules take the form $\mathcal{X}_i \rightarrow y_j \in \mathcal{Y}$. That is to say, if criteria $\mathcal{X}_i$ is met, attribute $\mathcal{Y}$ is predicted to equal y_j. $\mathcal{Y}$ is known as the consequent or class attribute, and can be found dynamically or be user-defined. Functioning as the antecedent, $\mathcal{X}_i$ is a set of conditions for various attributes $\mathcal{A}_k$, where $\mathcal{A}_k = \{a_{k,0}, a_{k,1}, \ldots, a_{k,v}, \ldots, a_{k,|\mathcal{A}_k|}\}$, $\mathcal{A}_k \in \mathcal{A}$. Conditions take the form $\mathcal{A}_k = a_{k,v}$, or can use other operators such as $\mathcal{A}_k > a_{k,v}$ if $\mathcal{A}_k$ is ordered. An example would be $\{Education = PhD,\ Age > 45\} \rightarrow Income = High$, or $\mathcal{X}_i$ might detail the presence of items such as $\{milk, bread\}$ and predict that the consequent *eggs* will also be present. Other examples can be seen in Table 1. Patterns in any of these forms are often referred to as decision rules or association rules, and when collected together can be called a decision list or rule list (Letham et al. 2013).[1]

1.1 Problem Statement

Consider the following question:

> *Given two rule lists $\mathcal{Z}_1$ and $\mathcal{Z}_2$, how similar are they?*

To the best of our knowledge, this question has yet to be answered in the literature, and is answered in this paper. The ability to answer this question has clear

[1] For example, a Decision Tree might be "flattened" so as to no longer have roots or leaves, and it merely becomes an unordered set (a rule list) of unordered sets of attribute conditions (rules). The Decision Tree would then be indistinguishable from a Rule List.

Table 1: Some examples of patterns (rules) discovered by CART in the WBC dataset.

i	$\mathcal{X}_i$	y_j
0	$Clump\ Thickness \leq 5.5$ AND $Uniformity\ of\ Cell\ Size \leq 2.5$ AND $Bare\ Nuclei \leq 4.5$	Benign
1	$Uniformity\ of\ Cell\ Size > 2.5$ AND $Uniformity\ of\ Cell\ Shape \leq 2.5$	Benign
2	$Clump\ Thickness > 6.5$ AND $Uniformity\ of\ Cell\ Size > 2.5$ AND $Uniformity\ of\ Cell\ Size \leq 4.5$ AND $Uniformity\ of\ Cell\ Shape > 2.5$ AND $Bare\ Nuclei > 2.5$ AND $Mitoses \leq 1.5$	Malignant
3	$Uniformity\ of\ Cell\ Size > 4.5$ AND $Uniformity\ of\ Cell\ Shape > 2.5$ AND $Bland\ Chromatin > 4.5$	Malignant

benefits in many areas of Data Mining and Knowledge Discovery, such as:

- comparing different classifiers, or classifiers with different parameters (Islam & Giggins 2011, Shotton et al. 2013);
- comparing patterns discovered manually by statisticians, to patterns discovered with machine learning techniques (Breiman 2001*b*);
- comparing the quality of the patterns discovered in data before and after applying privacy-preserving techniques to the data (Fletcher & Islam 2014, 2015*a*); and
- finding differences in different samples of data, including temporal scenarios with time series (Baron et al. 2003).

Answering the above question requires a measurement of some kind, and it is a measure (more specifically, a metric) that we propose in this paper. We convert the rules in $\mathcal{Z}_1$ and $\mathcal{Z}_2$ into discrete elements, and take advantage of the Jaccard Index to measure the similarity between sets made up of these elements. This is discussed in full in Section 3. In order to be a useful measure, we consider several external factors to be part of the problem statement. These are factors that often determine whether researchers and data miners willfully choose to use a measure:

- conceptual simplicity;
- computational simplicity;
- interpretability; and
- wide applicability.

We discuss how our proposed measure fulfills these external factors in Section 3. Section 2 provides additional background information and related work. In Section 4, we demonstrate our measure in action with an illustrative real-world scenario in which a data miner wishes to compare two classifiers. We conclude with Section 5.

2 Related Work

Patterns play a key role in the knowledge discovery and decision-making process. Several fields of machine learning – notably Frequent Pattern Mining (Han et al. 2007, 2000) – focus specifically on finding patterns. Fields such as Classification can also find patterns by using Decision Forests (Breiman 2001*a*) or other classifiers (Han et al. 2006). Patterns can be assessed for their usefulness (Geng & Hamilton 2006) and their interpretability (Letham et al. 2013), or monitored for any changes in temporal scenarios (Baron et al. 2003). While differing in methodology, approaches such as these agree on the importance of patterns and attest to the value of patterns for gaining knowledge.

It is important to note that assessing the quality of patterns in these ways is not the same as measuring the performance of a model at achieving a goal (Caruana & Niculescu-Mizil 2004, Sokolova & Lapalme 2009). Prediction Accuracy is often used to measure a model's performance (Cheng et al. 2007, Letham et al. 2013), but is disconnected from any reliable assessments of pattern quality (Fletcher & Islam 2014, Islam et al. 2003). A user should be aware of the specific goals they wish their model to achieve and how important the truthfulness (Kifer & Gehrke 2006) or interpretability of the patterns in the model are, and then use multiple measures to assess if their needs are met.

Our proposed application of the Jaccard Index assesses if two lists of patterns are similar. Attempts to measure the similarity between trusted patterns and newly discovered patterns have been made in the past (Islam & Brankovic 2011), but the measure used was designed for a specific problem, lacking any applicability in a wider context. We discuss the value of widely applicable measures in Section 3.2.

The field of Privacy Preserving Data Mining (Dwork 2008, Fung et al. 2010) is known for struggling with the inherent trade-off that must be made between the amount of privacy provided and the quality of the perturbed data (Fung et al. 2005, Nergiz & Clifton 2007). A naive approach would be to use measures like RMSE to compare the original data to the perturbed data directly (Willmott 1982, Willmott et al. 2009), however this ignores the correlations in the data necessary for information discovery (Agrawal & Aggarwal 2001, Fletcher & Islam 2015*b*). Just like how Frequent Pattern Mining and other fields use

Prediction Accuracy heavily, so too does Privacy Preserving Data Mining (Chaudhuri et al. 2011, Friedman & Schuster 2010, Fung et al. 2005, Mohammed et al. 2011), but the same problems encountered by the former when assessing patterns also plague the latter (Fletcher & Islam 2014, Islam et al. 2003). Attempts at directly measuring the loss of pattern retention as privacy needs are increased have been made (Fletcher & Islam 2014), but the measure is not applicable in any wider context outside of privacy preservation.

There is therefore a need for insightful ways to compare between patterns gained by different means, regardless of what data mining algorithms or data are used. We do so in this paper, taking both practical and mathematical considerations into account (Meila 2007).

3 Comparing Rule Lists with the Jaccard Index

The Jaccard Index (Jaccard 1901) is a well-known measurement of the similarity between two sets $\mathcal{S}$ and $\mathcal{T}$, defined as the size of the intersection divided by the size of the union of the two sets:

$$J(\mathcal{S},\mathcal{T}) = \frac{|\mathcal{S} \cap \mathcal{T}|}{|\mathcal{S} \cup \mathcal{T}|} \ , \tag{1}$$

where we say $J(\mathcal{S},\mathcal{T}) = 1$ if $|\mathcal{S} \cup \mathcal{T}| = 0$. By reinterpreting a rule list (a collection of patterns) as a set of elements, we are able to use the Jaccard Index to measure the similarity between two rule lists. In Section 3.1 we describe how we convert rules in rule lists $\mathcal{Z}_1$ and $\mathcal{Z}_2$ into elements for sets $\mathcal{S}$ and $\mathcal{T}$. In Sections 3.2 and 3.3 we outline the practical and mathematical benefits of using the Jaccard Index to measure the difference between two rule lists.

3.1 Converting rules into elements in a set

Our aim is to compare two rule lists and describe their similarities with an intuitive, quantitative number. To do so with the Jaccard Index, we must first translate each rule $\mathcal{X}_i \rightarrow y_j$ (such as those seen in Table 1) into element s_i of set $\mathcal{S}$. Each antecedent $\mathcal{X}_i$ is made up of attributes – numerical, categorical or binary attributes – that specify the conditions that must be met in order for the consequent $\mathcal{Y}$ to be predicted to equal y_j. To condense the set $\mathcal{X}_i$ as well as y_j into a single element s_i, we use the following equation:

$$s_i = \mathbf{1}_{\mathcal{X}_i}(\mathcal{A}_0), \ldots, \mathbf{1}_{\mathcal{X}_i}(\mathcal{A}_k), \ldots, \mathbf{1}_{\mathcal{X}_i}(\mathcal{A}_{|\mathcal{A}|}), y_j^i \tag{2}$$

where we use y_j^i to simply refer to the class value (consequent) of rule $\mathcal{X}_i$, and $\mathbf{1}_{\mathcal{X}_i}(\mathcal{A}_k)$ is the indicator function:

$$\mathbf{1}_{\mathcal{X}_i}(\mathcal{A}_k) := \begin{cases} 1 & \mathcal{A}_k \in \mathcal{X}_i \\ 0 & \mathcal{A}_k \notin \mathcal{X}_i \end{cases} \ . \tag{3}$$

Essentially, Equation 2 is recording the presence or absence of each attribute in antecedent $\mathcal{X}_i$, as well as recording the consequent y_j. Table 2 illustrates this with converted versions of the rules seen in Table 1. The three 1's in $\mathcal{X}_0$ refer to the positions in the WBC dataset of the three attributes used by $\mathcal{X}_0$ in Table 1, and similarly for the other rules. We include the consequent in s_i because of the role it plays in the definition of a pattern – without a consequent, an antecedent hardly means anything at all.

Table 2: The encoded versions of the rules shown in Table 1.

i	s_i
0	$110001000a$
1	$011000000a$
2	$111001001b$
3	$011000100b$

Note that it is not possible to include the attribute *values* in the encoding without a large number of heuristics to handle the edge cases, as well as heuristics for handling the fundamental difference between ordered and unordered data. For example, "PhD" and "Bachelors" are clearly different, but what about "45" and "44"? Imagine the following pair of rules: $\{Age > 45\} \rightarrow Income = High$ and $\{Age \leq 44\} \rightarrow Income = Low$; it is computationally infeasible to detect that pairs of rules such as these are, in fact, the same.

3.2 Practical considerations

As discussed in Section 1.1, a good measure should not be difficult for data analysts to harness effectively. We assess our proposed application of the Jaccard Index with four factors that influence an analyst's decision to use a measure:

3.2.1 Conceptual simplicity

A good measure is one that can be easily and intuitively understood. If a measure has too many moving parts or variables, it can quickly become a "black box" of sorts, where analysts can no longer conceptualize all the possible outputs that the measure could produce. Our implementation of the Jaccard Index avoids these risks by being very straight-forward – it is the number of patterns two rule lists share, divided by the number of unique patterns across both lists. There are no variables beyond the two rule lists, and there are no parameters that need expert knowledge to properly adjust. Our encoding of the rules into single elements is simple and intuitive, making the conceptualization of the processes involved in the measure no more difficult than in a standard application of the Jaccard Index. Incorporating attribute value conditions into the encoding process would require a robust definition of "similarity" or "distance" for both numerical and categorical attribute values that did not induce any biases in the calculation, and would undoubtedly increase the conceptual complexity of the measure. This is one avenue for future work, but currently appears to be infeasible due to the fundamental differences between numerical and categorical attributes.

3.2.2 Computational simplicity

A measure can become infeasible if it does not scale well as the variables become large. The computation time of our measure is very satisfactory, mostly due to the fact that it does not use the underlying raw data in any way, or even a classifier – it just needs the rule lists. Encoding all the rules in $\mathcal{Z}_1$ and $\mathcal{Z}_2$ as the sets $\mathcal{S}$ and $\mathcal{T}$ requires each rule in both lists to be read and converted once, with the presence of each

attribute being checked once in each rule. Therefore the computational complexity is $O(|\mathcal{A}|(|\mathcal{Z}_1| + |\mathcal{Z}_2|))$. Once $\mathcal{S}$ and $\mathcal{T}$ have been generated, the intersection and union in the Jaccard Index can both be calculated by comparing each element in $\mathcal{S}$ to every element in $\mathcal{T}$, where each element is $|\mathcal{A}|$ digits long. This gives a computational complexity of $O(|\mathcal{A}| \cdot |\mathcal{S}| \cdot |\mathcal{T}|)$. Since the length of the respective rule lists and sets will always be equal, the total computation time of our measure is $O(|\mathcal{A}|(|\mathcal{Z}_1| + |\mathcal{Z}_2| + |\mathcal{Z}_1| \cdot |\mathcal{Z}_2|))$. As a rough example, classification algorithms such as Random Forest (Breiman 2001*a*) might generate several hundred rules, and the kinds of datasets that Random Forest might be applied to generally have less than 1000 attributes.

3.2.3 Interpretability

In order for analysts to judge the result of a measure and act on it in a meaningful way, the result needs to be interpretable. This can be as simple as being able to parse the result into a sentence. In our case, our measure can be interpreted as "out of all the rules that appear in rule lists $\mathcal{Z}_1$ and $\mathcal{Z}_2$, $(J(\mathcal{S},\mathcal{T}) \times 100)\%$ can be found in both lists". An addendum could be added about how the rules are compared at a level of granularity that ignores differences in attribute value conditions, without making the measure any more difficult to understand and interpret.

3.2.4 Wide applicability

Measures that accurately encapsulate specific scenarios are undeniably useful, but measures capable of crossing academic discipline boundaries are far more appealing. They provide ways for researchers and professionals to interface with work outside of their personal scope and build connections between otherwise isolated fields of science. Historically, the Statistics and Machine Learning disciplines have learned this lesson the hard way, with the relatively new field of Machine Learning often recreating mathematics and methods invented previously by statisticians (Breiman 2001*b*). The ubiquity of Prediction Accuracy as a measure of performance and the role it plays in creating dialogue among researchers demonstrates the advantages of measures with wide applicability. Our proposed encoding method and application of the Jaccard Index is general enough to be viable in any situation involving two groups of patterns discovered from data with the same attributes. Our measure is completely independent of how the patterns were discovered, built, or connected. It can handle non-binary consequents – a common constraint for other measures (Felkin 2007). It can handle continuous consequents as well, but only if the range of values are first discretized into "buckets" (Kotsiantis & Kanellopoulos 2006). It can even work in scenarios without a $\mathcal{Y}$ component, where $\mathcal{X}_i$ might merely represent a collection of attributes that appear together frequently. In this situation, s_i would intuitively drop the y_j component and become equal to $\mathbf{1}_{\mathcal{X}_i}(\mathcal{A}_0), \ldots, \mathbf{1}_{\mathcal{X}_i}(\mathcal{A}_k), \ldots, \mathbf{1}_{\mathcal{X}_i}(\mathcal{A}_{|\mathcal{A}|})$.

Our measure is conceptually and computationally simple, it produces easily interpretable results, and it is an appropriate measure in a wide variety of scenarios. Not constraining ourselves by requiring definitions of similarity or distance for numerical and categorical attribute value conditions allows these qualities to be improved even further.

3.3 Mathematical properties

A good measure should be capable of distinguishing between slightly different rule lists and very different rule lists. The Jaccard Index possess several properties that make this possible, as well as possessing properties that strengthen the versatility of the measure. Our encoding method is designed in a way that does not interfere with these properties.

3.3.1 Bounds

The Jaccard Index has the bounds $0 \leq J(\mathcal{S},\mathcal{T}) \leq 1$. We can narrow down the upper bound further by using the difference in size between the lists. In situations where $|\mathcal{S}| > |\mathcal{T}|$, the maximum similarity is when $\mathcal{T} \subseteq \mathcal{S}$, in which case $J(\mathcal{S},\mathcal{T}) = |\mathcal{T}|/|\mathcal{S}|$. The larger the size difference between the sets, the smaller the upper bound is. To put it formally:

$$J(\mathcal{S},\mathcal{T}) \leq \frac{|\mathcal{T}|}{|\mathcal{S}|},\ \textit{where}\ |\mathcal{S}| \geq |\mathcal{T}|\ . \tag{4}$$

It is reasonable to assume that the user will know the size of sets, making this upper bound easy to incorporate when interpreting the measure's result, and possibly being quite informative. A similar situation exists for the popular Prediction Accuracy measure, where the lower bound is equal to the relative frequency of the most common class value.[2]

If the flexibility of the Jaccard Index's upper bound is undesirable, a user is free to shorten $\mathcal{S}$ until both lists have equal length. The Frequent Pattern Mining discipline does this often, only selecting rules with high support (Han et al. 2007, 2000). Many other measures could be used to remove the least valuable rules depending on how the user defines "valuable" (Geng & Hamilton 2006). A user could also divide the Jaccard Index result by the upper bound and reinterpret the result as "the ratio between the actual result and the best possible result", but this approach could easily misguide the user about how similar the rule lists actually are and is not recommended.

The lower bound can also be narrowed in very specific scenarios: when the number of rules in $\mathcal{S}$ and $\mathcal{T}$ is more than the number of unique rules that could exist. This indicates that $|\mathcal{S} \cap \mathcal{T}| > 0$; some overlap between the sets must exist. This can occur when the number of unique rules that could exist given the number attributes and class values in the dataset (i.e. the number of ways you could write s_i, $|\mathcal{Y}|(2^{|\mathcal{A}|} - 1)$) is less than $|\mathcal{S}| + |\mathcal{T}|$.[3][4]

It is also possible that the pattern creation process puts a constraint k on the number of attributes that could be present in any one pattern (Webb & Brain 2002). It has been demonstrated that increased rule length can actually decrease the information gained due to the decreased generality of the patterns (Cheng et al. 2007), as well as decreasing the interpretability of the rules (Freitas 2013, Huysmans et al. 2011, Letham et al. 2013, Vellido et al. 2012). Note that the data mining processes used to generate $\mathcal{Z}_1$ and $\mathcal{Z}_2$ might have different k's, denoted $k_{\mathcal{Z}_1}$ and $k_{\mathcal{Z}_2}$, but the combinations possible with a smaller k are a subset of a larger k's combinations. We can write the

[2] Theoretically it can go lower, but then the model is providing no benefit to the user and is actually causing harm – it is worse than a random guess.

[3] We subtract one from $2^{|\mathcal{A}|}$ because an empty $\mathcal{X}_i$ is not a legal antecedent.

[4] Note that $|\mathcal{S} \cup \mathcal{T}| \leq |\mathcal{S}| + |\mathcal{T}|$.

number of combinations possible in a scenario with constraint k as $\sum_{i=1}^{k}\binom{|\mathcal{A}|}{i}$, where $\binom{|\mathcal{A}|}{i}$ is the binomial coefficient and equals $|\mathcal{A}|!/i!(|\mathcal{A}|-i)!$.

Strictly speaking, the lower bound of our measure is

$$J(\mathcal{S},\mathcal{T}) \geq max(\frac{|\mathcal{S}|+|\mathcal{T}|-|\mathcal{Y}|\cdot\sum_{i=1}^{k}\binom{|\mathcal{A}|}{i}}{|\mathcal{Y}|\cdot\sum_{i=1}^{k}\binom{|\mathcal{A}|}{i}},\ 0)\ , \tag{5}$$

where $k = max(k_{\mathcal{Z}_1}, k_{\mathcal{Z}_2}) \leq |\mathcal{A}|$. If $k = |\mathcal{A}|$ (i.e. No constraint was put on the length of the rules),

$$|\mathcal{Y}|\cdot\sum_{i=1}^{|\mathcal{A}|}\binom{|\mathcal{A}|}{i} = |\mathcal{Y}|(2^{|\mathcal{A}|}-1)\ . \tag{6}$$

To use some example numbers, if $|\mathcal{S}|+|\mathcal{T}| = 100$ and there are five attributes and two class values, there are $2(2^5-1) = 62$ possible combinations of attributes and class values. The lower bound then becomes 0.61, ruling out over half the original range. We reiterate that in most situations, the lower bound will simply equal 0. These simple upper and lower bounds allow the user to interpret the measure's results with far more insight than would be possible if the bounds were ignored.

3.3.2 Metric properties

Metrics are a subset of measures, defined by four mathematical properties they possess: non-negativity; identity of indiscernibles; symmetry; and triangle inequality. We can describe each of these, respectively, with the Jaccard Index: $J(\mathcal{S},\mathcal{T}) \geq 0$; $J(\mathcal{S},\mathcal{T}) = 1 \iff \mathcal{S} = \mathcal{T}$; $J(\mathcal{S},\mathcal{T}) = J(\mathcal{T},\mathcal{S})$; and $J(\mathcal{S},\mathcal{U}) \geq J(\mathcal{S},\mathcal{T}) + J(\mathcal{T},\mathcal{U})$.[5] It is straightforward to see how the Jaccard Index satisfies the first three properties, since its maximum bounds are $0 \leq J(\mathcal{S},\mathcal{T}) \leq 1$ and Equation 1 does not change if $\mathcal{S}$ and $\mathcal{T}$ are swapped. A proof of the Jaccard Index satisfying the triangle inequality has been previously constructed (Levandowsky & Winter 1971, Lipkus 1999). Because the Jaccard Index has these properties, mathematicians can use them when constructing proofs and can use more assumptions that are guaranteed to hold true. The triangle inequality especially is well-known as a strong mathematical property when analyzing metrics (Chawla et al. 2005) and designing efficient data structures and algorithms (Meila 2007). Measures that are not metrics often output results that require unintuitive interpretations (Meila 2007, Willmott et al. 2009).

4 Example Scenario: Comparing Classifiers

In this section we perform a short case study on one of the example scenarios given in Section 1.1: comparing classifiers with different parameters. Specifically, we use the implementation of the CART classifier (Breiman et al. 1984) found in the scikit-learn software (Pedregosa et al. 2011), and we compare two different objective functions for the splitting criteria: the Gini Index (Breiman et al. 1984) and Information Gain (Quinlan 1996). This comparison represents a reasonably straight-forward scenario where a common question is being asked: "How different would our results be if we changed classifier?". By using single Decision Trees (rather than Decision Forests or other classifiers), we generate a manageable number of rules directly from the classifier[6], rather than needing to filter a larger number of rules down using additional processes. By using a simple experimental set-up, our results are easily reproducible.

We apply the CART classifier on 19 datasets from the UCI Machine Learning Repository (Bache & Lichman 2013) using five-fold cross-validation with stratified folds[7], and measure the Jaccard Index between CART with the Gini Index (referred to as CART-G) and CART with Information Gain (referred to as CART-I). We also measure the Prediction Accuracy of both CART-G and CART-I on the test data from the unused fold. Our aim is to see whether the user could learn anything new by going beyond a simple comparison of Prediction Accuracies and also comparing the classifiers with the Jaccard Index.

For each dataset in our experiments, the minimum support threshold of each leaf in the Decision Trees is 2% of the records. The lower bound of the Jaccard Index is 0 for all datasets. The lower bound for Prediction Accuracy depends on the relative frequency of the majority class label in each dataset. The upper bound for the Jaccard Index depends on the number of rules found with each classifier and is reported along with the Jaccard Index results in Table 3. Since our question is "How different would our results be if we changed classifier?", we report the absolute difference between the Prediction Accuracy of CART-G and CART-I, also in Table 3. We display the same information as a scatter plot in Figure 1, where we graph the Jaccard Index and Prediction Accuracy difference for each dataset as a point in 2D space.

We can see that the detected differences in Prediction Accuracy are small – too small to trust that the detected difference is solely due to the classifiers (Hand 2006), or that the result is precise enough to remain consistent after repeated runs of the experiment (Fletcher & Islam 2014). If a user found themselves in the scenario described above, the only thing they could reasonably conclude from the Prediction Accuracy results is "the two classifiers are very similar". However the Jaccard Index results inform the user that this is not true at all – the rules uncovered by the two classifiers are similar for some datasets, but on others they can have very few rules in common. It turns out that while there is almost no difference in Prediction Accuracy when choosing either the Gini Index or Information Gain, the structure of the trees can be very different for some datasets! The Jaccard Index allows the user to learn information such as "only 40% of the rules in the Adult dataset found by either classifier were found by both classifiers", while Prediction Accuracy tells the user "for the Adult dataset, the two classifiers differ by only a tenth of a percent when predicting the consequent of unseen records". Our proposed measure does not replace Prediction Accuracy, but instead provides information that Prediction Accuracy (or any other measure, to the best of our knowledge) is incapable of providing.

[5] Note that the Jaccard Index is traditionally a metric of similarity, not distance. If a distance is preferred, the Jaccard Distance can be very easily used: $d_J(\mathcal{S},\mathcal{T}) = 1 - J(\mathcal{S},\mathcal{T})$.

[6] In a Decision Tree, each path from the root to a leaf is considered a rule, predicting the most common class value found in the leaf.

[7] Stratified folds have the same distribution of class values as the dataset as a whole.

Table 3: The Jaccard Index and Prediction Accuracy Difference between CART-G and CART-I for 19 datasets.

Dataset	Jaccard Index	Jaccard Upper Bound	Prediction Accuracy Difference
WBC	0.077	$6/8 = 0.750$	0.007
Vehicle	0.184	$21/24 = 0.875$	0.011
Banknotes	0.750	$7/7 = 1.000$	0.027
RedWine	0.125	$21/24 = 0.875$	0.013
Spambase	0.100	$21/23 = 0.913$	0.008
PageBlocks	0.546	$14/16 = 0.889$	0.004
OptDigits	0.019	$26/29 = 0.897$	0.022
PenWritten	0.000	$21/24 = 0.875$	0.020
GammaTele	0.310	$18/20 = 0.900$	0.005
Shuttle	0.714	$6/6 = 1.000$	0.000
Credit	0.136	$12/13 = 0.920$	0.014
Yeast	0.364	$14/16 = 0.875$	0.004
Cardio	0.143	$7/9 = 0.778$	0.007
Adult	0.400	$17/18 = 0.944$	0.001
Bank	0.172	$17/17 = 1.000$	0.008
TicTacToe	0.161	$17/19 = 0.895$	0.004
Car	0.400	$6/8 = 0.750$	0.028
Nursery	0.615	$10/11 = 0.909$	0.000
Chess	0.476	$14/17 = 0.824$	0.005

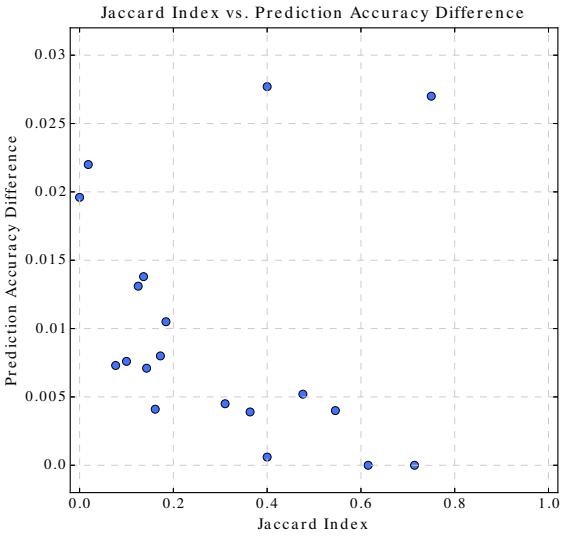


Figure 1: The Jaccard Index compared to the Prediction Accuracy difference for each dataset. Note the scope of the axes – the maximum Prediction Accuracy difference is small, while the Jaccard Index results make use of the full range between the bounds.

5 Conclusion

In this paper we propose a method for measuring the similarity between two lists of rules. The method was designed with a focus on conceptual simplicity, computational simplicity, interpretable results, and applicability in a wide variety of scenarios. One such scenario was explored in detail to demonstrate the information a user could learn from our proposed measure. Strong mathematical properties are provided to aid users in interpreting their results and making comparisons between results. For example if a third classifier was added to the scenario portrayed in Section 4, the triangle inequality allows users to use their intuition that the similarity between classifier A and C cannot be lower than the sum of the similarities between classifiers A and B and classifiers B and C.

Our use of the Jaccard Index successfully measures an aspect of data mining results that no pre-existing measure is able to do. As the Data Mining community continues to put more focus on the discovery of interpretable patterns in data, the ability to distinguish between different sets of patterns will be highly useful.

References

Agrawal, D. & Aggarwal, C. (2001), On the design and quantification of privacy preserving data mining algorithms, *in* 'Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems.', ACM, pp. 247–255.

Aodha, O. M., Stathopoulos, V., Terry, M., Jones, K. E., Brostow, G. J. & Girolami, M. (2014), Putting the Scientist in the Loop - Accelerating Scientific Progress with Interactive Machine Learning, *in* 'Proceedings of the 22nd International Conference on Pattern Recognition', IEEE, Stockholm, Sweden, pp. 9–17.

Bache, K. & Lichman, M. (2013), 'UCI Machine

Learning Repository'.
URL: *http://archive.ics.uci.edu/ml/*

Baron, S., Spiliopoulou, M. & Günther, O. (2003), Efficient monitoring of patterns in data mining environments, *in* '7th East-European Conference on Advances in Databases and Informations Systems', Springer Berlin Heidelberg, Dresden, Germany, pp. 253–265.

Breiman, L. (2001*a*), 'Random forests', *Machine learning* **45**(1), 5–32.

Breiman, L. (2001*b*), 'Statistical Modeling: The Two Cultures', *Statistical Science* **16**(3), 199–231.

Breiman, L., Friedman, J., Stone, C. & Olshen, R. (1984), *Classification and regression trees*, Chapman & Hall/CRC.

Caruana, R. & Niculescu-Mizil, A. (2004), Data Mining in Metric Space: An Empirical Analysis of Supervised Learning Performance Criteria, *in* 'Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, Seattle, Washington, pp. 69–78.

Chaudhuri, K., Monteleoni, C. & Sarwate, A. (2011), 'Differentially private empirical risk minimization', *The Journal of Machine Learning Research* **12**(1), 1069–1109.

Chawla, S., Dwork, C. & McSherry, F. (2005), Toward privacy in public databases, *in* 'Theory of Cryptography', pp. 363–385.

Cheng, H., Yan, X., Han, J. & Hsu, C.-W. (2007), Discriminative Frequent Pattern Analysis for Effective Classification, *in* '2007 IEEE 23rd International Conference on Data Engineering', IEEE, pp. 716–725.

Dwork, C. (2008), Differential Privacy: A survey of results, *in* 'Theory and Applications of Models of Computation', Springer Berlin Heidelberg, Xi'an, China, pp. 1–19.

Felkin, M. (2007), Comparing classification results between n-ary and binary problems, *in* 'Quality Measures in Data Mining', Springer Berlin Heidelberg, chapter 12, pp. 277–301.

Fletcher, S. & Islam, M. Z. (2014), Quality evaluation of an anonymized dataset, *in* '22nd International Conference on Pattern Recognition', IEEE, Stockholm, Sweden, pp. 3594–3599.

Fletcher, S. & Islam, M. Z. (2015*a*), A Differentially Private Decision Forest, *in* 'Proceedings of the 13th Australasian Data Mining Conference', Conferences in Research and Practice in Information Technology, Sydney, Australia, pp. 1–10.

Fletcher, S. & Islam, M. Z. (2015*b*), 'Measuring Information Quality for Privacy Preserving Data Mining', *International Journal of Computer Theory and Engineering* **7**(1), 21–28.

Freitas, A. (2013), 'Comprehensible classification models: A position paper', *ACM SIGKDD Explorations Newsletter* **15**(1), 1–10.

Friedman, A. & Schuster, A. (2010), Data Mining with Differential Privacy, *in* '16th SIGKDD Conference on Knowledge Discovery and Data Mining', ACM, Washington, DC, USA, pp. 493–502.

Fung, B., Wang, K., Chen, R. & Yu, P. (2010), 'Privacy-preserving data publishing: A survey of recent developments', *ACM Computing Surveys* **42**(4), 1–53.

Fung, B., Wang, K. & Yu, P. (2005), Top-down specialization for information and privacy preservation, *in* 'Proceedings of the 21st International Conference on Data Engineering', IEEE, pp. 205–216.

Geng, L. & Hamilton, H. J. (2006), 'Interestingness measures for data mining: a survey', *ACM Computing Surveys* **38**(3), 1–32.

Han, J., Cheng, H., Xin, D. & Yan, X. (2007), 'Frequent pattern mining: current status and future directions', *Data Mining and Knowledge Discovery* **15**(1), 55–86.

Han, J., Kamber, M. & Pei, J. (2006), *Data mining: concepts and techniques*, Morgan Kaufmann Publishers.

Han, J., Pei, J. & Yin, Y. (2000), Mining frequent patterns without candidate generation, *in* 'ACM SIGMOD Record', Vol. 29, ACM, Dallas, Texas, pp. 1–12.

Hand, D. J. (2006), 'Classifier Technology and the Illusion of Progress', *Statistical Science* **21**(1), 1–14.

Huysmans, J., Dejaeger, K., Mues, C., Vanthienen, J. & Baesens, B. (2011), 'An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models', *Decision Support Systems* **51**(1), 141–154.

Islam, M. Z., Barnaghi, P. & Brankovic, L. (2003), Measuring Data Quality: Predictive Accuracy vs. Similarity of Decision Trees, *in* 'Proceedings of the 6th International Conference on Computer & Information Technology', Vol. 2, Dhaka, Bangladesh, pp. 457–462.

Islam, M. Z. & Brankovic, L. (2011), 'Privacy preserving data mining: A noise addition framework using a novel clustering technique', *Knowledge-Based Systems* **24**(8), 1214–1223.

Islam, M. Z. & Giggins, H. (2011), Knowledge discovery through SysFor: a systematically developed forest of multiple decision trees, *in* 'Ninth Australasian Data Mining Conference-Volume 121', Australian Computer Society, Inc., Ballarat, Australia, pp. 195–204.

Jaccard, P. (1901), 'Etude comparative de la distribution florale dans une portion des Alpes et du Jura', *Bulletin de la Société Vaudoise des Sciences Naturelles* **37**, 547–579.

Kifer, D. & Gehrke, J. (2006), Injecting utility into anonymized datasets, *in* 'Proceedings of the 2006 ACM SIGMOD international conference on Management of data', ACM, New York, New York, USA, pp. 217–228.

Kotsiantis, S. & Kanellopoulos, D. (2006), 'Discretization techniques: A recent survey', *GESTS International Transactions on Computer Science and Engineering* **32**(1), 47–58.

Letham, B., Rudin, C., Mccormick, T. H. & Madigan, D. (2013), Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model, Technical Report 609, University of Washington.

Levandowsky, M. & Winter, D. (1971), 'Distance between sets', *Nature* **234**(5323), 34–35.

Lipkus, A. (1999), 'A proof of the triangle inequality for the Tanimoto distance', *Journal of Mathematical Chemistry* **26**(1-3), 263–265.

Meila, M. (2007), 'Comparing clusterings-an information based distance', *Journal of Multivariate Analysis* **98**, 873–895.

Mohammed, N., Chen, R., Fung, B. C. & Yu, P. S. (2011), Differentially private data release for data mining, *in* 'Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '11', ACM Press, New York, New York, USA, pp. 493–501.

Nergiz, M. E. & Clifton, C. (2007), 'Thoughts on k-anonymization', *Data & Knowledge Engineering* **63**(3), 622–645.

Ordonez, C. & Zhao, K. (2011), 'Evaluating association rules and decision trees to predict multiple target attributes', *Intelligent Data Analysis* **15**, 173–192.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, E. (2011), 'Scikit-learn: Machine Learning in Python', *Journal of Machine Learning Research* **12**, 2825–2830.

Quinlan, J. R. (1993), *C4.5: programs for machine learning*, 1st edn, Morgan kaufmann.

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

Shotton, J., Sharp, T., Kohli, P., Nowozin, S., Winn, J. & Criminisi, A. (2013), Decision Jungles: Compact and Rich Models for Classification, *in* 'Advances in Neural Information Processing Systems', pp. 234–242.

Sokolova, M. & Lapalme, G. (2009), 'A systematic analysis of performance measures for classification tasks', *Information Processing & Management* **45**(4), 427–437.

Vellido, A., Martin-Guerroro, J. D. & Lisboa, P. J. (2012), Making machine learning models interpretable, *in* 'European Symposium on Artificial Neural Networks', Bruges, Belgium, pp. 163–172.

Wagstaff, K. L. (2012), Machine Learning that Matters, *in* 'Proceedings of the 29th International Conference on Machine Learning', Edinburgh, Scotland.

Webb, G. & Brain, D. (2002), Generality is predictive of prediction accuracy, *in* 'Proceedings of the 2002 Pacific Rim Knowledge Acquisition Workshop (PKAW 2002)', pp. 117–130.

Willmott, C. J. (1982), 'Some comments on the evaluation of model performance', *Bulletin of the American Meteorological Society* **63**(11), 1309–1313.

Willmott, C. J., Matsuura, K. & Robeson, S. M. (2009), 'Ambiguities inherent in sums-of-squares-based error statistics', *Atmospheric Environment* **43**(3), 749–752.

Tackling Imbalanced Data Sets For Sequential Feature Explanation Generation Using Cost Sensitive Learning And Sampling

Tshepiso Mokoena[1] **Vukosi Marivate**[1] **Turgay Celik**[2]

[1]Modelling and Digital Science,
Council for Scientific and Industrial Research(CSIR),
Meiring Naude Road, Brummeria, Pretoria, South Africa
Email: {tmokoena1,vmarivate}@csir.co.za

[2]School of Computer Science and Applied Mathematics,
University of the Witwatersrand,
Braamfontein, Johannesburg, 2000, South Africa

Abstract

In anomaly detection applications, human analysts are usually required to manually investigate all of the data points that were detected as anomalies by the anomaly detector in order to determine whether they are truly anomalies or not. However, existing anomaly detection methods provide no additional information to the analysts about why detected data points were are flagged as anomalous. The analysts are then forced to manually browse through the feature space of the detected data points in order to identify the subset of features that are responsible for the detection in order to make their decision. A sequential feature explanation(SFE) of a detected data point is an ordered sequence of features which are presented to the analysts one at a time until the information contained in the set of already presented features is enough for the analysts to make a decision. However, anomalies are far fewer than the normal data points, therefore generating SFEs that will work with any existing anomaly detector using supervised feature selection based approaches will lead to SFEs that are biased towards the normal data points. In this paper, we will generate SFEs that will work with any existing anomaly detector and tackle the problem of using an imbalanced data set to generate these SFEs by (*i*) using a cost-sensitive model and (*ii*) using sampling to obtain a balanced sample. Lastly (*iii*), we will compare and benchmark all of the SFEs generated in (*i*) and (*ii*) on different anomaly detection data sets. Our results show that the SFEs generated using the cost-sensitive model outperform the SFEs generated using sampling.

Keywords: Anomaly explanations, Sequential feature explanations, Outlier explanations, Outlier interpretation

1 Introduction

Anomaly detection is defined as the problem of identifying anomalies in a data set by a process that is distinct from the process generating the "normal" data points in the data set (Siddiqui et al. 2015). In many applications, a human analyst is required to manually investigate all of the data points that were detected as anomalies by the anomaly detector. The analyst typically investigates the anomalies in order to decide which ones are truly anomalies and deserve further action to be taken (Siddiqui et al. 2015). However, existing anomaly detection methods produce outputs in the form of labels or scores. Methods that assign labels assign a label of the data point either being "normal" or "anomalous". Whereas methods that assign scores, assign a real number to the data points which reflects the extent to which the data points deviates from the rest of the data (Chandola et al. 2009, Micenková et al. 2013). However, these outputs provide the analysts with no further explanations about what caused the detected data points to be considered as anomalies by the anomaly detector.

Given a data point $\mathbf{a} \in \mathbb{R}^m$ that was detected as an anomaly, the analyst has to manually browse through the feature space of $\mathbf{a}$ to find the subset of features that caused $\mathbf{a}$ to deviate from the normal data points. However, identifying this subset of features could be challenging as there are $2^m - 1$ possible subsets in the feature space of $\mathbf{a}$. The subset of features that explain why the data point $\mathbf{a}$ was detected as an anomaly are called anomaly explanations. Anomaly explanations are therefore required to aid any existing anomaly detection algorithms in order for them to be more usable to human analysts in practice (Micenková et al. 2013).

An anomaly explanation called a sequential feature explanation(SFE) for a data point $\mathbf{a}$ is an ordered sequence of features, where the order of the features indicates their importance with respect to explaining why the data point could be anomalous (Siddiqui et al. 2015). Given an SFE for a detected anomaly point $\mathbf{a}$, the point $\mathbf{a}$ is incrementally presented to the analyst by first presenting the first feature of $\mathbf{a}$ in the SFE. If the analyst can make a decision that the detected data point $\mathbf{a}$ is anomalous with only that one feature presented, then the analyst's job is done for that particular data point. If the analyst needs more information, then the next feature in the SFE is presented. Now the analyst has seen the two most important features that explain how the detected data point $\mathbf{a}$ deviates from the normal data points. This process of incrementally adding features to the set of features already presented to the analyst will continue until the analyst is able to make a decision about whether the detected data point $\mathbf{a}$ is truly

anomalous or not. However, because anomalous data points are usually far fewer compared to normal data points, generating SFEs that will work with any existing anomaly detection algorithm using supervised feature selection based approaches will result in SFEs that are biased towards the normal data points.

In this paper, we will generate SFEs that will work with any existing anomaly detector and tackle the problem of using an imbalanced data set to generate these SFEs by (*i*) using a cost-sensitive model and (*ii*) using sampling to obtain a balanced sample. Lastly (*iii*), we will compare and benchmark all of the SFEs generated in (*i*) and (*ii*) on different anomaly detection data sets.

The rest of the paper is organised as follows. Section 2 gives a brief description of related works, Section 3 describes the sampling framework we will use to obtain a balanced sample in order to generate SFEs, Section 4 describes the SFE methods we will use to generate SFEs in our experiments, Section 5 describes the cost sensitive weighted random forests model and how we will use it to predict class probabilities, Section 6 gives a summary of the data sets we will use in our experiments, Section 7 details the evaluation method we will use to evaluate our SFEs, Section 8 gives a comprehensive analysis of our experiments and results, Section 9 presents a discussion about our results, Section 10 describes the future work that needs to be done in order to improve the current work on SFEs and lastly in Section 11 we conclude the paper.

2 Related Works

Most of the work in the anomaly detection community has been focused on the detection part (Vinh et al. 2016). Anomaly explanations have received relatively little attention compared to the detection. Dang et al. (2014) and Dang et al. (2013) both introduce anomaly detection methods that provide anomaly explanations too. However, the explanations that were developed in these works are specific to their anomaly detectors. Whereas in our work, we want to develop anomaly explanations that will work with any anomaly detector.

Siddiqui et al. (2015) were the first to introduce the concept of an SFE. The goal of the SFE is to minimise the number of features that must be presented to the analysts until they are able to make a judgement about whether the detected data points are truly anomalous or not. Siddiqui et al. (2015) used Marginal and Dropout methods to generate the SFEs for all the detected anomalies by the anomaly detector. These methods assumed that the normal data points are generated according to a Bayesian classifier f. The Marginal and Dropout methods make use of filter and wrapper based feature selection algorithms (Chandrashekar & Sahin 2014) to generate the SFEs based on the predictions made by f. They used random forests(RF) (Breiman 2001) in their experiments to build f as a probabilistic model over the features to predict the probability of normality of the data points given the subset of features presented to the analysts. However, RFs are known to give poor results when a highly imbalanced data set is used (Chen et al. 2004). Therefore, the drawback in their experiments is that they used imbalanced data sets to generate their SFEs.

Micenková et al. (2013) try to identify the subset of features where the anomaly deviates from the rest of the data. They introduce a sampling framework that enables them to generate anomaly explanations that are not biased towards the normal data points. For each anomaly **a**, a positive class of size $2k$ is constructed containing **a** and $2k-1$ synthetic samples similar to **a**. The negative class is constructed by selecting the *k-nearest neighbours* of **a** and k other random data points from rest of the data set. The two classes are constructed in such a way that they are of the same size in order to avoid having an imbalanced data set. They then used supervised feature selection algorithms to select the subset of features that separate the two classes. This subset of features is interpreted as the anomaly explanation of the data point **a**. One of the drawbacks of their approach is that their sampling framework is independent of the anomaly detectors outputs since they compare every anomaly with the rest of the other data points in the data set regardless of whether they are normal or anomalous data points. In addition, the anomaly explanations are not SFEs because they do not consider revealing the features to the analyst sequentially in their order of importance.

In our work, we will firstly improve the sampling framework introduced by Micenková et al. (2013) so that it generates anomaly explanations that are SFEs and dependent on the anomaly detector's outputs. We will do this by comparing every detected anomaly data point with only the normal data points declared by the anomaly detector. To generate the SFEs, we will sequentially present the features that separate the negative and positive classes in their order of importance to the analyst. Secondly, by using the improved sampling framework, we will generate multiple different SFEs by using the wrapper and embedded based feature selection algorithms (Chandrashekar & Sahin 2014) to select the features that separate the anomalous and normal classes. In addition, we will generate the Marginal and Dropout SFEs under the improved sampling framework by using RFs because the data set will now balanced. Lastly, we will improve the work done by Siddiqui et al. (2015) by generating the Marginal and Dropout SFEs using weighted random forests(WRF) (Chen et al. 2004) which use cost sensitive learning (Liu & Zhou 2006) to solve the problem of having an imbalanced data set. This will ensure that the SFEs generated by the Marginal and Dropout methods are not biased towards the normal data points that were declared by the anomaly detector.

3 Sampling framework for SFE generation

3.1 Introduction

In this section, we will describe the sampling framework we will use to turn the SFE generation problem into a classical supervised feature selection problem. This sampling framework is derived from the work done by Micenková et al. (2013). It will enable us to use various feature selection methods to generate multiple SFEs that are dependent on the anomaly detector's outputs.

Let D represent a labelled data set from the anomaly detector. Let the set $A \subset D$ contain all the detected anomaly data points and $D^{Normal} \subset D$ represent the

normal data points declared by the anomaly detector.

We will first define the definitions of *k-distance* and *reference set*, adapted from the work in (Micenková et al. 2013):

Definition 3.1. The *k-distance* of a data point $\mathbf{p} \in D$, denoted by *k-distance*($\mathbf{p}$), is the distance $d(\mathbf{p}, \mathbf{p}')$ between $\mathbf{p}$ and its k-th nearest neighbour $\mathbf{p}'$

Definition 3.2. The *reference set* of a data point $\mathbf{p}$, denoted by $R_k(\mathbf{p})$, is the set of points $\mathbf{x}$ whose distance from $\mathbf{p}$ is less than or equal to *k-distance*:

$$R_k(\mathbf{p}) = \{\mathbf{x} \in D \setminus \{\mathbf{p}\} | d(\mathbf{p}, \mathbf{x}) \leq \textit{k-distance}(\mathbf{p})\}$$

3.2 Sampling framework

For each anomaly $\mathbf{a} = (x_1, \ldots, x_m) \in A$, we will follow the following steps to generate its corresponding SFE:

1. Let $S = \phi$ denote the set of selected feature indices and $V = \{1, \ldots, m\}$ denote the set of unselected feature indices.
2. We will define the sampled inlier class of $\mathbf{a}$ from D^{Normal} as a union of its reference set $R_k(\mathbf{a})$ and a set of randomly drawn data points from D^{Normal} as:

   $$\mathcal{I}_{\mathbf{a}} = R_k(\mathbf{a}) \cup \{\mathbf{q}_j\}_1^r$$

   where $\mathbf{q}_j \in D^{Normal} \setminus R_k(\mathbf{a})$, and $r = |R_k(\mathbf{a})|$
3. We will define our outlier class of $\mathbf{a}$ as:

   $$\mathcal{O}_{\mathbf{a}} = \{\mathbf{a}\} \cup \{\mathbf{z}_j\}_1^s$$

   where $\mathbf{z}_j \sim \mathcal{N}(\mathbf{a}, \lambda^2 I)$, $s = |\mathcal{I}_{\mathbf{a}}|$, I is the $m * m$ identity matrix and $\lambda = \alpha \cdot \frac{1}{\sqrt{m}} \cdot \textit{k-distance}(\mathbf{a})$, where α is a user defined parameter which controls the width of the distribution.
4. We use the selected feature selection algorithm to generate the next best feature to be added into the set S.
5. If the analyst has seen enough features to be able to make a decision or $V = \phi$, then we will stop and set the SFE of $\mathbf{a}$ to S , otherwise go back to step 4.

4 SFE Methods

4.1 Introduction

In this section, we will define the SFEs that we will use in our experiments. The notations and definitions defined below are important as they will be used throughout the rest of the paper.

- $SFE(\mathbf{x})$ represents the set of feature indices in the SFE list of $\mathbf{x}$
- $SFE(\mathbf{x})_k$ represents the k^{th} element in $SFE(\mathbf{x})$ which is the k^{th} most important feature
- $SFE(\mathbf{x})^k$ represents the first k features in $SFE(\mathbf{x})$
- Let S be any set of the feature indices. Then we denote x_S as the projection of the data point $\mathbf{x}$ onto the subspace specified by S, while $\mathbf{x}$ represents the data point in the full feature space.
- Let the normal data points be generated according to a Bayesian classifier f. f is a probabilistic model over the features and is used to predict the probability of normality of the data points given the feature subset presented to the analyst

4.2 Marginal Methods

4.2.1 Independent marginal (SFE_{IM})

This approach requires the computation of individual marginals $f(x_i)$ for $i = 1, \ldots, n$. Then $SFE(\mathbf{x})$ is obtained by sorting the features in increasing order of $f(x_i)$ (Siddiqui et al. 2015).

4.2.2 Sequential marginal (SFE_{SM})

This method starts with an empty set and at each iteration adds the feature that minimises the joint marginal density of f with the previously selected features. More formally, SFE_{SM} computes the following explanation:

$$SFE_{SM} : SFE(\mathbf{x})_i = arg \min_{j \in \overline{SFE}(\mathbf{x})^{i-1}} f(x_{SFE(\mathbf{x})^{i-1}}, x_j),$$

where $\overline{SFE}(\mathbf{x})^{i-1}$ is the complement and represents all the feature indices not included in $SFE(\mathbf{x})^{i-1}$ (Siddiqui et al. 2015).

4.3 Dropout Methods

4.3.1 Independent dropout(SFE_{ID})

In this method each feature x_i is assigned a score of $f(\mathbf{x} - x_i) - f(\mathbf{x})$, where we denote the removal of feature x_i from $\mathbf{x}$ by $\mathbf{x} - x_i$ (Siddiqui et al. 2015). Features with larger scores are the ones that make the point appear the most normal when they are removed. The SFE is then obtained by sorting the features in decreasing order of their scores.

4.3.2 Sequential dropout (SFE_{SD})

A sequential version of the dropout method is given by the following explanation (Siddiqui et al. 2015):

$$SFE_{SD} : SFE(\mathbf{x})_i = arg \max_{j \in \overline{SFE}(\mathbf{x})^{i-1}} f(x_{\overline{SFE}(\mathbf{x})^{i-1} - x_j})$$

4.4 Forward selection svm SFE(SFE_{FS})

Forward selection support vector machines(FS-SVM) (Chandrashekar & Sahin 2014) is a wrapper based feature selection algorithm that greedily incorporates features into SFE_{FS}, while being navigated by the classification accuracy of the support vector machine(SVM) (Guyon et al. 2002) of the training data (Micenková et al. 2013). The algorithm starts with an empty set and at each iteration adds the feature that maximises the classification accuracy of the SVM with the current subset of already chosen features. This process of incrementally adding features is repeated until the required number of features are added into SFE_{FS}.

4.5 Backward selection svm SFE(SFE_{BS})

Backward selection support vector machines(BS-SVM) (Chandrashekar & Sahin 2014) is similar to FS-SVM. Unlike FS-SVM, BS-SVM starts with the complete set of features and at each iteration removes the feature whose removal gives the lowest decrease in classification accuracy of the SVM of the training data and adds it into SFE_{BS}.

4.6 SVM recursive feature elimination SFE ($SFE_{SVM-RFE}$)

SVM recursive feature elimination (SVM-RFE) (Guyon et al. 2002) is an embedded feature selection algorithm that was first proposed by Guyon et al. (2002). Its output is a ranked feature list and the top features are chosen by selecting the highest ranked features. For linear SVM-RFE the ranking criterion for feature k is $J(k) = \mathbf{w}_k^2$, where $\mathbf{w}_k$ is the k^{th} element of $\mathbf{w}$, the normal vector to the hyperplane of the SVM (Guyon et al. 2002). For each iteration of the recursive elimination a linear SVM is trained and the feature with the smallest ranking is removed because it has the least effect on the classification (Guyon et al. 2002). The remaining features are then kept for the SVM model in the next iteration. This process is continued until all the features have been removed (Guyon et al. 2002). The features are then sorted according to the order of their removal and the later a feature is removed the more important it is.

4.7 Random forests SFE(SFE_{RF})

Random forests provide us with additional information about the importance of features. For each tree in the RF, the RF algorithm estimates the importance of a feature by measuring how much the prediction error increases when the data for that feature is permuted while all the other features are left unchanged (Liaw & Wiener 2002). The importance score of a feature is then calculated by averaging the difference of the prediction error before and after the permutation over all the trees in the RF (Liaw & Wiener 2002, Menze et al. 2009). The features with the highest scores are the most important features.

4.8 Trade off between SFEs

The SFE_{ID} and SFE_{IM} methods are both filter based feature selection methods, while the SFE_{SM}, SFE_{DO}, SFE_{FS} and SFE_{BS} are wrapper based feature selection methods. SFE_{ID} and SFE_{IM} are the cheapest SFEs to compute however, they do not take feature interactions into account because each feature is assessed independently. Whereas SFE_{SM}, SFE_{DO} and SFE_{FS} take feature interactions into account but are the most expensive to compute. $SFE_{SVM-RFE}$ and SFE_{RF} are both SFEs derived from embedded feature selection methods, but unlike wrapper based methods, the search for the optimal subset of features are built into the classifiers construction and reduce the computational time taken for reclassifying feature subsets (Saeys et al. 2007, Chandrashekar & Sahin 2014). Embedded methods take feature interactions into account, while also being less computationally expensive than wrapper based methods (Saeys et al. 2007) but more expensive than filter based methods. An analyst might also take into consideration the running times required to generate the SFEs. Some SFEs might produce good results but may also take longer to compute the features in the SFEs. The analyst might then choose to trade off a good SFE for SFEs that are cheaper to generate if their results are not significantly different. However, in this paper, we chose to focus on the performances of the SFEs rather than the trade off between good results and running times required to generate the SFEs.

5 Weighted Random Forests and predicting class probabilities

Cost sensitive learning solves the problem of classifying imbalanced data sets by assigning a high cost to the misclassification of the minority class (Chen et al. 2004). RFs tend to be biased towards the majority class because they assume each class' misclassification cost is equally weighted. WFRs solve this problem by assigning a higher weight to the minority class. The class weights are incorporated into the RF model in the tree induction procedure and terminal nodes. For each tree in the RF, the class weights are used to weight the Gini criterion for finding node splits (Chen et al. 2004) during the tree induction procedure (Chen et al. 2004). While in the terminal nodes of each tree the class weights are used to determine the weighted vote for the class predictions (Chen et al. 2004). For each tree in the RF, class probabilities are estimated by taking the relative frequency of the class of interest in that terminal node (Dankowski & Ziegler 2016). The probability of a data point falling in a certain class in the RF is therefore obtained by averaging the relative class frequencies over all the trees in the RF (Bostrom 2007).

6 Data and Data Processing

We will use the data sets provided by Goldstein & Uchida (2016) to generate our SFEs and to also compare and benchmark our SFEs. Goldstein & Uchida (2016) sampled and converted data sets mostly available from the UCI machine learning repository (Lichman 2013) into unsupervised anomaly detection data sets for algorithm benchmarking. Another data set that we used that they included was the Amsterdam Library of Object Images(ALOI) data set (Geusebroek et al. 2005). These data sets all have ground truth and Table 1 contains a summary of the data sets.

Since the purpose of our research is not on anomaly detection, we will assume that the ground truth of the data comes from an unsupervised anomaly detector. Therefore we have an "anomaly detector" that has a 100% detection rate with no false detections. For each of the data sets, we used min-max normalisation to normalise each of the data points in the range of $[0, 1]$.

7 Evaluations

In order to evaluate our generated SFEs, we will simulate the human analyst as the conditional distribution of the anomaly class given the features presented to the analyst from the SFE (Siddiqui et al. 2015). Let $A(\mathbf{x}, S) = P(anomaly|S)$, which returns the probability that a data point $\mathbf{x}$ is anomalous given only the features specified by the set S. Given $SFE(\mathbf{x})$, we will generate an analyst curve that plots $A(\mathbf{x}, SFE(\mathbf{x})^k)$ vs k for $k = 1, \ldots, m$, where k is the number of features revealed to our simulated analyst. We will measure the area under the curve(AUC) of the simulated analyst curve for each detected anomaly and divide it by the total number of features so that we have an AUC that is in the range of $[0, 1]$. We will then average the analyst curves across all the detected anomalies and compare the different SFE methods based on the average AUC of these analyst curves. The SFE method with the highest average AUC will be chosen as the best per-

Data set name	Size	Dimensions	Anomalies(%)
Landsat Satellite	5100	36	1.49
Statlog Shuttle	46464	9	1.89
Thyroid Disease	6916	21	3.61
ALOI	50000	27	3.02

Table 1: Data set summary

Data set name	IM	SM	ID	SD
Thyroid Disease	0.8972	**0.9011**	0.8817	0.8533
Landsat Satellite	**0.8562**	0.8323	0.8457	0.8106
Statlog Shuttle	0.8834	**0.8848**	0.8828	0.8827
ALOI	0.6731	**0.6855**	0.6694	0.6587

Table 2: AUCs of cost sensitive learning sfe

Data set name	*Random*	IM	SM	ID	SD	FS	BS	$SVM-RFE$	RF
Thyroid Disease	0.7462	0.8495	**0.8627**	0.8537	0.7888	0.7857	0.7395	0.8364	0.8242
Landsat Satellite	0.8016	0.8394	0.8064	0.8346	0.7976	0.8224	0.8013	**0.8414**	0.8407
Statlog Shuttle	0.8543	**0.8839**	0.8830	**0.8839**	0.8800	0.8810	0.8812	0.8810	0.8834
ALOI	0.6529	0.6701	0.6684	0.6696	0.6593	0.6682	0.6337	**0.6760**	0.6639

Table 3: AUCs of sampling framework SFEs

forming SFE. To estimate $A(\mathbf{x}, S)$, we will use the entire data set and label it with the labels provided by the anomaly detector to train the WRF's that will be used to estimate the probability of the detected anomalies being anomalous given only the features presented to the analyst.

8 Experiments and Results

All the SFEs defined in Section 4 will be generated under the sampling framework. Only the Marginal and Dropout SFEs will be generated using the cost sensitive model. For the cost sensitive Marginal and Dropout SFEs, we used WRFs to build the Bayesian classifier f, while for the Marginal and Dropout SFEs generated under the sampling framework we used regular RFs because the framework ensures that for each anomaly the classes are balanced. We ran each of our experiments 5 times and then averaged the resulting SFE analyst curves and plotted the first standard deviation from the mean on the analyst curves. Our baseline SFE is the random SFE, where the features of the anomalies detected are presented to the analyst in random order. We expect the AUC scores of the SFEs to be between the AUC score of the random SFE and 1.

8.1 Parameter setting

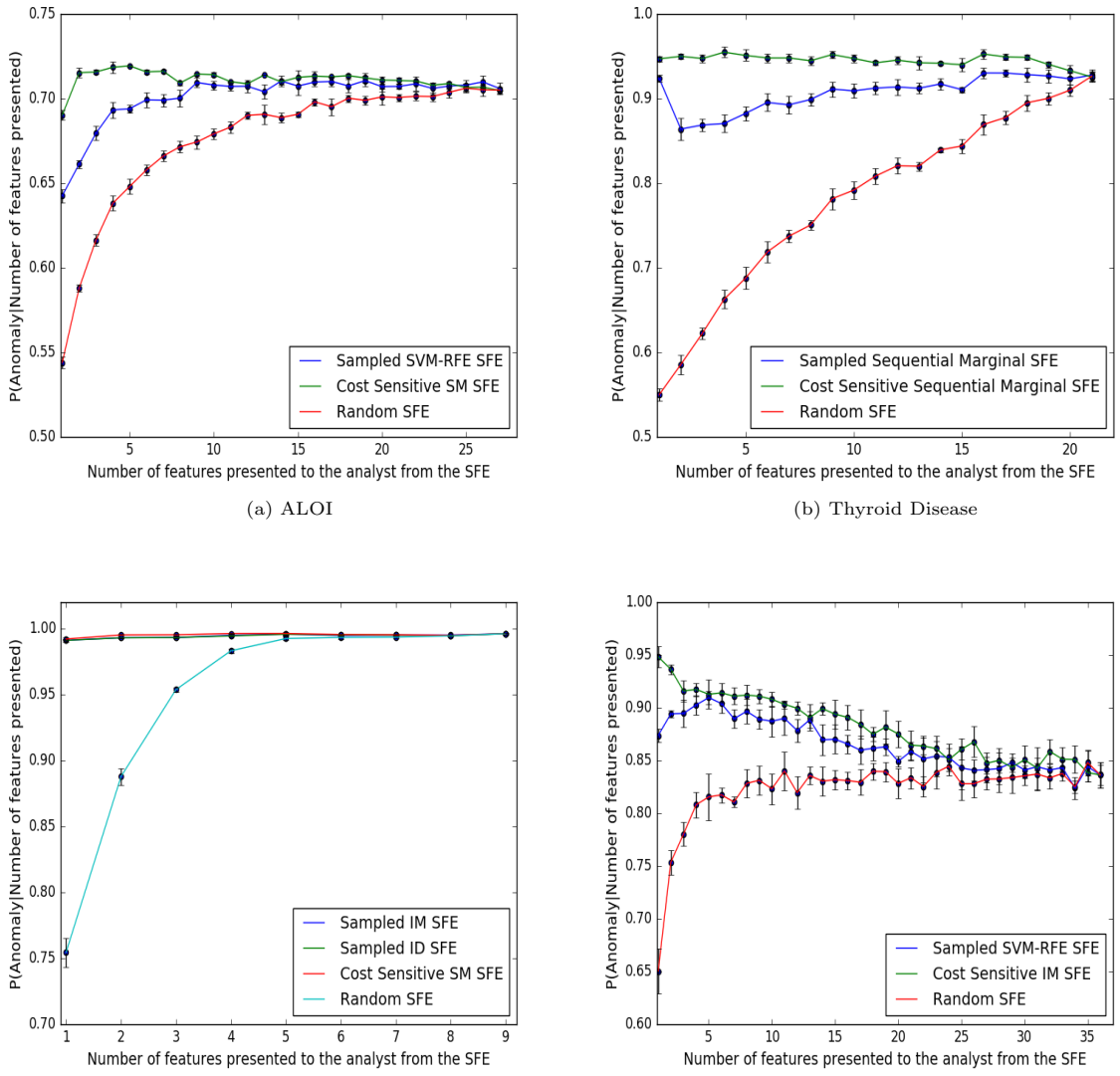
In our experiments, we computed the SFEs for each data set. For SFEs generated under the SFE sampling framework, we fixed $k = 250$ and $\alpha = 0.35$ for all of our experiments. Micenková et al. (2013) fixed $k = 35$ and $\alpha = 0.35$ for all of their experiments. We chose to use the same α value but increased k so that a bigger sample is used and therefore improve the results of the generated SFEs. For the SFEs generated using SVMs, we used a linear SVM, with $C = 1$ as specified by Micenková et al. (2013). All RFs and WRFs models were composed of 10 trees.

8.2 Analysis of analyst curves

In this section, we will analyse the results of the different SFE methods in our data sets. Even though we use the AUCs of the analyst curves to evaluate which methods perform the best, we get greater insight on the performance of the SFEs by analysing the analyst curves as well. From the analyst curves, we get information about the minimum number of features that need to be presented to the analyst so that the analyst understands why the data points were detected as anomalous by the anomaly detector. We only presented the analyst curves for the best performing SFEs so that comparisons can be made. Table 2 and Table 3 show the AUCs of the averaged analyst curves of the SFEs generated using cost sensitive learning and the sampling framework respectively. While, Figure 1 displays the averaged analyst curves.

8.2.1 ALOI data set

From Table 3, the AUC of the baseline SFE for this data set is 0.6529. Under the sampling framework $SFE_{SVM-RFE}$ has the highest AUC of 0.6760, while amongst the cost sensitive SFEs, SFE_{SM} has the highest AUC of 0.6855. The best performing SFE is the cost sensitive SFE_{SM}. Even though the AUCs of the cost sensitive SFE_{SM} and the AUC of the $SFE_{SVM-RFE}$ generated under the sampling framework are not significantly greater than the baseline SFE by value, the analyst curves of these SFEs in Figure 1(a) tell us a different story. Looking at the analyst curves, we can see that if the analyst is given all the features of the anomalies, then the probability that the detected anomalies are truly anomalous is 0.7. From the cost sensitive SFE_{SM}, we can see from the analyst curve that presenting the first two features of the SFE gives the analyst a higher probability of the detected anomalies being anomalous than presenting the analyst with the full set of features of the detected anomalies. The remaining features in the cost sensitive SFE_{SM} provide the analyst with no extra information as the analyst curve has plateaued toward 0.7 after the second feature. This plateau is caused by redundant features which provide the analyst with no extra information as the first two features in the SFE already explain the reason behind why the data points were detected as anomalies. On the other hand, $SFE_{SVM-RFE}$ generated under the sampling framework only plateaus to 0.7 after the fifth feature in the SFE, while the baseline SFE only plateaus after



(a) ALOI

(b) Thyroid Disease

(c) Statlog Shuttle

(d) Landsat Satellite

Figure 1: Analyst Curves

the eighteenth feature. Even though the AUCs give us an idea about which of the SFEs give us better performances, we see that by delving deeper into the analyst curves we will get more information about the SFEs performances overall.

8.2.2 Thyroid Disease data set

From the SFEs generated under the sampling framework, SFE_{SM} is the best performing SFE, while from the cost sensitive SFEs, SFE_{SM} is also the best performing SFE. The cost sensitive SFE SFE_{SM} has the highest AUC amongst all the SFEs. From the analyst curves in Figure 1(b), if the analyst is given all the features of the anomalies, then the probability that the detected anomalies are truly anomalous is around 0.91. Both the cost sensitive SFE_{SM} and the SFE_{SM} generated under the sampling framework only require the first feature in the SFEs to explain the why the data points were detected as anomalies. This is because, given the first feature from both SFEs, the probability that the detected anomalies appear anomalous to the analyst is greater than 0.91. The probability of the data points being anomalies given the number of features presented to the analyst in the SFE for the cost sensitive SFE_{SM} fluctuates between 0.91 and 0.95. While for the SFE_{SM} that was generated under the sampling framework, the analyst's probabilities given the number of features presented drops to below 0.9 when the second feature is presented and then it steadily increases again as more feature are presented. The reason for the drop in the probability of the second feature could be because the SFE fails to identify the optimal first feature that was identified by the cost sensitive SFE_{SM}. The baseline SFE doesn't plateau and thus requires the analyst to be shown the full set of features. Therefore these SFEs perform significantly better than the baseline SFE.

8.2.3 Statlog Shuttle data set

Both SFE_{IM} and SFE_{ID} have the same AUCs under the sampling framework, while the cost sensitive SFE_{SM} has the highest overall AUC. However we can see from Figure 1(c) that these SFEs are almost horizontal around 0.99. This shows that the SFEs were able to capture the important features where the

anomalies deviate from the normal data points. The SFEs only require the first feature to be presented to the analyst, while the baseline SFE only plateaus after the fifth feature and requires the analyst to be shown the first 5 features.

8.2.4 Landsat Satellite data set

Under the sampling framework, $SFE_{SVM-RFE}$ has the highest AUC, while SFE_{IM} has the highest AUC under the cost sensitive learning SFEs. The cost sensitive SFE_{IM} has the highest AUC amongst all of the SFE methods. From the analyst curves in Figure 1(d), we can see that if the analyst is given all the features of the anomalies, then the probability that the detected anomalies are truly anomalous is 0.83. However, we can see from the analyst curves of the SFEs that they both decreasingly fluctuate between 0.83 and 0.91 as the number of features presented to the analyst increases. This might be because the anomalies are more anomalous in the lower feature subspaces identified by the SFEs and the remaining features are both noisy and redundant and therefore they distort the analyst's judgement. For both SFEs, the analyst only needs to be shown the first feature in the SFEs in order to understand why the data points were detected as anomalies. While for the baseline SFE, the analyst must be shown at least the first 8 features in the SFE.

9 Results Discussion

Overall, our results show that the SFEs generated using the cost-sensitive model outperform the SFEs generated under the sampling framework. In particular, SFE_{SM} was the best performing cost sensitive model generated SFE. Whereas for the SFEs generated under the sampling framework, $SFE_{SVM-RFE}$ was the best performing SFE. From the AUCs of the analyst curves, it is clear that the cost sensitive Marginal and Dropout SFEs provide better SFEs than the Marginal and Dropout SFEs generated under the sampling framework. For each anomaly, the sampling framework generates synthetic data points to create the anomaly class and samples from the normal data points to create the normal class. Therefore, the sampling framework throws away valuable data points which could have been used to generate better SFEs. On the other hand, the cost sensitive SFE methods learn which features need to be added into the SFE by considering all of the normal and anomalous data points from the anomaly detector. However, it is clear from the AUCs that the sampling framework SFEs are competitive as they still provide SFEs that are significantly better than the baseline SFE.

The worst performing SFE is SFE_{BS}. It performed worse than the baseline SFE for the Thyroid Disease, Landsat Satellite and the ALOI data sets. These data sets all have more than 20 dimensions. SFE_{BS} is generated under the sampling framework and uses a wrapper based BS-SVM feature selection algorithm to greedily add features into the SFE. BS-SVM starts off with a full set of features and at each iteration uses the linear SVM to remove the feature that returns the lowest classification accuracy and adds it into the SFE. One of the reasons that could explain its bad performance is that a linear SVM kernel was used to generate the SFE. The bad results show that the decision boundary between the anomalous and the normal class in the higher feature subspaces is not linear at all because it has to linearly separate the classes in higher dimensions. SFE_{FS} also uses a linear SVM but it obtains better results because the search starts with an empty set of features and greedily adds the features that return the highest classification accuracy into the SFE, unlike SFE_{BS} which has to start with a full set of features and linearly separate the classes in higher dimensions. Therefore, it is more likely for the anomalous and normal classes to be linearly separable in the lower feature spaces, but as the number of features increase, the decision boundary becomes more complex and nonlinear.

10 Future Work

SFEs suffer from an effect known as the "nesting effect", that is, once a feature is added into the subset it cannot be removed(Lv et al. 2015). For example, the subset of the five best features chosen must contain the subset of the one, two, three and four best chosen features and so on. In practice, the five best features may not contain any of the best one, two, three or four best chosen features (Nakariyakul & Casasent 2009). In future work, we would like to develop a new anomaly explanation that doesn't contain feature subsets that are nested. This explanation must contain the best subset of features that need to be presented to the analyst sequentially. The first subset in this anomaly explanation contains only one feature, the second subset contains two features, the third subset contains three features and so on. Unlike the SFE, the second feature subset in the explanation does not need to be a subset of the first feature subset and each subsequent feature subset does not need to be a subset of the previous feature subsets in the explanation.

11 Conclusion

In this paper, our aim was to generate SFEs that will work with any existing anomaly detector and solve the problem of using an imbalanced data set to generate these SFEs by (*i*) using a cost-sensitive model and (*ii*) using sampling to obtain a balanced sample. We also (*iii*) compared and benchmarked all of the SFEs generated in (*i*) and (*ii*) on different anomaly detection data sets. We achieved (*i*) by presenting the sampling framework which enabled us to generate multiple different SFEs using supervised feature selection algorithms and (*ii*) was achieved by using WRFs to generate the Dropout and Marginal SFEs. Our results show that the SFEs generated using the cost-sensitive model outperform the SFEs generated under the sampling framework. In particular, SFE_{SM} was the best performing cost sensitive model generated SFE. Whereas for the SFEs generated under the sampling framework, $SFE_{SVM-RFE}$ was the best performing SFE. However, it is clear from the AUCs that the SFEs generated under the sampling method are competitive because they provide SFEs that are significantly better than the baseline SFEs.

References

Bostrom, H. (2007), Estimating class probabilities in random forests, *in* 'Machine Learning and Applications, 2007. ICMLA 2007. Sixth International Conference on', IEEE, pp. 211–216.

Breiman, L. (2001), 'Random forests', *Machine learning* **45**(1), 5–32.

Chandola, V., Banerjee, A. & Kumar, V. (2009), 'Anomaly detection: A survey', *ACM computing surveys (CSUR)* **41**(3), 15.

Chandrashekar, G. & Sahin, F. (2014), 'A survey on feature selection methods', *Computers & Electrical Engineering* **40**(1), 16–28.

Chen, C., Liaw, A. & Com, A. L. (2004), Using random forest to learn imbalanced data.

Dang, X. H., Assent, I., Ng, R. T., Zimek, A. & Schubert, E. (2014), Discriminative features for identifying and interpreting outliers, *in* '2014 IEEE 30th International Conference on Data Engineering', IEEE, pp. 88–99.

Dang, X. H., Micenková, B., Assent, I. & Ng, R. T. (2013), Local outlier detection with interpretation, *in* 'Joint European Conference on Machine Learning and Knowledge Discovery in Databases', Springer, pp. 304–320.

Dankowski, T. & Ziegler, A. (2016), 'Calibrating random forests for probability estimation', *Statistics in medicine* .

Geusebroek, J.-M., Burghouts, G. J. & Smeulders, A. W. (2005), 'The amsterdam library of object images', *International Journal of Computer Vision* **61**(1), 103–112.

Goldstein, M. & Uchida, S. (2016), 'A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data', *PloS one* **11**(4), e0152173.

Guyon, I., Weston, J., Barnhill, S. & Vapnik, V. (2002), 'Gene selection for cancer classification using support vector machines', *Machine learning* **46**(1-3), 389–422.

Liaw, A. & Wiener, M. (2002), 'Classification and regression by randomforest', *R news* **2**(3), 18–22.

Lichman, M. (2013), 'UCI machine learning repository'.
URL: *http://archive.ics.uci.edu/ml*

Liu, X.-Y. & Zhou, Z.-H. (2006), The influence of class imbalance on cost-sensitive learning: An empirical study, *in* 'Sixth International Conference on Data Mining (ICDM'06)', IEEE, pp. 970–974.

Lv, J., Peng, Q. & Sun, Z. (2015), A modified sequential deep floating search algorithm for feature selection, *in* 'Information and Automation, 2015 IEEE International Conference on', IEEE, pp. 2988–2993.

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

Micenková, B., Dang, X.-H., Assent, I. & Ng, R. T. (2013), Explaining outliers by subspace separability, *in* 'Data Mining (ICDM), 2013 IEEE 13th International Conference on', IEEE, pp. 518–527.

Nakariyakul, S. & Casasent, D. P. (2009), 'An improvement on floating search algorithms for feature subset selection', *Pattern Recognition* **42**(9), 1932–1940.

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

Siddiqui, M. A., Fern, A., Dietterich, T. G. & Wong, W.-K. (2015), 'Sequential feature explanations for anomaly detection', *arXiv preprint arXiv:1503.00038* .

Vinh, N. X., Chan, J., Romano, S., Bailey, J., Leckie, C., Ramamohanarao, K. & Pei, J. (2016), 'Discovering outlying aspects in large datasets', *Data Mining and Knowledge Discovery* pp. 1–36.

A Novel Technique for Integrating Monotone Domain Knowledge into the Random Forest Classifier

Chris Bartley, Wei Liu and Mark Reynolds

School of Computer Science & Software Engineering
University of Western Australia,
35 Stirling Hwy, Crawley WA 6009,
Email: christopher.bartley@research.uwa.edu.au

Abstract

In many machine learning applications there exists prior knowledge that the response variable should be increasing (or decreasing) in one or more of the features. Integrating this knowledge of 'monotone' relationships is useful because it can improve predictive performance, and satisfy user requirements. This paper presents a novel technique for incorporating monotone knowledge into Random Forest classifiers. Rather than monotonising the trees in the ensemble, we consider Random Forest as a form of weighted neighbourhood scheme and formulate an optimisation problem to minimally mutate the model to increase monotonicity. This approach has the advantage of straightforwardly incorporating *partial* monotonicity (in *some*, rather than *all*, features). We apply the new technique to real datasets and investigate the impact of monotonicity and sample size on predictive accuracy. We find that Random Forest is often very good at recognising monotonicity without modification. However, when it does fail to reflect monotone knowledge, our technique reliably and significantly increases monotonicity. In addition, the increase in monotonicity is significantly positively correlated to increases in accuracy.

Keywords: Classification, Random Forest, Tree Ensemble, Domain Knowledge

1 Introduction

Prior domain knowledge has many forms, such as knowledge of class invariance in regions of the input space (Mangasarian & Wild 2008) (e.g. if $tumor > 4 \wedge lymphnodes > 5 \implies Recurrence$), class invariance under transformations of the input (Lauer & Bloch 2008) (used mostly for image processing) and shape knowledge such as convexity (Wang & Ni 2012) and monotonicity (Chen & Li 2014, Li & Chen 2014).

We focus in this paper on *monotone* prior knowledge. Monotone knowledge is easy to obtain from experts or domain knowledge for many types of problems, and is informative without being overly prescriptive. A monotone relationship between X and Y means that an increase in X should not lead to a decrease in Y. For example, a house with three bedrooms should not be cheaper than one with two bedrooms (all other factors constant). Monotonicity between input variable x_j and output y may thus be defined as:

> *For an increase in variable x_j, variable y should not decrease (all other variables held constant).*

Monotonicity can apply to problems where the model output is ordered, such as regression and ordinal classification. This paper considers *ordinal classification*, where the task is to assign objects to two or more classes when there is an *order* assigned to the classes, but the *distances* between classes are irrelevant. Examples of ordinal classification include credit ratings (AAA, AA, A, BBB, ...), where a better rating may be considered 'higher'. Similarly a cancer diagnosis of No/Yes may be considered ordered (in terms of the risk of being cancerous).

Monotone domain knowledge integration has received significant attention in decision tree literature and numerous approaches have been developed. Ben-David (1995) provided one of the first inductive decision tree techniques that included a penalty for node splits that would increase tree monotonicity (i.e. the total-ambiguity-score to be minimised at each node included a non-monotonicity term.). Potharst & Feelders (2002) proposed an improvement by weighting the penalty by probability of occurrence. Makino et al. proposed a technique to guarantee a monotone tree with a 'direct' induction approach to ensure a monotone tree is created by adding 'corner' datapoints if required during node creation, which was limited to a binary response and required monotone training data. Potharst & Bioch (2000) extended this to multi-class classification, which however still required monotone training data. Potharst & Bioch (2000) and Potharst & Feelders (2002) also proposed a 'generate and test' technique to build trees from randomised bootstrap samples and select the best monotone one for use. Bioch & Popova (2003) made Potharst's technique applicable to non-monotone data by preprocessing (relabelling) the dataset if required to ensure monotonicity, and further examined a 'number of conflicts' splitting criterion to minimise the number of non-monotone pairs in the tree nodes. Daniels & Velikova (2003) and Velikova & Daniels (2004) proposed a combined approach of minimal data relabelling to ensure monotonicity, followed by selection of the best monotone tree from the complexity series of pruned trees. Feelders & Pardoel (2003) also proposed a minimal pruning approach to make a non-monotone tree monotone that combines with existing cost-complexity pruning approaches to create a sequence of monotone trees for evaluation against a test set, as per cost-complexity tree selection processes.

Monotone Random Forest has only received attention recently, with the proposal from González

et al. (2015). This approach combines two existing monotone techniques: (a) monotonising each component decision tree, using the approach by Ben-David (1995); and (b) pruning non-monotone trees from the ensemble. The main limitation of this approach is that it assumes monotonicity in *all* features. In many situations we have monotone knowledge about some features and do not wish to impose universal positive or negative impacts for the other features. For example, for the `Ljubljana Breast Cancer Recurrence` dataset (included in the experiments below), domain knowledge suggests the chance of recurrence is monotone *increasing* in the features for`tumor size`, `degree of malignancy`, and `number of involved nodes`, and monotone *decreasing* when `irradiation therapy` has been used or the `nodes are encapsulated`. But we do not wish to impose a direction for the remaining features related to `age`, `breast` (left/right), `breast quadrant`, or `menopause`. Yet we also do not want to throw these features out either. As it is not straightforward how the González et al. (2015) approach could be extended to partial monotonicity, we propose an approach designed specifically for partial monotonicity.

As demonstrated by (Lin & Jeon 2006), Random Forest can be considered as a *weighted neighbourhood scheme* with similarities to k-nearest neighbours. We exploit this fact to derive a secondary optimisation problem to minimally mutate this model to achieve monotone constraints, which is convex and solvable by any quadratic solver. We test two variants of the algorithm on nine real datasets, with the following key findings:

- **Random Forest Monotonicity**: Unmodified Random Forest is often very good at recognising monotone relationships.
- **Algorithm Effectiveness**: When standard Random Forest is *not* monotone, the simpler of our two algorithm variants reliably and substantially improves partial monotonicity.
- **Impact on Accuracy**: When the opportunity to increase monotonicity is present, the increase in monotonicity is statistically significantly positively correlated with increased accuracy.

We hope to stimulate interest in this area and have provided a python implementation of the algorithm on `github` (https://github.com/chriswbartley/PMRF).

This paper is organised as follows. Section 2 outlines partially monotone classification and Random Forest. Section 3 is our core contribution of a new technique for integrating monotonicity into Random Forest. Section 4 describes the experiments and Section 5 reports the results, concluding in Section 6.

2 Background

2.1 Partially Monotone Classification

The classification literature on monotonicity (Feelders 2000, Feelders & Pardoel 2003, Milstein et al. 2013, Marsala & Petturiti 2015, Potharst & Feelders 2002, Potharst & Bioch 2000, Potharst et al. 2009, Marsala & Petturiti 2015, Chen & Li 2014, Li & Chen 2014, Velikova & Daniels 2004, Duivesteijn & Feelders 2008) ubiquitously defines monotonicity in the context of the *dominance relation* $\succeq$ (or *partial ordering*), which for m-dimensional vectors $\mathbf{x}, \mathbf{x}'$ are defined as:

$$\mathbf{x} \succeq \mathbf{x}' \Leftrightarrow x_j \geq x'_j, \quad \mathbf{x}, \mathbf{x}' \in \Re^m, \forall j \in 1..m \tag{1}$$

A function $f : \Re^m \rightarrow \Re$ is defined as *monotone* if:

$$\mathbf{x} \succeq \mathbf{x}' \Rightarrow f(\mathbf{x}) \geq f(\mathbf{x}'), \ \forall \mathbf{x}, \mathbf{x}' \in \Re^m \tag{2}$$

This definition describes monotonicity in *all* features, and if some features are not necessarily monotone, they are sometimes removed prior to analysis (e.g. Feelders & Pardoel (2003), Kotlowski (2008)). For this paper we cater for the more general situation of *partial* monotonicity, in a selected set of features C. We define partial monotonicity similarly to van de Kamp et al. (2009), under the *ceteris paribus* assumption (all other features being equal):

Definition 2.1. ***Partial Dominance***
Given monotone features $C \subseteq \{1, ..., m\}$, the partial order $\preceq_C$ over $\boldsymbol{x}, \boldsymbol{x}' \in \Re^m$ is:

$$\boldsymbol{x} \preceq_C \boldsymbol{x}' \Leftrightarrow \begin{cases} x_j \leq x'_j, \forall j \in C \\ x_j = x'_j, \forall j \in \{1, ..., m\} \setminus C \end{cases} \tag{3}$$

Definition 2.2. ***Partially Monotone Function***
Function $f : \Re^m \rightarrow \Re$ is monotonic *in $C \subseteq \{1, ..., m\}$ if*

$$\boldsymbol{x} \preceq_C \boldsymbol{x}' \Rightarrow f(\boldsymbol{x}) \leq f(\boldsymbol{x}'), \forall \boldsymbol{x}, \boldsymbol{x}' \in \Re^m \tag{4}$$

This definition is a multi-variate deterministic form of the probabilistic definitions for *First Order Stochastic Dominance (FSD) Monotonicity* as proposed by Wellman as a *qualitative influence* (S^+) (Wellman 1990). In this context the classifier, as a deterministic function, has the degenerate probability distribution and $cdf(f(\mathbf{x})) \ FSD \ cdf(f(\mathbf{x}')) \Leftrightarrow f(\mathbf{x}) \geq f(\mathbf{x}')$.

2.2 Random Forest (RF)

The Random Forest (RF) algorithm was originally proposed by Breiman (2001) and emerges from the tree ensemble techniques developed in the late 1990s. It is an ensemble of decision trees, where each component tree is built with two levels of randomisation: first, each tree is trained from a *bootstrap sample* of the training data (rather than the entirety); and second, as each tree is inductively built from the top node down, only a random subset of the available features are considered for the best split at each node. The bootstrap sampling technique is known as 'bagging', for 'bootstrap aggregating'. Thus each tree differs from the well known CART algorithm (Breiman 1984) in that the best split at each node is determined from *mtry* randomly selected (rather than all) features, and also the trees are not pruned.

The accuracy of Random Forest became immediately competitive with the best algorithms. It brought together the benefits of bagging (demonstrated to reduce the sensitivity of classification trees to small perturbations in training data (Breiman 1996)), the random subspace approach developed by Ho (1998), and the use of out-of-bag accuracy estimates. Despite its age and the multitude of new algorithms developed since, it remains one of the most adaptable, successful and efficient algorithms, as demonstrated in the recent comparison of 179 different algorithms on 121 datasets, where it ranked first on accuracy (Fernández-Delgado et al. 2014).

Random Forest is extensively used in bioinformatics, genetic epidemiology and microarray analyses and other high dimensional problems. It also has significant computational advantages over comparable algorithms (such as SVM), being easily parallelised and

having only one hyper-parameter to be tuned (*mtry*). The computation time is linear in the number of trees. It also effectively has a 'built-in' estimate for generalisation error for the ensemble using the out-of-bag samples. This estimate is possible because for each data point, there exists an 'out-of-bag' classifier consisting of the ensemble of approximately one third of the total number of trees built from bootstrap samples which *exclude* that point. Finally, it can handle high dimensional problems, non-linearity, interaction effects, and correlated predictors, and can be used for regression or classification problems.

2.3 Weighted Neighbourhood Form of RF

As shown by Lin & Jeon (2006), the classifier produced by the Random Forest algorithm may be considered as a *weighted neighbourhood scheme*. For a binary RF classifier $f(\mathbf{x})$ trained on the labelled training set $\{(\mathbf{x}_i, y_i) \mid i = 1..N\}$ with $\mathbf{x} \in \Re^m$ and class $y \in \{-1, +1\}$, classifier takes the form:

$$f(\mathbf{x}) = sign\left(\frac{1}{N}\sum_{i=1}^{N} y_i \sum_{t=1}^{T} W_t(\mathbf{x}_i, \mathbf{x})\right) \tag{5}$$

where:

$$W_t(\mathbf{x}_i, \mathbf{x}) = \begin{cases} \frac{b_{i,t}}{K_{t,l}}, \text{ if } \mathbf{x}_i, \mathbf{x} \text{ are in leaf } l \text{ in tree } t \\ 0, \text{ otherwise} \end{cases}$$

$b_{i,t}$ is the number of occurrences of $\mathbf{x}_i$ in the bootstrap sample used to build tree t

$K_{t,l}$ is the number of bootstrap points in leaf l of tree t

T is the number of trees in the ensemble

N is the number of training points

In this case the weighted neighbourhood function $W(\mathbf{x}_i, \mathbf{x}) = \sum_{t=1}^{T} W_t(\mathbf{x}_i, \mathbf{x})$. Thus RF effectively produces a neighbourhood function that represents the similarity between a training point $\mathbf{x}_i$ and any point $\mathbf{x}$ by the number of trees in the ensemble where $\mathbf{x}$ is in the same leaf node as $\mathbf{x}_i$, weighted by the number of bootstrap training points in those leaf nodes.

3 Partially Monotone Random Forest

3.1 PMRF Algorithm

3.1.1 Hard Constrained PMRF

Our algorithm modifies the standard RF classifier as described in (5) by allowing the contribution of each training data-point to be non-uniform, effectively magnifying or diminishing their impact on classifier predictions. Inspired by approaches within shape constrained multivariate regression, including Du et al. (2013) and Hall & Huang (2001), we introduce weights $\mathbf{p} = \{p_i \mid i = 1..N\}$ for each training data point, and so allowing each training data-point to have a weight different to the uniform weight $1/N$. Thus (5) becomes:

$$f(\mathbf{x}) = sign\left(\sum_{i=1}^{N} p_i y_i \sum_{t=1}^{T} W_t(\mathbf{x}_i, \mathbf{x})\right) = sign\left(g(\mathbf{x})\right) \tag{6}$$

Further, for any two points $(\underset{\sim}{\mathbf{x}}, \tilde{\mathbf{x}})$ it is trivial to show that $g(\underset{\sim}{\mathbf{x}}) \leq g(\tilde{\mathbf{x}}) \implies f(\underset{\sim}{\mathbf{x}}) \leq f(\tilde{\mathbf{x}})$. Thus, assuming a set of K pairs of constraint points MC exists ($MC = \{(\underset{\sim}{\mathbf{x}}_k, \tilde{\mathbf{x}}_k), k = 1..K\}$), we propose the following constrained optimisation problem:

$$\min_{\mathbf{p}} \sum_{i=1}^{N} (p_i - \frac{1}{N})^2 \tag{7}$$

$$\text{subject to: } \sum_{i=1}^{N} p_i = 1$$

$$\sum_{i=1}^{N} p_i y_i \sum_{t=1}^{T} W_t(\mathbf{x}_i, \underset{\sim}{\mathbf{x}}_k) \leq \sum_{i=1}^{N} p_i y_i \sum_{t=1}^{T} W_t(\mathbf{x}_i, \tilde{\mathbf{x}}_k)$$

$$k = 1..K$$

This problem finds for the set of N weights $p_i \in \Re$ that minimises the l_2-norm (distance) from the uniform case $p_i = \frac{1}{N}$, while respecting $\sum_{i=1}^{N} p_i = 1$ and the $g(\underset{\sim}{\mathbf{x}}_k) \leq g(\tilde{\mathbf{x}}_k)$ for the K discrete constraints supplied in MC. The uniform case ($p_i = \frac{1}{N}$) corresponds to standard RF (Equation 5), and it is trivial to show that if the constraints $g(\underset{\sim}{\mathbf{x}}_k) \leq g(\tilde{\mathbf{x}}_k)$ are satisfied by the unconstrained RF, the optimal solution will be $p_i = \frac{1}{N}, i = 1..N$ (i.e. unchanged from RF).

This problem is quadratic in $\mathbf{p}$ and can be formulated as below, where $\mathbf{p}$ is an $N \times 1$ column vector, $\mathbf{I}_N$ is an $N \times N$ identity matrix, $\mathbf{1}$ is a $N \times 1$ column vector of ones, $\mathbf{G}$ is a $K \times N$ matrix, $\mathbf{0}$ is a $K \times 1$ column vector of zeros, and $\preceq$ is element-wise inequality:

$$\max_{\mathbf{p}} \; -\frac{1}{2}\mathbf{p}^T\mathbf{I}_N\mathbf{p} + \frac{2}{N}\mathbf{1}^T\mathbf{p} \tag{8}$$

$$\text{subject to } \mathbf{1}^T\mathbf{p} = 1$$

$$\mathbf{G}\mathbf{p} \preceq \mathbf{O}$$

$$\text{where } G_{k,i} = y_i \sum_{t=1}^{T} W_t(\mathbf{x}_i, \underset{\sim}{\mathbf{x}}_k) - W_t(\mathbf{x}_i, \tilde{\mathbf{x}}_k),$$

$$i = 1..N, \quad k = 1..K$$

Note that we did not constrain p_i to be non-negative. When p_i is negative (incurring the appropriate penalty to the objective function), the corresponding training data-point is effectively re-labelled with the opposing class. Given the family of monotone data relabelling approaches (e.g. Duivesteijn & Feelders (2008), Velikova & Daniels (2004)) and leaf node relabelling approaches (e.g. van de Kamp et al. (2009)) such potential for relabelling seems reasonable, and is potentially needed due to noise.

Given the identity matrix $\mathbf{I}$ is positive definite, the objective function is strictly concave. The constraints amount to the intersection of the K halfspaces described by $\mathbf{G}\mathbf{p} \preceq \mathbf{0}$ and the hyperplane described by $\mathbf{1}^T\mathbf{p} = 1$. Since these are all closed convex sets the intersection is also a closed convex set (which may be the empty set {}). Hence there exists one global unique solution to the above problem, unless the constraints are inconsistent and the set of possible $\mathbf{p}$ is {} making the problem infeasible. In practice, with the set of constraint pairs MC as generated in Section 3.1.3, a solution was found for all experiments performed and so this does not appear to be a practical limitation. The problem can be solved by any standard quadratic programming solver such as `quadprog`, variants of which are available in MATLAB, R, and python.

3.1.2 Soft Penalty PMRF

A potential issue with domain knowledge integration is that it may be incorrect, or even if it is correct, can still impair accuracy as discussed by Shmueli (2010). To allow for this potential incorrectness of some domain knowledge, we also formulated a soft penalty form of the problem. We introduce a new hyper-parameter $\lambda \geq 0$ to weight the soft penalty of constraint violation and revise our problem to be:

$$\min_{\mathbf{p}} \sum_{i=1}^{N}(p_i - \frac{1}{N})^2 + \lambda \sum_{k=1}^{K}[g(\underset{\sim}{\mathbf{x}}_k) - g(\tilde{\mathbf{x}}_k)]_+ \quad (9)$$

$$\text{subject to: } \sum_{i=1}^{N} p_i = 1$$

Reformulating it as inequality constraints yields:

$$\min_{\mathbf{p},\boldsymbol{\epsilon}} \sum_{i=1}^{N}(p_i - \frac{1}{N})^2 + \lambda \sum_{k=1}^{K} \epsilon_k \quad (10)$$

$$\text{subject to: } g(\underset{\sim}{\mathbf{x}}_k) - g(\tilde{\mathbf{x}}_k) \leq \epsilon_k, \quad k = 1..K$$
$$\epsilon_k \geq 0$$
$$\sum_{i=1}^{N} p_i = 1$$

Applying Karush Kahn Tucker conditions for optimality, the solution is found at the stationary point of the Lagrangian. First we construct the Lagrangian by introducing dual multipliers $\boldsymbol{\alpha}$,$\boldsymbol{\beta}$,γ:

$$\mathcal{L}(\mathbf{p}, \boldsymbol{\epsilon}; \boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma) = \sum_{i=1}^{N}(p_i - \frac{1}{N})^2 + \lambda \sum_{k=1}^{K} \epsilon_k + \sum_{k=1}^{K} \alpha_k (g(\underset{\sim}{\mathbf{x}}_k) - g(\tilde{\mathbf{x}}_k) - \epsilon_k) - \sum_{k=1}^{K} \beta_k \epsilon_k + \gamma(\sum_{i=1}^{N} p_i - 1) \quad (11)$$

$$\alpha_k \geq 0, \beta_k \geq 0, \quad k = 1..K$$

Setting partial derivatives to zero reveals γ and $\mathbf{p}$ as:

$$\gamma = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} \sum_{t=1}^{T} y_i \alpha_k (W_t(\mathbf{x}_i, \tilde{\mathbf{x}}_k) - W_t(\mathbf{x}_i, \underset{\sim}{\mathbf{x}}_k))$$

$$p_i = \frac{1}{N} - \frac{\gamma}{2} + \frac{1}{2} \sum_{k=1}^{K} \sum_{t=1}^{T} y_i \alpha_k (W_t(\mathbf{x}_i, \tilde{\mathbf{x}}_k) - W_t(\mathbf{x}_i, \underset{\sim}{\mathbf{x}}_k))$$

Re-substitution allows $\boldsymbol{\beta}$, γ, $\boldsymbol{\epsilon}$ and $\mathbf{p}$ to be eliminated and enables the problem to be reduced to an optimisation problem in $\boldsymbol{\alpha}$:

$$\min_{\boldsymbol{\alpha}} \ \frac{1}{2} \boldsymbol{\alpha}^T \mathbf{H} \boldsymbol{\alpha} - \mathbf{1}^T \mathbf{L} \boldsymbol{\alpha} \quad (12)$$
$$\text{subject to } 0 \leq \alpha_k \leq \lambda$$

$$\text{where } L_{i,k} = y_i \sum_{t=1}^{T} W_t(\mathbf{x}_i, \tilde{\mathbf{x}}_k) - W_t(\mathbf{x}_i, \underset{\sim}{\mathbf{x}}_k)$$
$$\mathbf{H} = -\frac{1}{2N} \mathbf{L}^T \mathbf{I}_N \mathbf{L} + \frac{1}{2} \mathbf{L}^T \mathbf{L}$$
$$i = 1..N, \quad k = 1..K$$

This standard quadratic form can be solved by any standard QP solver. Once $\boldsymbol{\alpha}$ is solved, γ and $\mathbf{p}$ can be calculated as above.

If $\mathbf{H}$ is strictly positive definite, this problem is strictly convex and has one unique minima. Experimentally we found that $\mathbf{H}$ was occasionally non-positive definite, in particular for lower dimensional data with large numbers of constraints (`haberman` dataset in particular). When this occurred we eliminated rows/columns of $\mathbf{H}$ that were linearly dependent by using only the pivot vectors of the row echelon form of $\mathbf{H}$. This would typically remove about 10% of the rows/columns of $\mathbf{H}$. Because each row/column corresponding to one constraint pair $(\underset{\sim}{\mathbf{x}}_k, \tilde{\mathbf{x}}_k)$, this effectively removed the same number of constraints from MC.

3.1.3 Constraint Generation

The above algorithms require a set of monotonicity constraints $MC = \{(\underset{\sim}{\mathbf{x}}_k, \tilde{\mathbf{x}}_k), k = 1..K\}$. We want to design this set of constraints to achieve monotonicity of $f(\mathbf{x})$ in a subset of features $C \subseteq \{1, ..., m\}$ as proposed by domain knowledge over the expected input space $\mathcal{X}$ (i.e. we want $\underset{\sim}{\mathbf{x}}_k \preceq_C \tilde{\mathbf{x}}_k$, for which our algorithm will ensure $f(\underset{\sim}{\mathbf{x}}_k) \leq f(\tilde{\mathbf{x}}_k)$).

In Definition 2.2, $\mathbf{x} \preceq_C \mathbf{x}'$ may be usefully considered to be made up of two cases: (a) strictly *univariate* changes (when exactly one feature $c_i \in C$ increases between $\mathbf{x}$ and $\mathbf{x}'$), and (b) strictly *conjunctive* changes (when more than one feature in C increases). Then it is straightforward to show that univariate monotonicity in all $c_i \in C$ is equivalent to the conjunctive monotonicity in any $\{c_i, c_j ...\} \subseteq C$ as per Theorem 3.1:

Theorem 3.1 (Equivalence of Univariate & Conjunctive Monotonicity)**.** *Given monotone features $C \subseteq \{1, ..., m\}$ and function $f : \Re^m \to \Re$, the condition for conjunctive partial monotonicity in (4) is equivalent to:*

$$\boldsymbol{x} \preceq_{c_i} \boldsymbol{x}' \Rightarrow f(\boldsymbol{x}) \leq f(\boldsymbol{x}'), \ \forall c_i \in C, \boldsymbol{x}, \boldsymbol{x}' \in \Re^m \quad (13)$$

Proof:

(i) (4) $\implies$ (13): this follows since each $c \in C$ is simply a special case of (3), where $x_j = x'_j$ for $j \in (C \setminus c)$ i.e. univariate monotonicity in each monotone feature is simply a subset of conjunctive monotonicity.
(ii) (13) $\implies$ (4): Let function $f : \Re^m \to \Re$ comply with (13) for univariate monotonicity for all $c \in C$. For convenience and without loss of generality let $C = \{1, 2, ..., c_{max}\}, c_{max} \leq m$. Let $\mathbf{x} = (x_1, x_2, ..., x_{c_{max}}, ..., x_{m-1}, x_m)$ and $\mathbf{x}' = (x_1 + \delta_1, x_2 + \delta_2, ..., x_{c_{max}} + \delta_{c_{max}}, ..., x_{m-1}, x_m)$, where $\delta_c \geq 0$. It is apparent that $\mathbf{x} \preceq_C \mathbf{x}', \forall \mathbf{x}, \mathbf{x}' \in \Re^m$ thus encompassing (4). We will now show that $f(\mathbf{x}') \geq f(\mathbf{x})$ using only (13):

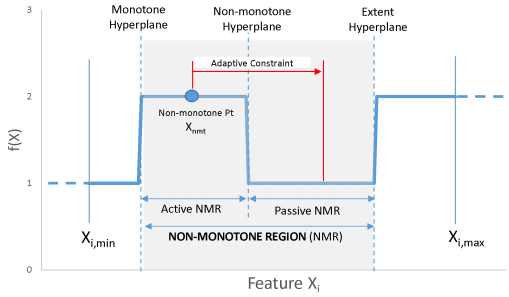


Figure 1: *Non-Monotone Region Illustration. Around each point that is non-monotone in feature X_i is a 'Non-Monotone Region' (NMR) composed of an 'Active' NMR (containing the point) and a 'Passive' NMR (of the alternative class). A constraint is created between the non-monotone point and the mid-point of its 'Passive' NMR.*

$$
\begin{aligned}
f(\mathbf{x}') &= f((x_1+\delta_1, x_2+\delta_2, ..., \\
&\quad x_{c_{max}}+\delta_{c_{max}}, ..., x_{m-1}, x_m)) \\
&= f((x_1, x_2+\delta_2, ..., x_{c_{max}}+\delta_{c_{max}}, ..., \\
&\quad x_{m-1}, x_m) + (\delta_1, 0, ..., 0, ..., 0, 0)) \\
&\geq f(x_1, x_2+\delta_2, ..., x_{c_{max}}+\delta_{c_{max}}, ..., \\
&\quad x_{m-1}, x_m) \\
&= f((x_1, x_2, ..., x_{c_{max}}+\delta_{c_{max}}, ..., \\
&\quad x_{m-1}, x_m) + (0, \delta_2, ..., 0, ..., 0, 0)) \\
&\geq f(x_1, x_2, ..., x_{c_{max}}+\delta_{c_{max}}, ..., \\
&\quad x_{m-1}, x_m) \\
&... \\
&\geq f(x_1, x_2, ..., x_{c_{max}}, ..., x_{m-1}, x_m) \\
&= f(\mathbf{x})
\end{aligned}
$$

Thus compliance with conjunctive monotonicity (4) in monotone features C for any $\mathbf{x}, \mathbf{x}' \in \Re^m, \mathbf{x} \preceq_C \mathbf{x}'$ is demonstrated for any function $f : \Re^m \rightarrow \Re$ complying with univariate monotonicity (13).

Thus it is sufficient to aim for *univariate* monotonicity in each monotone feature c_i. In fact, any strictly conjunctive constraints (where more than one feature increases) are redundant in the presence of univariate constraints. Hence we propose techniques for *univariate* constraint generation, where $\underset{\sim}{\mathbf{x}}_k$ and $\tilde{\mathbf{x}}_k$ differ in the value of *one feature only*.

Given the optimisation problem is polynomial in the number of constraints, we propose to tailor our constraints to correct known problems with the RF model. We first observe that if we investigate how the RF model acts around each training data point as we vary only monotone feature c_i, we can identify the non-monotone regions. Figure 1 illustrates what we term a 'Non-Monotone Region' (NMR) around a point. Note from this figure that given $f(\mathbf{x})$ is a binary function $\Re^m \Rightarrow \{1, 2\}$ and that $f(\mathbf{x}_{nmt}) = 2$ and feature i is monotone increasing, we only need to look for non-monotonicity in values of feature i that are higher than $x_{nmt,i}$. Evaluating $f(x)$ for higher values of feature i identifies a non-monotone hyperplane (when $f(x) = 1$), until at higher values we reach the 'extent' hyperplane (when $f(x) = 2$ again). To correct this non-monotone region, either the 'active NMR' (containing $\mathbf{x}_{nmt}$) or the 'passive NMR' (not-containing $\mathbf{x}_{nmt}$) need to fully reverse.

We thus tailor our constraints to correct the identified Non-Monotone Regions. Although a number of constraints could be used to span these regions and ensure non-monotone 'islands' do not appear in the gaps, in practice we found that it is sufficient to place a single constraint between the non-monotone point and the mid-point of the passive NMR. Algorithm 1 describes this process.

Algorithm 1 Constraint Generation

Input:
$f(\mathbf{x})$ ▷ Trained classifier
$C = \{c_j \mid j = 1..p\}$ ▷ Monotone features
$\{\mathbf{x}_i = (x_{i,1}, .., x_{i,m}) \mid i = 1..N\}$ ▷ Training data
$L > 0$ ▷ No. Continuous variable partitions
Output:
$MC = \{(\underset{\sim}{\mathbf{x}}_k, \tilde{\mathbf{x}}_k) \mid k = 1..M\}$ ▷ Constraint pairs

$MC = \{\}$
for each $c_j \in C$ **do**
  $mn = min_i(x_{i,c_j})$ ▷ minimum feature value
  $mx = max_i(x_{i,c_j})$ ▷ maximum feature value
  $P = (mx - mn)/L$ ▷ partition width
  $G = mn : P : mx$ ▷ grid vector
  for each $\mathbf{x}_i$ $(i = 1..N)$ **do**
    $D = f(\mathbf{x}_i) > 0 \ ? +1 \ : -1$
      ▷ non-monotone search direction
    $class = f(\mathbf{x}_i)$
    $hyperplane = NULL$
    $extent = NULL$
    $G_s = \{g \mid g \in G, Dg > Dx_{i,c_j}\}$
    for each $g \in G_s$ **do**
      $\mathbf{x}_g = (x_{i,1}, ..., x_{i,c_j} = g, ..., x_{i,m})$
      if $f(\mathbf{x}_g) \neq class$ **then**
        if isnull($hyperplane$) **then**
          $hyperplane = g$
          $class = f(\mathbf{x}_g)$
        else
          $extent = g$
          break
        end if
      end if
    end for
    if !isnull($hyperplane$) **then**
      if isnull($extent$) **then**
        $extent = g$
      end if
      $midpt = (hyperplane + extent)/2$
      $\mathbf{x}_c = (x_{i,1}, ..., x_{c_j} = midpt, ..x_{i,m})$
      $MC = MC \cup (\mathbf{x}_i, \mathbf{x}_c)$
    end if
  end for
end for
Return MC

3.2 Measurement of Partial Monotonicity

The practicality of this approach depends on achieving satisfactory levels of monotonicity with reasonable numbers of constraints. To evaluate this we need a measure for the partial monotonicity of a classifier. The dominant measure in the classification literature is to conduct all possible pairwise comparisons of the data points and their predictions. The measure of (non)monotonicity is the proportion of (non)violations, and has various names such as Frequency Monotonicity Rate (FMR) (Chen & Li 2014), degree of monotonicity (NmDeg) (Feelders & Pardoel 2003, Potharst & Feelders 2002), Non-monotonicity Index (NMHI) (Milstein et al. 2013, Marsala & Petturiti 2015), fraction of monotone pairs (Velikova & Daniels 2004, Duivesteijn & Feelders 2008).

Each of the data point comparisons ($N^2 - N$ excluding self-comparisons) is defined as 'comparable' if

the points comply with the dominance relation (1):

$$\mathbf{x} \textit{ comparable with } \mathbf{x}' \Leftrightarrow \begin{cases} (x_j \geq x'_j,\ \forall j = 1..m), \text{ OR} \\ (x_j \leq x'_j,\ \forall j = 1..m) \end{cases} \tag{14}$$

The 'incomparable' pairs (IP) are ignored. The comparable pairs (CP) are then assessed for compliance with monotonicity of the class predicted by the function or given by the data set (Equation 2). *Comparable* pairs that do not comply with this equation (NM) are deemed non-monotone, and typically these are represented as a proportion of the total number of comparisons, for example the Non-Monotonicity Index (NMI), which is given by $NMI = NM/(N^2+N)$.

This measure is useful for monotonicity in *all* features, but in practice cannot be used for our situation of *partial* monotonicity, because in addition to requiring that the constrained features are either all greater than (or less than) or equal, we must modify Equation 14 to require *equality in all unconstrained features*. This dramatically reduces the number of comparable pairs, such that most of the datasets in this paper had a comparability of 0.0%. In practice even one continuous unconstrained feature will typically reduce comparability to zero. Even if equality is relaxed, comparability remains low, such as the `Pima` data-set (Lichman 2013), which only has categorical variables and still has comparability of 4.6%. This makes the NMI measure either impossible, or at least unreliable, in most partially monotone situations.

We thus propose a new measure for a function's monotonicity we call *Monotonicity Compliance* (MCC). Applying Theorem 3.1, we observe that if a function possesses univariate partial monotonicity in all constrained features $c_i \in C$, it also possesses conjunctive monotonicity for any subset $C_s \subseteq C$. Thus we propose to simply measure univariate partial monotonicity for each $c_i \in C$ (MCC_{c_i}), and use the feature average $MCC_{featavg} = \frac{1}{p}\sum_{i=1}^{p} MCC_{c_i}$ as a measure of overall compliance. MCC is intuitively defined as the proportion of the input space where the requested monotonicity constraints are *not* violated, weighted by the joint probability distribution of the input space. This aims to provide a practical measure of how monotone a function is within the expected input space.

Definition 3.1. ***Monotonicity Compliance (MCC)***
For function $f(\boldsymbol{x}) : \Re^m n \Rightarrow \Re$, $\boldsymbol{x} \in X$ with monotone constraints on features $C = \{c_1, ..., c_p\} \subseteq \{1, ..., m\}$, and with joint probability density function $P(\boldsymbol{x})$, the Monotonicity Compliance of f with respect to constrained feature c_i is

$$MCC_{c_i}(f) = \int \cdots \int_X P(\boldsymbol{x}) m_{c_i}(f, \boldsymbol{x})\, dx_1 \ldots dx_m \tag{15}$$

where

$$m_{c_i}(f, \boldsymbol{x}) = \begin{cases} 1,\ if\ m^+_{c_i}(f, \boldsymbol{x}) \geq 0\ and\ m^-_{c_i}(f, \boldsymbol{x}) \geq 0 \\ \frac{1}{2},\ if\ (m^+_{c_i}(f, \boldsymbol{x}), m^-_{c_i}(f, \boldsymbol{x})) \in \\ \quad \{(-1, +1), (+1, -1)\} \\ 0,\ otherwise \end{cases} \tag{16}$$

$$m^+_{c_i}(f, \boldsymbol{x}) = \begin{cases} +1,\ if\ \exists\ Q > 0\ s.t.\ f(\boldsymbol{x}) < \\ \quad f(x_1, \ldots x_{c_i} + Q, \ldots x_m)\ and \\ \quad f(\boldsymbol{x}) = f(x_1, \ldots, x_{c_i} + q, \ldots, x_m) \\ \quad \forall\ 0 < q < Q \\ 0,\ if\ f(\boldsymbol{x}) = f(x_1, \ldots x_{c_i} + q, \ldots, x_m) \\ \quad \forall\ q > 0 \\ -1,\ otherwise \end{cases} \tag{17}$$

$$m^-_{c_i}(f, \boldsymbol{x}) = \begin{cases} +1,\ if\ \exists\ Q > 0\ s.t.\ f(\boldsymbol{x}) > \\ \quad f(x_1 \ldots, x_{c_i} - Q \ldots x_m)\ and \\ \quad f(\boldsymbol{x}) = f(x_1, \ldots, x_{c_i} - q, \ldots, x_m) \\ \quad \forall\ 0 < q < Q \\ 0,\ if\ f(\boldsymbol{x}) = f(x_1, \ldots, x_{c_i} - q, \ldots, x_m) \\ \quad \forall\ q > 0 \\ -1,\ otherwise \end{cases} \tag{18}$$

Essentially the $m_{c_i}(\mathbf{x})$ function looks in the positive and negative directions of x_{c_i} for the first change in f (if any). If in both of directions the function f either does not change, or changes in the correct direction, $m_{c_i}(\mathbf{x})$ returns 1 (a monotone point). If the change in one direction is correct and the other is incorrect, $m_{c_i}(\mathbf{x})$ returns 1/2 (it is 'half' monotone). Otherwise, the point is non-monotone and $m_{c_i}(\mathbf{x}) = 0$. In practice $P(\mathbf{x})$ is unknown, but for given data set $\hat{X}$ of size N, $MCC_{c_i}(f)$ can be simply estimated by the plug-in estimate:

$$\hat{MCC}_{c_i}(f) = \frac{1}{N}\sum_{i=1}^{N} m_{c_i}(f, \mathbf{x}_i) \tag{19}$$

MCC is analogous to the partial derivative based technique in (Daniels & Kamp 1999), but for non-continuous and non-differentiable functions. Although the partial derivative of the real valued SVM function (prior to extracting the sign) could have been used similarly, this is not as accurate an indicator of classifier non-monotonicity, because despite a negative partial derivative the effect of a feature on the resulting class may not cause a change in sign over the input space.

4 Experiments and Datasets

Datasets. Nine datasets were used as described in Table 1, from the UCI Machine Learning Repository (Lichman 2013) and the KEEL Dataset Repository (Alcalá et al. 2010). Rows with missing values were removed.

Experiment Design. For each dataset we conducted 50 experiments, on an Intel i7-4790 8GB RAM PC. For each experiment 2/3 of available data points were randomly selected as the maximum training partition. This training partition was then randomly sub-sampled down to 200, 100, and 50 data point samples to assess the effect of sample size. All sampling was stratified to retain the original dataset class distribution. For each sample size, the remainder of the available data were used as the test partition. The same training/test partitions and cross-validation partitions were used for all constraint techniques to ensure a fair comparison. For the base Random Forest implementation we used `RandomForestClassifier` from the `sklearn` python package (version 0.17.1), with 200 trees in the forest.

Dataset	Src	No. Rows	No. Feats	Output Class	Constrained Attributes
German Credit Ratings	**UCI**	1000	24	Good (-1) Bad (+1)	**Increasing (4):** Loan Dur'n, Single Appl (vs Guarantor), Co-Appl (vs Guarantor), Renting **Decreasing (6):** Cheque Bal, Savings, Mths in Job, Age, Other Installment Plans, Owns House
Pima Indians Diabetes	**UCI**	394	8	Normal (-1) Diabetes (+1)	**Increasing (8):** Num. Preg, Glucose Test, BP, Triceps Fold Thk, Serum Insulin, BMI, Diab Pedigree, Age. **Decreasing (0):** -
Cleveland Heart Disease	**UCI**	299	18	Normal (-1) Disease (+1)	**Increasing (11):** Age, Typ Angina, Atyp Angina, Male, BP, Cholest, ECG=1, ECG=2, Exerc Ang, Exerc ST depn, Num Vessels Fluoro **Decreasing (1):** Max Hrt Rate
South African Heart Disease	**KEEL**	462	9	Normal (-1) Disease (+1)	**Increasing (8):** Systolic BP, Cum Tobacco, Cholest, Adiposity, Family Hist, Type A Behav, Age, Obesity **Decreasing (0):** -
Ljubljana Breast Cancer	**UCI**	277	13	No Recurrence (-1) Recurrence (+1)	**Increasing (3):** Tumor size, No. Inv Nodes, Degree of Malignancy. **Decreasing (2):** Nodes Encapsulated, Irradiation
Car Acceptability	**UCI**	390	6	Unacceptable (-1) Acc (acc/gd/vgd) (+1)	**Increasing (2):** Persons, Safety **Decreasing (2):** Price, Maintenance Reqd
Auto Mileage	**UCI**	392	7	MPG<=28 (-1) MPG>28 (+1)	**Increasing (2):** Origin Japan, Year **Decreasing (2):** Displacement, Weight
Haberman BC Survival	**UCI**	306	3	Died <5yrs (-1) Survived 5yrs (+1)	**Increasing (1):** Year **Decreasing (2):** Age, Nodes
Wisconscin Breast Cancer	**UCI**	683	9	Benign (-1) Malignant (+1)	**Increasing (9):** Clump Thk, Uniform Size, Uniform Shape, Marg Adhes, Epit Size, Bare Nucl, Bland Chrom, Norm Nucl, Mitos **Decreasing (0):** -

Table 1: *Dataset Summary Table.*

Monotone features were identified from domain knowledge and extant literature, as shown in Table 1. The two algorithm variants (hard constraint and soft penalty) were evaluated. Constraints were generated as per Algorithm 1, and the same constraint set was used for both algorithms for each experiment. The number of constraints varies depending on the number of non-monotone regions found.

Each experiment proceeded as follows:

1. **Estimate standard RF hyper-parameter:** The optimal *mtry* was estimated for standard RF using a parameter sweep on $mtry \in \{1, 2, 3, 4, 5, 6, 8, 10, 12, 14\}$. The optimal *mtry* had the minimum MCR as determined by stratified 10-fold cross-validation.
2. **Fit standard RF model:** The standard *unconstrained* RF model was fitted to the whole training partition using the optimal *mtry*.
3. **Generate constraints:** Constraints were generated based on the *training* data and fitted RF model using Algorithm 1.
4. **Fit constrained RF model:** The PMRF model (estimating new sample weights p_i) was fitted to the training partition using the constraints and fitted standard RF model. For the *hard constrained* PMRF a single fit was required. For the *soft penalty* PMRF a parameter sweep on the penalty weight λ was used on $\lambda \in \{0, 0.1, 0.5, 1, 3, 5, 7.5, 10, 50, 100\}$, and the optimal λ had the minimum MCR as determined by stratified 10-fold cross-validation.
5. **Estimate performance:** Accuracy and monotonicity were both measured on the *test* partition.

5 Results and Discussion

5.1 Classifier Partial Monotonicity

The impact on partial monotonicity of the classifier is in Figure 2, in terms of the average MCC over the monotone features. For the unconstrained RF model, it can be seen that generally the datasets can be roughly grouped as (a) less than 95% monotone (`haberman`, `ljubBC`, `SAheart`); (b) 96 to 98% monotone (`pima`, `cleveland`, `german`); and (c) 99% monotone or higher (`car`, `autompg`, `WBCdiag`). In fact for group (c), the constraint generation algorithm usually found zero violations of the requested monotone features, resulting in zero constraints and thus leaving the RF model unchanged ($p_i = \frac{1}{N}$).

Hard-constrained PMRF is consistently able to increase monotonicity to about 98% or higher. Thus the increase in monotonicity for group (a) is large, for group (b) is moderate and group (c) is absent. Soft constrained PMRF was uniformly part way between the unconstrained RF and the hard-constrained PMRF, indicating that the monotonicity violation penalty weighting term λ was selected to impose less than full compliance with the constraint pairs.

5.2 Classifier Accuracy

Figure 3 shows the variation in Accuracy with sample size. The group of datasets with virtually monotone unconstrained RF models is omitted because the PMRF model is usually identical to the RF model as discussed above. For the datasets where monotonicity improvement is possible (as shown) we generally see accuracy improvement of a similar order to the increase in monotonicity. Thus the largest accuracy increase for hard-constrained PMRF is for the dataset with the largest increase in monotonicity (`haberman` - 3-5% increase), followed by `ljubBC` (1-3%) and `SA heart` (0.5-1%) and `pima` 0-0.5%.

These results are reflected in Figure 4 which show a scatter plot of all experiments for Increase in Monotonicity (MCC) vs Increase in Accuracy. The linear least squares trend line is shown, and the slope is 0.20 (positive) with p-value of 0.000 (< 0.05). While the variance is high and there is clearly no underlying linear causal model, the dominance of the experimental points in the non-negative quadrant and the significant positive slope suggests that where monotonicity can be increased, an increase in accuracy is likely. Table 2 also summarises the change in accuracy and Cohen's κ for the largest training size for each dataset.

Thus generally it appears that accuracy increases are available for the monotone models *if* the unconstrained model is insufficiently monotone. In terms of sample size, we expected that the largest increases in accuracy would occur at lower sample sizes (van de Kamp et al. 2009), because additional data would

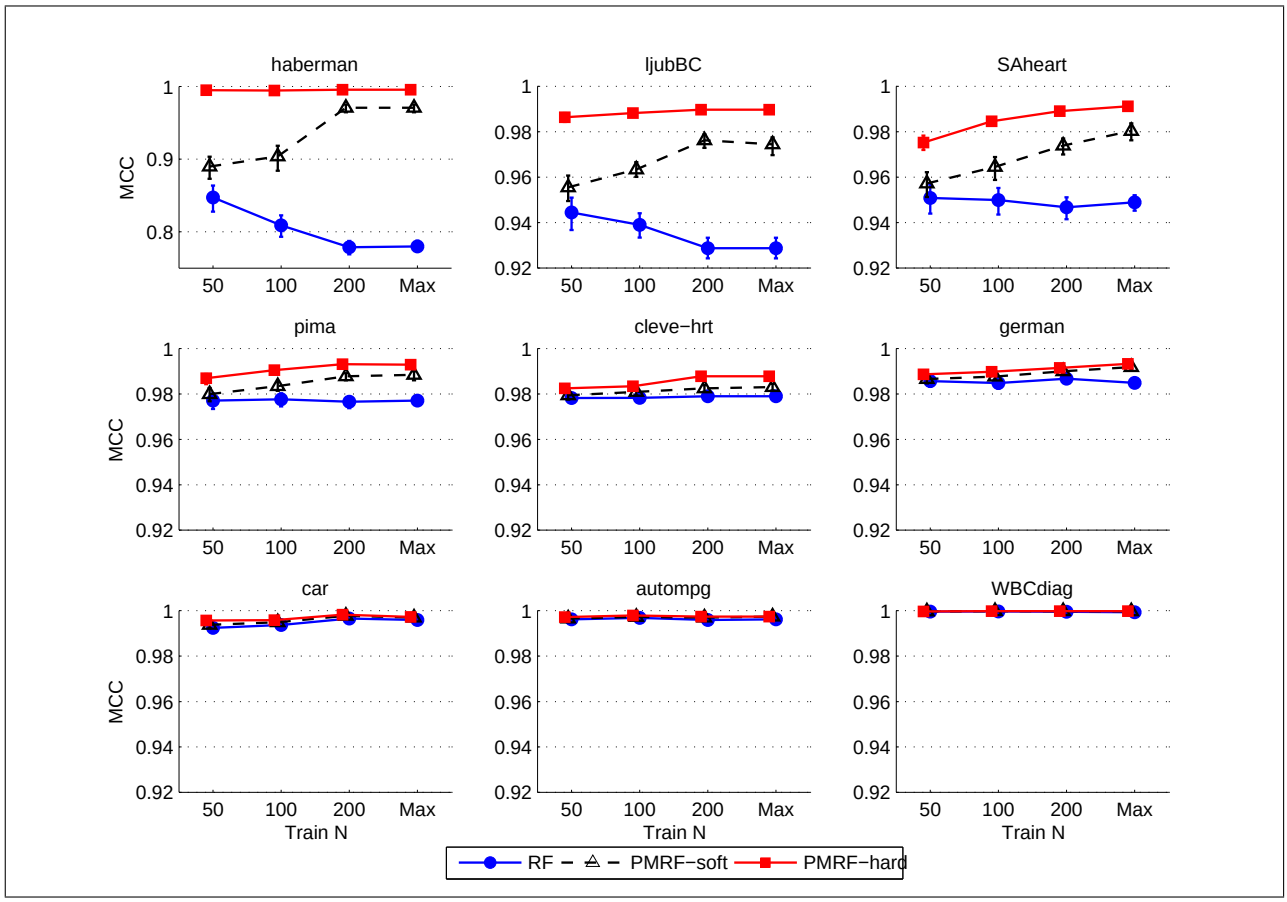


Figure 2: *Classifier Partial Monotonicity vs Technique and Sample Size (note: different y-scale for* `haberman`*). The datasets may be roughly grouped by unconstrained RF monotonicity as (a) Low ($MCC \leq 95\%$)- top row; (b) Moderate ($95 < MCC < 99\%$) - middle row; and High ($MCC \geq 99\%$) - bottom row.*

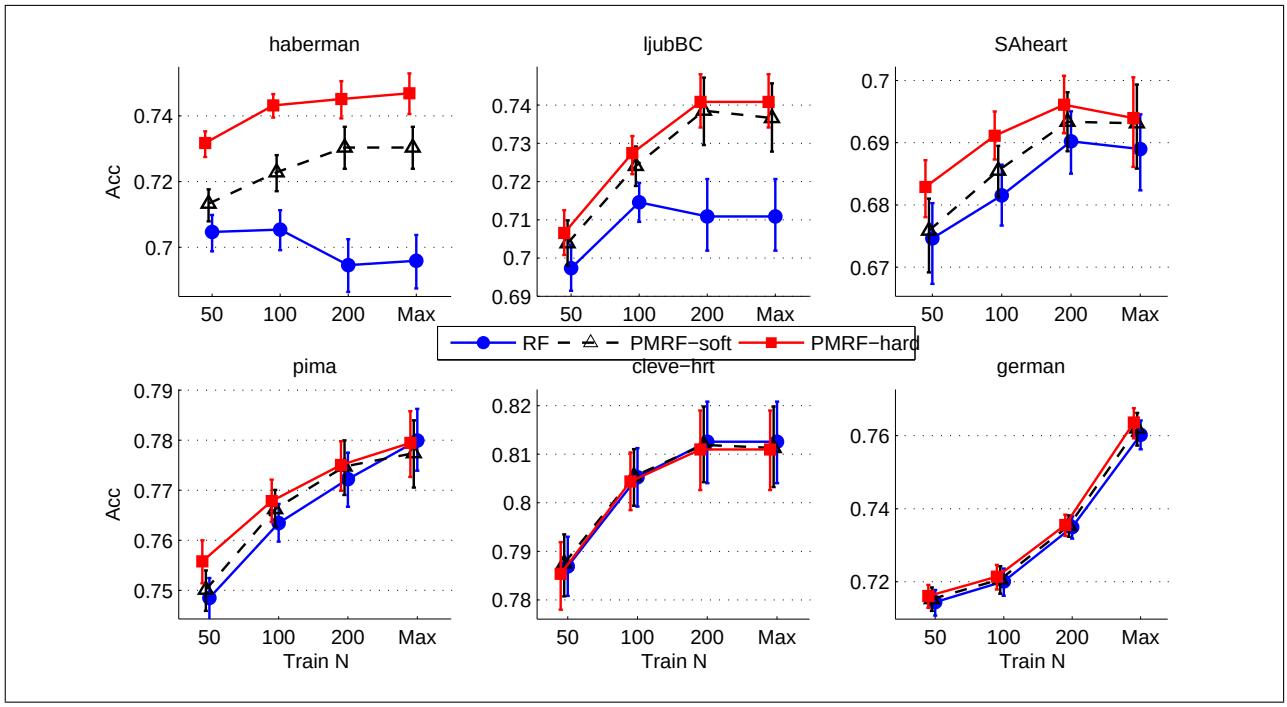


Figure 3: *Effect of PMRF techniques on Accuracy at Different Sample Sizes*

.

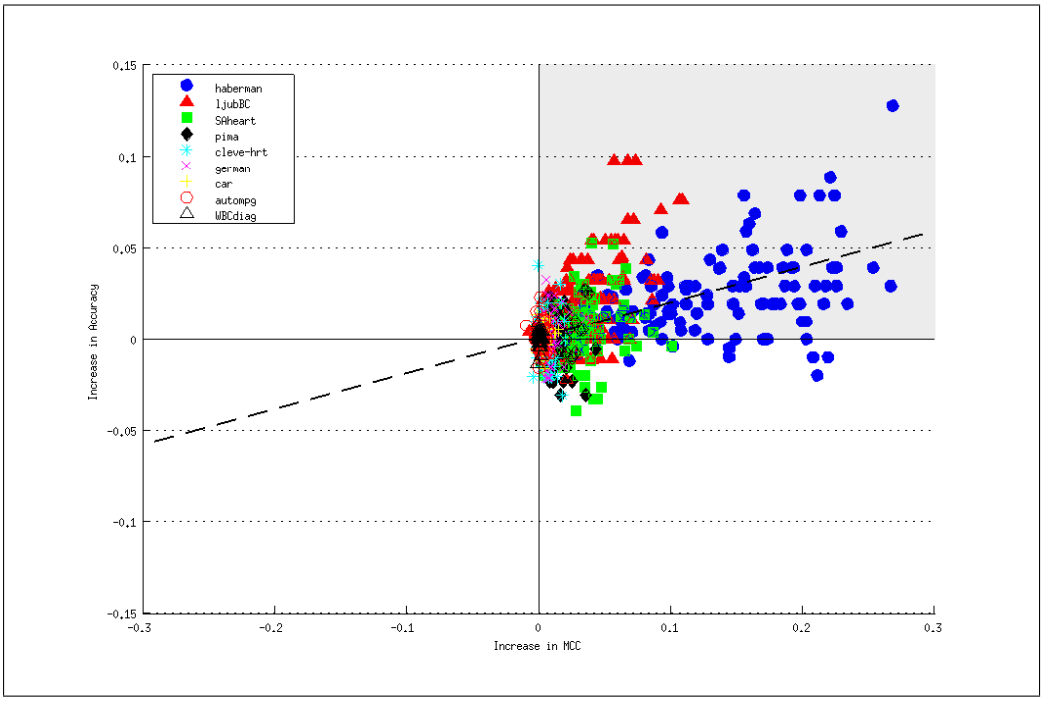


Figure 4: *Increase in Monotonicity vs Increase in Accuracy for Hard-Constrained PMRF (all datasets, experiments and sample sizes). Least Squares trendline shown in dashed blue.*

		ACCURACY				COHEN'S KAPPA		
		RF Mean	MCSVM Mean*	Exptwise Increase	Exptwise Incr P-	RF Mean	MCSVM Mean*	Exptwise Increase
Haberman	PMRFs	0.696	0.730	**0.035**	0.000	0.156	0.200	**0.044**
	PMRFh	0.696	0.747	**0.051**	0.000	0.156	0.238	**0.082**
LjubBC	PMRFs	0.711	0.737	**0.026**	0.000	0.231	0.271	**0.040**
	PMRFh	0.711	0.741	**0.030**	0.000	0.231	0.271	**0.040**
Saheart	PMRFs	0.689	0.693	0.004	0.118	0.266	0.287	**0.021**
	PMRFh	0.689	0.694	0.005	0.102	0.266	0.289	**0.024**
Pima	PMRFs	0.780	0.777	-0.003	0.182	0.489	0.483	-0.006
	PMRFh	0.780	0.779	-0.001	0.856	0.489	0.488	-0.001
Cleve	PMRFs	0.813	0.811	-0.001	0.390	0.622	0.619	-0.003
	PMRFh	0.813	0.811	-0.002	0.446	0.622	0.618	-0.003

		ACCURACY				COHEN'S KAPPA		
		RF Mean	MCSVM Mean*	Exptwise Increase	Exptwise Incr P-	RF Mean	MCSVM Mean*	Exptwise Increase
German	PMRFs	0.760	0.762	0.002	0.298	0.373	0.392	**0.019**
	PMRFh	0.760	0.764	**0.003**	0.026	0.373	0.386	**0.013**
Car	PMRFs	0.966	0.967	0.000	0.624	0.920	0.921	0.001
	PMRFh	0.966	0.966	-0.001	0.264	0.920	0.918	-0.002
AutoMF	PMRFs	0.922	0.922	0.001	0.354	0.797	0.799	0.002
	PMRFh	0.922	0.922	0.001	0.474	0.797	0.798	0.002
WBCdiag	PMRFs	0.971	0.972	0.000	0.686	0.938	0.938	0.000
	PMRFh	0.971	0.972	0.001	0.062	0.938	0.939	0.002

Table 2: *Classifier Performance Summary at Maximum Sample Size ($\frac{2}{3}$ available). Statistically significant change shaded (at $p < 0.05$)*

tend to eliminate the need for domain knowledge. However, while this holds for some datasets (`pima`, `SAheart`), interestingly others show the largest increase in accuracy at higher sample sizes (`haberman`, `ljubBC`). This perhaps suggests that in those cases the monotone relationships are permanently obscured by noise, regardless of sample size.

The soft-penalty PMRF yielded accuracy increases inbetween unconstrained RF and hard-constrained PMRF, similar to the reduced increases in monotonicity it enforced. We had hoped that it would allow constraints that were damaging to accuracy to be omitted while retaining the useful constraints, potentially improving overall performance. There was one case where this did occur, for the `SA Heart` dataset, when performance in terms of Cohen's κ was slightly impaired by the hard constrained PMRF for the $N = 50$ and 100 sample sizes. In this case the soft penalty PMRF successfully prevented this loss of performance. However, far more often it merely served to reduce the improvement produced by hard constrained PMRF, suggesting that in general the significant additional computational expense (to perform a parameter sweep with cross validation) is not justified.

6 Conclusions

This paper presented two variants of a novel algorithm for the integration of domain knowledge regarding partial monotonicity into the Random Forest classifier, and investigated the results for nine real datasets. We aimed to augment this powerful classifier with the flexibility of *partial* (rather than *full*) monotonicity, while retaining the high levels of accuracy RF often achieves.

The experimental results found that:

- Standard unconstrained Random Forest is often very good at recognising monotone relationships, being almost perfectly monotone for three of the nine datasets ($MCC > 99\%$), even at small sample sizes.
- For the remaining six datasets ($MCC \leq 98\%$), our proposed hard-constrained PMRF algorithm variant was reliably able to lift monotonicity to 98% or higher.
- Accuracy improvements were seen for five of these six datasets, generally in proportion to the increase in monotonicity induced.

The *soft-penalty* PMRF algorithm variant chose to enforce a lower degree of monotonicity and had lower increases in accuracy than the hard constrained version, making the additional computation it required unwarranted. The simpler hard constrained PMRF is generally preferable.

Overall, *for datasets where RF does not fully recognise feature monotonicity*, our hard-constrained PMRF algorithm offers potentially substantial improvements in both monotonicity and accuracy (up to 5% increase). In addition, the proposed monotonicity measure (MCC) offers a way to identify whether the technique is likely to help. We consider this a useful augmentation of an already extremely high performing algorithm. Future exciting avenues include improved monotone feature selection and constraint generation processes, and extension to multiclass Random Forest.

References

Alcalá, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L. & Herrera, F. (2010), 'Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework', *Journal of Multiple-Valued Logic and Soft Computing* **17**(255-287), 11.

Ben-David, A. (1995), 'Monotonicity maintenance in information-theoretic machine learning algorithms', *Machine Learning* **19**(1), 29–43.

Bioch, C. & Popova, V. (2003), Induction of ordinal decision trees, Technical report, Erasmus Research Institute of Management (ERIM), ERIM is the joint research institute of the Rotterdam School of Management, Erasmus University and the Erasmus School of Economics (ESE) at Erasmus University Rotterdam.

Breiman, L. (1996), 'Bagging predictors', *Machine learning* **24**(2), 123–140.

Breiman, L. (2001), 'Random forests', *Machine learning* **45**(1), 5–32.

Chen, C.-C. & Li, S.-T. (2014), 'Credit rating with a monotonicity-constrained support vector machine model', *Expert Systems with Applications* **41**(16), 7235–7247.

Daniels, H. & Kamp, B. (1999), 'Application of mlp networks to bond rating and house pricing', *Neural Computing & Applications* **8**(3), 226–234.

Daniels, H. & Velikova, M. (2003), *Derivation of monotone decision models from non-monotone data*, Tilburg University.

Du, P., Parmeter, C. F. & Racine, J. S. (2013), 'Nonparametric kernel regression with multiple predictors and multiple shape constraints', *Stat Sin* **23**(3), 1343–1372.

Duivesteijn, W. & Feelders, A. (2008), 'Nearest neighbour classification with monotonicity constraints', *Machine Learning and Knowledge Discovery in Databases* pp. 301–316.

Feelders, A. J. (2000), *Prior knowledge in economic applications of data mining*, Springer, pp. 395–400.

Feelders, A. & Pardoel, M. (2003), *Pruning for monotone classification trees*, Springer, pp. 1–12.

Fernández-Delgado, M., Cernadas, E., Barro, S. & Amorim, D. (2014), 'Do we need hundreds of classifiers to solve real world classification problems?', *The Journal of Machine Learning Research* **15**(1), 3133–3181.

González, S., Herrera, F. & García, S. (2015), 'Monotonic random forest with an ensemble pruning mechanism based on the degree of monotonicity', *New Generation Computing* **33**(4), 367–388.

Hall, P. & Huang, L.-S. (2001), 'Nonparametric kernel regression subject to monotonicity constraints', *Annals of Statistics* pp. 624–647.

Ho, T. K. (1998), 'The random subspace method for constructing decision forests', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(8), 832–844.

Kotlowski, W. (2008), Statistical approach to ordinal classification with monotonicity constraints, PhD thesis, Pozna University of Technology Institute of Computing Science.

Lauer, F. & Bloch, G. (2008), 'Incorporating prior knowledge in support vector machines for classification: A review', *Neurocomputing* **71**(7), 1578–1594.

Li, S.-T. & Chen, C.-C. (2014), 'A regularized monotonic fuzzy support vector machine for data mining with prior knowledge', *Fuzzy Systems, IEEE Trans* **PP**(99).

Lichman, M. (2013), 'Uci machine learning repository'.
URL: *http://archive.ics.uci.edu/ml*

Lin, Y. & Jeon, Y. (2006), 'Random forests and adaptive nearest neighbors', *Journal of the American Statistical Association* **101**, 578–590.

Mangasarian, O. L. & Wild, E. W. (2008), 'Nonlinear knowledge-based classification', *Neural Networks, IEEE Transactions on* **19**(10), 1826–1832.

Marsala, C. & Petturiti, D. (2015), 'Rank discrimination measures for enforcing monotonicity in decision tree induction', *Information Sciences* **291**, 143–171.

Milstein, I., David, A. B. & Potharst, R. (2013), 'Generating noisy monotone ordinal datasets', *Artificial Intelligence Research* **3**(1), p30.

Potharst, R., Ben-David, A. & van Wezel, M. (2009), 'Two algorithms for generating structured and unstructured monotone ordinal data sets', *Eng. App. of Art. Intell.* **22**(4), 491–496.

Potharst, R. & Bioch, J. C. (2000), 'Decision trees for ordinal classification', *Intelligent Data Analysis* **4**(2), 97–111.

Potharst, R. & Feelders, A. J. (2002), 'Classification trees for problems with monotonicity constraints', *ACM SIGKDD Explorations Newsletter* **4**(1), 1–10.

Shmueli, G. (2010), 'To explain or to predict?', *Statistical science* pp. 289–310.

van de Kamp, R., Feelders, A. & Barile, N. (2009), 'Isotonic classification trees', *Advances in Intelligent Data Analysis VIII* pp. 405–416.

Velikova, M. & Daniels, H. (2004), 'Decision trees for monotone price models', *Computational Management Science* **1**(3-4), 231–244.

Wang, Y. & Ni, H. (2012), 'Multivariate convex support vector regression with semidefinite programming', *Knowledge-Based Systems* **30**, 87–94.

Wellman, M. P. (1990), 'Fundamental concepts of qualitative probabilistic networks', *Artificial Intelligence* **44**(3), 257–303.

WaterDM: A Knowledge Discovery and Decision Support Tool for Efficient Dam Management

Md Zahidul Islam **Michael Furner** **Michael J. Siers**

School of Computing and Mathematics, Charles Sturt University, Australia
Email: {zislam,mfurner,msiers}@csu.edu.au

Keywords: Decision Support System, Knowledge Discovery, Dam Water Management

Abstract

Dams provide many benefits to society. They provide water for domestic, industrial and irrigation purposes. Many dams also generate electricity using hydroelectric systems. Therefore, effective tools for analysing and monitoring dams are of importance to society. Our system, WaterDM, provides several tools for performing data mining on dam data. Our design can also be used to guide the implementation of data mining tools for other applications.

1 Introduction

Water is a limited resource. Dams are constructed to collect rain water. Hydropower is a process used by power stations to generate electricity by collecting the energy created by flowing water. According to the Australian Renewable Energy Agency, there were more than 120 hydroelectric power stations in Australia in 2013. They also report that the availability of water is a "key constraint on future growth in hydroelectricity generation in Australia" (*Australian Renewable Energy Agency* 2016). Hydroelectricity is considered a renewable energy, which is currently of great interest globally. Therefore, efficient management of dams is important. This paper presents our system *WaterDM* which uses data mining to allow a user to uncover insights from a dam's data, thereby providing a useful tool for the efficient management of dams. Our system is currently a proof of concept and can be modified to meet the requirements of a potential industry partner. WaterDM uses SysFor (Islam & Giggins 2011) models to predict several attributes including a dam's future volume, inflow, evaporation, and release[1]. Since the models built by SysFor are decision trees, the patterns which determine these attributes are easily observable and understandable.

WaterDM is a *decision support system*, and a *knowledge discovery tool*. The system presented in this paper may also guide readers in the design of other decision support and knowledge discovery systems. The rest of this paper is structured as follows. A summary of the related work is given in Section 2. Details on the implementation and usage of WaterDM are provided in Sections 3 and 4 respectively. Finally, our future improvements are given in Section 5.

[1] Volume, inflow, evaporation and release are typically measured as rates. However, in WaterDM, these are measured as the total per day.

2 Related Work

A decision support system (DSS) is a program which assists in decision making (Power et al. 2015). For data-driven DSSs, the user is provided with a set of data analysis tools. These tools typically allow the user to explore the data visually, discover patterns within the data, and make predictions on new data. Data mining techniques can be incorporated into a DSS to automate the knowledge discovery. Data-driven DSSs require data to be used as input. Thus, the design of data-driven DSSs requires the specification of where the data will be collected from and how it is preprocessed into a format suitable for data mining.

DSSs have been developed for predicting the required amount of water for irrigation on Australian farms (Khan et al. 2011*b*)(Khan et al. 2011*a*). Traditionally, the amount of required water is calculated using a formula. However, the authors showed that their proposed DSS could predict the required water more accurately than the traditional method. Many DSSs have been proposed for medical applications. For example, ADDIS (Van Valkenhoef et al. 2013) is a DSS which is used to analyse the results of clinical trial data. A DSS has been proposed which aids in the selection of locations to construct wind farms (Gorsevski et al. 2013). Such a system aids in the adoption of renewable energy sources.

3 Implementation of WaterDM

Figure 1 shows the implementation framework of WaterDM. Raw weather data is regularly downloaded from the Bureau of Meteorology (*Bureau of Meteorology* 2016), NSW Water (*New South Wales Water* 2016), and Weatherzone (*Weatherzone* 2016) using a Cron Job. WaterDM then preprocesses this data using R scripts. When a user requests a model to be built, a bash script runs the Java code for SysFor on the preprocessed data. The model is then added to the MySQL database. The PHP site acts as an interface to the system.

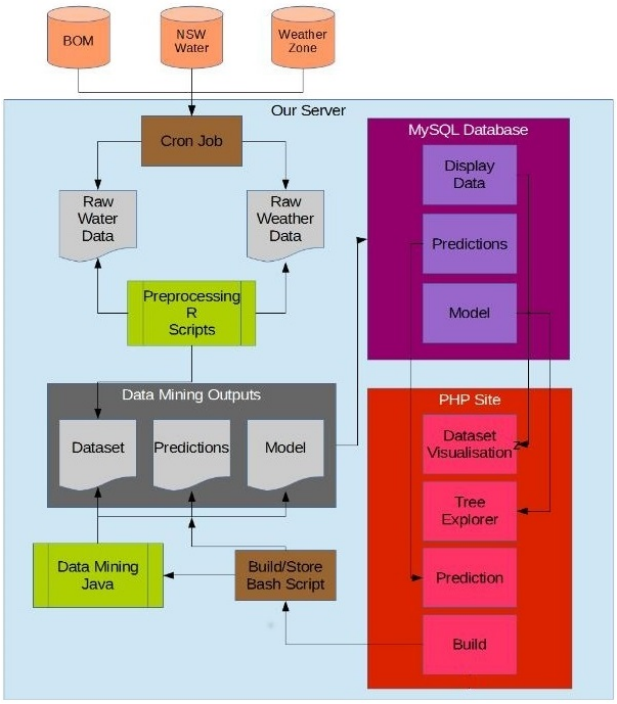


Figure 1: The Implementation Framework of our System: WaterDM

4 Using WaterDM

In this Section, we explain the usage of WaterDM by providing screenshots and descriptions of how the user interacts with the system.

4.1 Home Screen

After signing in to the system, users are sent to the home screen as shown in Figure 2.

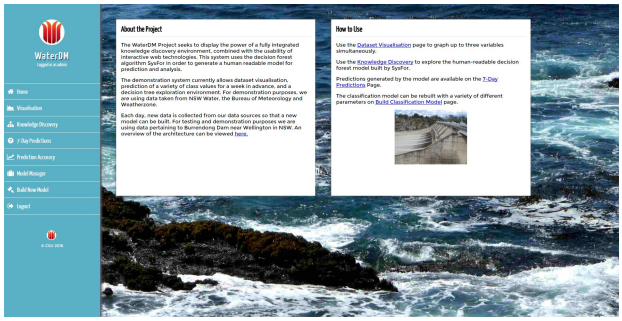


Figure 2: The Home Screen of WaterDM

On all pages, the main menu is displayed on the left hand side of the screen. This menu is shown in higher detail in Figure 3. From this menu, the user can access all functions of the WaterDM program. Each of these tasks are described in detail in the remainder of this Section.

4.2 Visualisation

From the main menu, the the *visualisation tool* can be accessed by clicking on "Visualisation". Within this tool, users may graph different attributes against each other for comparison. This may be used for exploring the relationship between one or more attributes. For

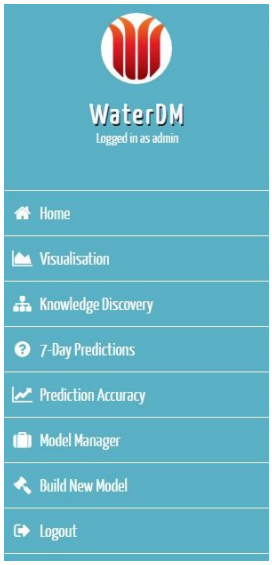


Figure 3: The Menu for WaterDM

example, in Figure 4, we can observe a clear correlation between the max temperature (upper curve) and the evaporation (lower curve). In this figure, each point along the x-axis represents a day and the y-axis is comprised of two different measures. The evaporation is in mm and the max temperature is in degrees celcius.

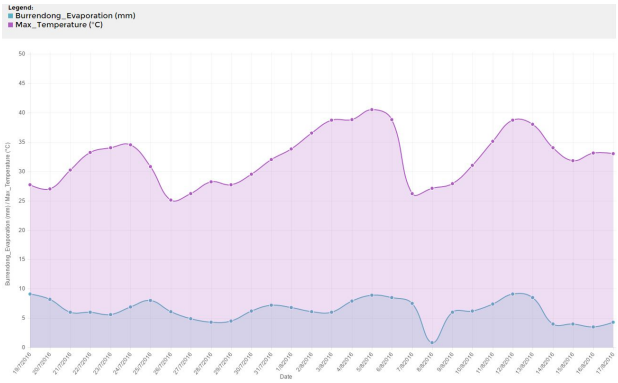


Figure 4: Comparing the evaporation of Burrendong dam (Upper Curve) with max temperature (Lower Curve) for each day from 19/7/16 till 17/8/16.

4.3 Building a Model

From the "Build new model" option in the main menu, multiple models can be built which are saved to a database. These models have parameter settings which are set using the interface shown in Figure 5. Within this interface, a subset of attributes may be chosen for building the model. To remove an attribute, a user can click on the x at the top right of an attribute. A preset of attributes can be saved and loaded for later by clicking on "Save Preset" and "Load Preset" respectively. Since the data mining algorithm used in WaterDM is a decision forest algorithm (Islam & Giggins 2011), the users can control the minimum number of leaf records, and number of trees.

The class attribute for a model must be specified before building. This class attribute can be volume, evaporation, inflow or release. The chosen class attribute is chosen using the drop-down menu. Since

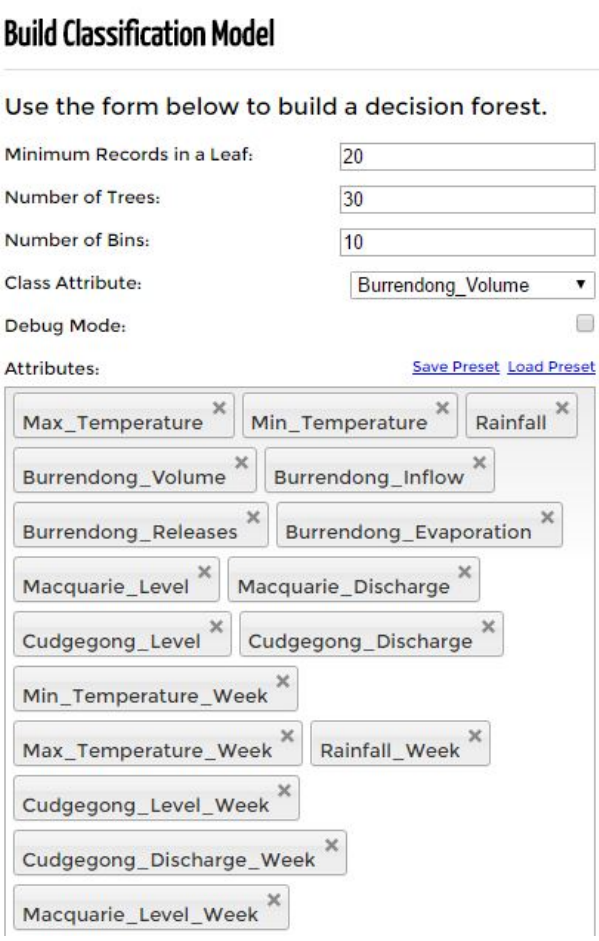


Figure 5: The Interface for Building a Model

these values are numerical, they must be discretised for use in the decision forest classifier. The user can set the number of bins used for discretisation in the third field of this interface. The number of bins must be set appropriately since too few bins will result in useless predictions, and too many bins will result in poor performance. This being a knowledge discovery system, it allows a user to explore knowledge by carefully choosing parameter values such as the number of bins. In this system, equal frequency binning has been used where each bin contains an equal number of instances. If the user does not know how many bins to use, then he/she can start the knowledge discovery process with the default value of 10.

4.4 Inspecting Model Accuracy

From the "Prediction Accuracy" option in the main menu, the user can inspect the accuracy of the built models. The prediction accuracy screen is shown in Figure 6.

There are two parts to this interface which are shown in Figures 7 and 8. In the interface shown in Figure 7, the user can select a model to inspect the accuracy for by selecting a model from the drop-down box. This figure shows five details about the chosen model (Model #118). The first, third and fourth details are the settings that were chosen when building the model (See Section 4.3). The last detail is the classification accuracy of the model.

The accuracy over time can also be inspected graphically as shown in Figure 8. In this figure, a user can observe the bins which have been fitted by

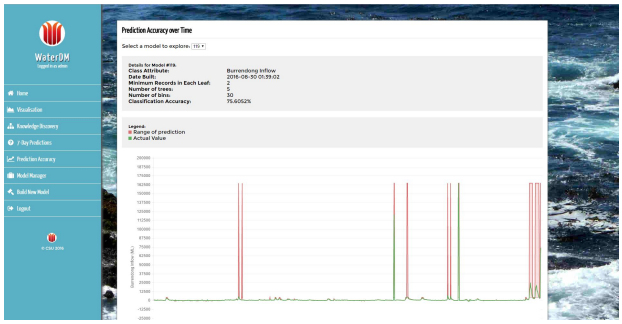


Figure 6: The "Prediction Accuracy" Screen

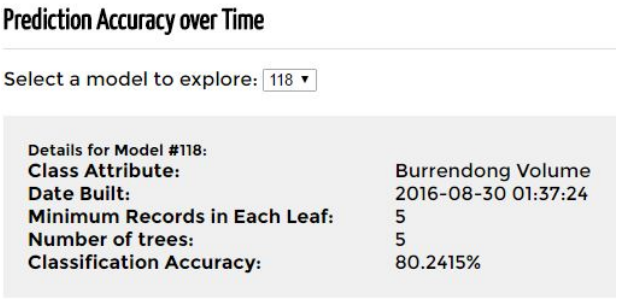


Figure 7: Inspecting the Accuracy of a Built Model

the data mining algorithm to the data. This provides the user with a simple way of viewing how close the model fits the data. The green line is drawn through a set of points where each point represents the volume of Burrendong Dam each day for the previous three years. The bins which have been fitted to the data are shown as red rectangles. We have used a simple discretization method in which each bin contains the same amount of records. This means that when the amount of training data points is low, the bins are larger. To get higher quality classification, we could alternatively use a more sophisticated approach which is optimised for creating small bins even when there are a small amount of training data points (Rahman & Islam 2016). However, since WaterDM achieves good performance already, this will be a future improvement.

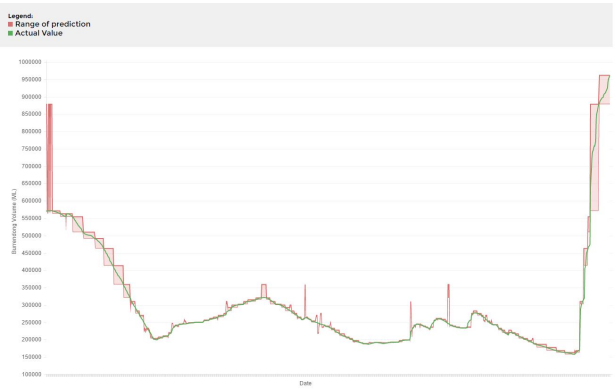


Figure 8: Inspecting the Accuracy of a Built Model Graphically (x-axis: Each day for the previous three years; y-axis: The volume in ML of Burrendong Dam) A larger version of this graph can be found on the last page.

4.5 Performing Knowledge Discovery

Each built tree can be visually explored as shown in Figure 9. Only a small section of a tree is shown here. In WaterDM, we draw nodes as diamonds, and leaf nodes are drawn as circles. After clicking on a leaf, the confidence, support and class label distribution is displayed as shown in Figure 10. The greener the circles around confidence and support, the higher their values. Conversely, the redder they are, the lower their values. This can be seen in Figure 10. Since the support is low, its color is red, whereas the confidence is 100% so it is green. This colour also applies to the leaf node circles in Figure 9. This allows the user to quickly find the high confidence rules by inspecting the colour of the leaf nodes.

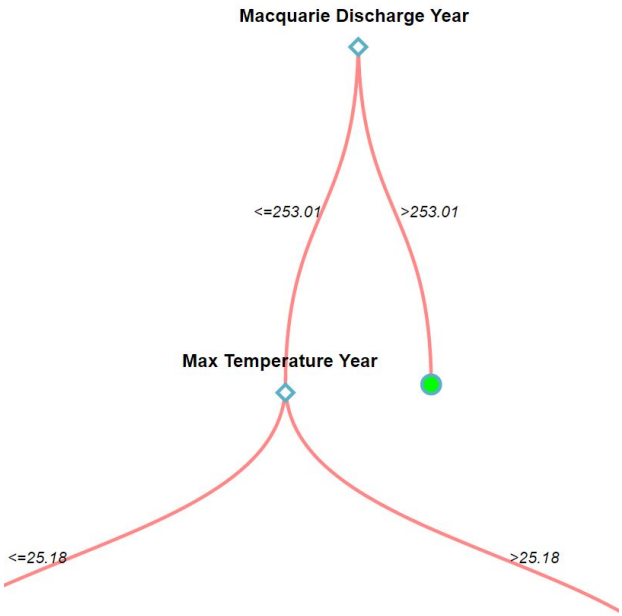


Figure 9: A Section of a Tree

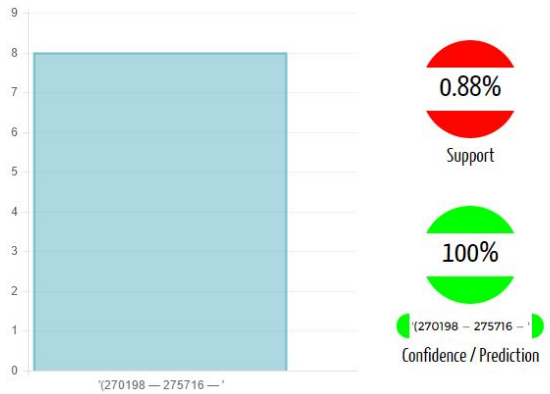


Figure 10: Inspecting a Leaf

4.6 Prediction on the Next Seven Days

Using the interface shown in Figure 11, the next seven days of the chosen class attribute may be predicted using an already built model. This figure shows the predicted volume of *Burrendong dam* for the next week. Burrendong dam is located 337km northwest of Sydney, Australia.

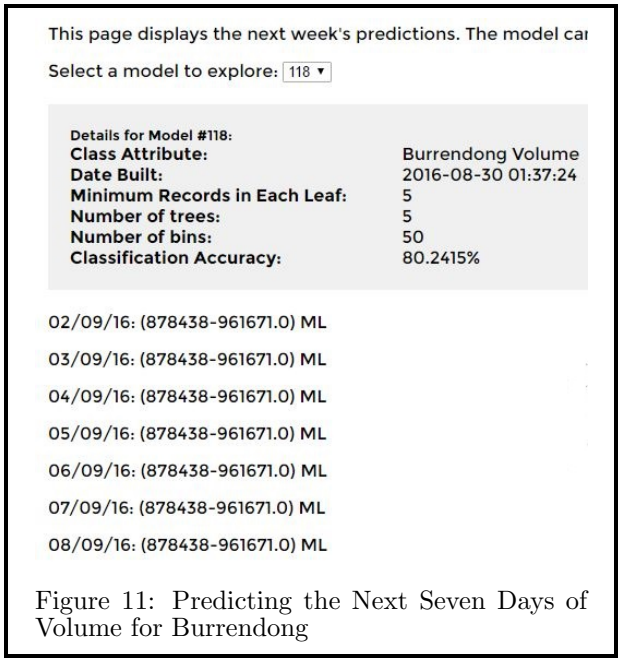


Figure 11: Predicting the Next Seven Days of Volume for Burrendong

As another example, Figure 12 shows the predictions for the next seven days for inflow in Burrendong dam.

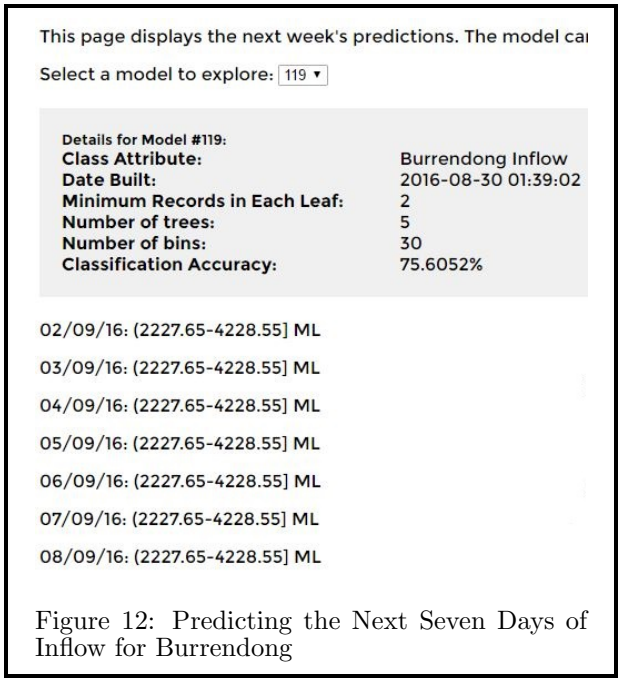


Figure 12: Predicting the Next Seven Days of Inflow for Burrendong

4.7 Managing the Built Models

The model manager can be accessed by clicking "Model Manager" in the main menu. The model manager allows users to view the already created models and their information. Models may also be deleted if the user is an admin. Basic level users cannot delete models. There are also links for each model to make predictions and view the trees. For example, the information provided for a model is shown in Figure 13. The search box at the top of the screen allows the user to search for previously built models.

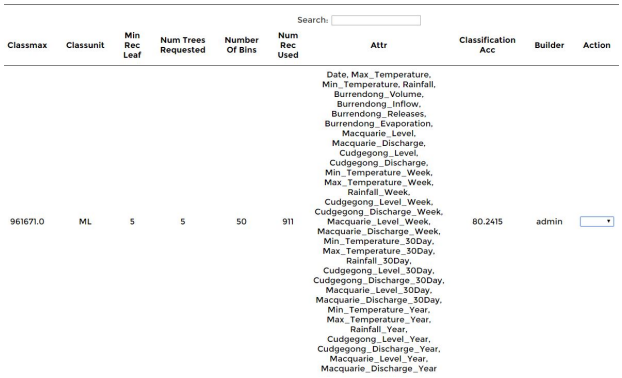


Figure 13: The Information Shown in the Model Manager. A larger version of this figure can be found on the last page.

5 Conclusion and Further Improvements

Our system *WaterDM* is a knowledge discovery and decision support tool for efficient dam management. This paper can also serve as a guide for creating knowledge discovery and decision support tools for other applications. Our future work includes support for cost-sensitivity using the decision forest algorithms CSForest (Siers & Islam 2014) and BCSForest (Siers & Islam 2015).

Acknowledgements

We would like to express our thanks to Peter Siers for his electrical engineering and dam expertise. We are also thankful to Professor Bernard Pailthorpe for his suggestions to the system.

References

Australian Renewable Energy Agency (2016), `http://arena.gov.au/about-renewable-energy/hydropower/`. Accessed: 2016-02-09.

Bureau of Meteorology (2016), `http://www.bom.gov.au/`. Accessed: 2016-18-08.

Gorsevski, P. V., Cathcart, S. C., Mirzaei, G., Jamali, M. M., Ye, X. & Gomezdelcampo, E. (2013), 'A group-based spatial decision support system for wind farm site selection in northwest ohio', *Energy Policy* **55**, 374–385.

Islam, Z. & Giggins, H. (2011), Knowledge discovery through sysfor: a systematically developed forest of multiple decision trees, *in* 'Proceedings of the Ninth Australasian Data Mining Conference-Volume 121', Australian Computer Society, Inc., pp. 195–204.

Khan, M. A., Islam, M. Z. & Hafeez, M. (2011*a*), 'Irrigation water requirement prediction through various data mining techniques applied on a carefully pre-processed dataset', *Journal of Research and Practice in Information Technology* **43**.

Khan, M. A., Islam, Z. & Hafeez, M. (2011*b*), Irrigation water demand forecasting: a data preprocessing and data mining approach based on spatio-temporal data, *in* 'Proceedings of the Ninth Australasian Data Mining Conference-Volume 121', Australian Computer Society, Inc., pp. 183–194.

New South Wales Water (2016), `http://www.water.nsw.gov.au/`. Accessed: 2016-18-08.

Power, D. J., Sharda, R. & Burstein, F. (2015), *Decision support systems*, Wiley Online Library.

Rahman, M. G. & Islam, M. Z. (2016), 'Discretization of continuous attributes through low frequency numerical values and attribute interdependency', *Expert Systems with Applications* **45**, 410–423.

Siers, M. J. & Islam, M. Z. (2014), Cost sensitive decision forest and voting for software defect prediction, *in* 'PRICAI 2014: Trends in Artificial Intelligence', Springer International Publishing, pp. 929–936.

Siers, M. J. & Islam, M. Z. (2015), 'Software defect prediction using a cost sensitive decision forest and voting, and a potential solution to the class imbalance problem', *Information Systems* **51**, 62–71.

Van Valkenhoef, G., Tervonen, T., Zwinkels, T., De Brock, B. & Hillege, H. (2013), 'Addis: a decision support system for evidence-based medicine', *Decision Support Systems* **55**(2), 459–475.

Weatherzone (2016), `http://www.weatherzone.com.au/`. Accessed: 2016-18-08.

Search:

Classmax	Classunit	Min Rec Leaf	Num Trees Requested	Number Of Bins	Num Rec Used	Attr	Classification Acc	Builder	Action
961671.0	ML	5	5	50	911	Date, Max_Temperature, Min_Temperature, Rainfall, Burrendong_Volume, Burrendong_Inflow, Burrendong_Releases, Burrendong_Evaporation, Macquarie_Level, Macquarie_Discharge, Cudgegong_Level, Cudgegong_Discharge, Min_Temperature_Week, Max_Temperature_Week, Rainfall_Week, Cudgegong_Level_Week, Cudgegong_Discharge_Week, Macquarie_Level_Week, Macquarie_Discharge_Week, Min_Temperature_30Day, Max_Temperature_30Day, Rainfall_30Day, Cudgegong_Level_30Day, Cudgegong_Discharge_30Day, Macquarie_Level_30Day, Macquarie_Discharge_30Day, Min_Temperature_Year, Max_Temperature_Year, Rainfall_Year, Cudgegong_Level_Year, Cudgegong_Discharge_Year, Macquarie_Level_Year, Macquarie_Discharge_Year	80.2415	admin	

Figure 14: The Information Shown in the Model Manager. This figure shows the information for one model. However, the user can scroll down to see information on more built models. From the drop-down box, the user can choose to delete the model, make predictions for the model, or view the model. However, the user requires admin privileges to be able to delete models.

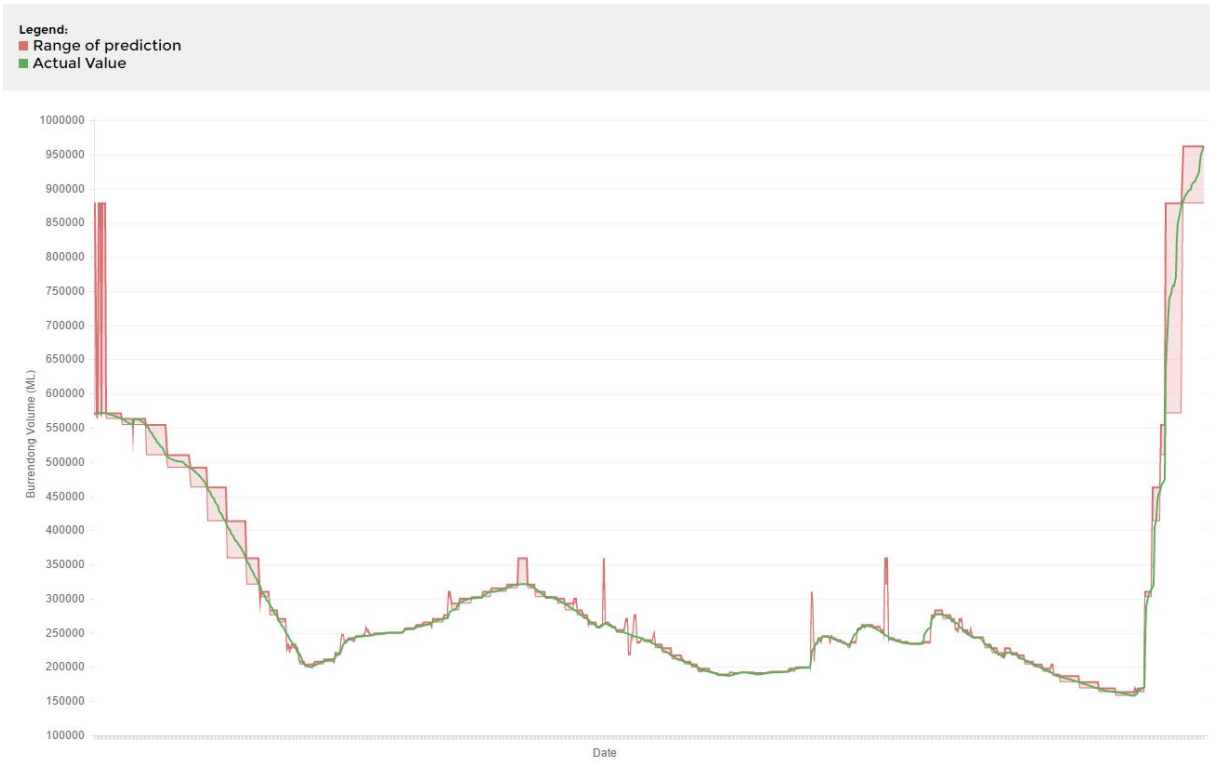


Figure 15: Inspecting the Accuracy of a Built Model Graphically. (x-axis: Each day for the previous three years; y-axis: The volume in ML of Burrendong Dam) This graph be generated for models which are built to predict other variables such as inflow and outflow.

Online Machine Learning Algorithms for Outlier Detection in Network Traffic Data

Andrew Gill **Paul Montague**

Defence Science and Technology Group
PO Box 1500, Edinburgh, South Australia 5111,
Email: (andrew.gill, paul.montague)@dsto.defence.gov.au

Abstract

Anomaly-based Network Intrusion Detection Systems (ANIDS) have not received as much attention from commercial vendors due to the success of signature-based systems and the spectre of high false alarm rates. However, ANIDS are capable of discovering new (zero-day) attacks.

Unsupervised machine learning (UML) techniques have been proposed to address this problem. Data captured at a network gateway satisfy the three-v's (volume, variety and velocity) of big data. One of the key challenges for UML is developing an effective online version to address these issues, particularly for the inference problem of Bayesian mixture models.

In this paper we explore five UML-based ANIDS, detail their (proposed) online implementations, and compare their performance on The Protected Repository for the Defense of Infrastructure against Cyber Threats (PREDICT) dataset – an initiative from the US Department of Homeland Security.

We find that online UML-based ANIDS show promise for high flow network traffic anomaly detection, with individual algorithms suited to differing regimes of True Positive Rates (TPRs) and with Alert Rates (ARs) centred around 10^{-3} with False Discovery Rates (FDRs) no greater than 50%. We also find mixed results regarding the value of incorporating analyst feedback in the models.

Keywords: Outlier, Network Intrusion, Machine Learning, Unsupervised, Online

1 Introduction

The requirements for the protection of data and the assurance of operations increasingly places greater importance on the protection of the computer networks of many businesses and organisations. An Intrusion Detection System of some sort is generally employed as part of the protection system, and a recent review of such is provided in (Bhuyan et al. 2014). Our interest, motivated by application to zero-day attacks, is with UML-based ANIDS, and in particular with online UML algorithms for efficient processing of streaming data. We focus on three simpler algorithms (k-Means, k-Nearest-Neighbour (k-NN) and the Histogram-Based-Outlier-Score (HBOS)) together with two more sophisticated algorithms (Gaussian Mixture Model (GMM) and Latent Dirichlet Allocation (LDA)). We propose modified k-NN and HBOS algorithms in order to support an online UML-based ANIDS, and tailorings to both GMM and LDA to adapt to the problem outlined. Support for non-stationary distributions is also considered, and the k-Means algorithm chosen is specifically adapted to this scenario.

The algorithms were tested on a subset of raw network traffic from the PREDICT dataset (Scheper et al. 2009). The DARPA Scalable Network Monitoring Program Traffic is a 6TB synthetic dataset of pcap files simulating legitimate traffic and a variety of network attacks over a 10 day period in a network of about 14,000 hosts. For the analysis in this paper, the dataset has been heavily subsampled to a dataset appropriate for effective and efficient performance measurement of our algorithms. That is, a number of categorical (*e.g.* service and connection status) and numerical *e.g.* data sizes and statistics for packet timings and lengths – normalised across a sample calibration subset of the training data) features were extracted for each record. LDA was used to model the categorical features, while the other four algorithms were each used to model the numerical features. A companion paper (Abraham et al. 2016) investigates the utility in ensembling these methods.

Though this dataset is labelled, we only used the labels in calculating the performance metrics. Hence we used a short sequence of (assumed) non-malicious training data to build an initial model (malicious data is generally rare – particularly for sampling post-filtering by signature-based Intrusion Detection Systems). Each model then estimated the anomalousness of each subsequent record in turn against a threshold, and if not classed as anomalous it was then included in the model update. An option whereby false positives (as assessed by analyst feedback) were also included in the model update was examined as well.

The typical performance metrics of TPR and False Positive Rate (FPR) were used, along with the AR (proportion of all records that were classed as anomalous) and the FDR (proportion of records classed as anomalous that were false positives). We consider the AR and FDR are more useful for practical applications where datasets are highly asymmetric with few attacks and alerts being presented to a human analyst for review.

2 Gaussian Mixture Model

2.1 Overview

Finite mixture models model the data as a simple finite linear combination of distributions, *e.g* in the

case of finite GMMs, as underlying normal distributions. The motivating intuition is that one is able to arbitrarily closely approximate any given probability density function with such a combination, at least with sufficiently many components. The use of finite GMMs for anomaly detection goes back to (Aitkin & Wilson 1980), and there have been many applications to cyber examples (Yamanishi et al. 2004).

The conjugate prior for the multinomial distribution (which is parametrised by the weights of the individual distributions in the combination) is the Dirichlet distribution in this finite case. We are interested instead in mixture models with an unbounded number of components, determined dynamically as the model evolves. For this non-parametric case, the limiting case of the Dirichlet distribution is the Dirichlet process (DP). Application of DP mixture models to anomaly detection has been explored in, for example, (Ngan et al. 2012). Such models provide a flexible and yet often tractably computable scheme with which to model the data.

(Carvalho, Lopes, Polson, Taddy et al. 2010) lists and summarises a number of the possible methods available for fitting mixture models, including Expectation Maximisation, Markov Chain Monte Carlo and Variational Bayes (VB). Each suffers from drawbacks, including computational inefficiencies, inability to handle high-dimensional data sets and questionable quality of approximation. (Carvalho, Lopes, Polson, Taddy et al. 2010) instead develops an efficient Sequential Monte Carlo approach for Bayesian inference, which is inherently suited to our preferred online learning scenario. Specifically, a particle learning (PL) approach is used (Carvalho, Johannes, Lopes & Polson 2010), which is distinct from the more traditional particle filtering techniques. The PL techniques avoid problems with particle degeneration and only require tracking of sufficient statistics – which can result in a significant reduction in storage requirements. Finally, the PL approach is naturally parallelisable, and thus lends itself to efficient implementation.

2.2 Implementation

The application of PL techniques with DP GMMs for anomaly detection is discussed in (Merl 2009). Though the implementation in our paper is based on the analysis contained in (Merl 2009), we extend that approach to encompass a second measure of outlierness, which will be described below. We refer the reader to Merl's paper for details. (However, we note that Merl's parameter ν differs by $d + 1$ from the definition used in (Murphy 2007), where d is the dimension of the data.)

As discussed in (Merl 2009), the online (sequential) nature of PL enables the analysis of the anomalousness of records as they arrive, based upon the records seen thus far only, rather than evaluating anomalousness across the data set as a whole. Merl proposes a measure of *anomalousness* as the proportion of particles for which the new record causes the creation of a new mixture component during the Propagate step.

In addition to this measure, we introduce a second measure, which we term *suspicion*, defined as the average across the particles of (1− number of points in selected distribution/average distribution size within that particle). This allows us to flag records which fall into sparse distributions created during establishment of the baseline model by outlier events – for noisy data this can be an issue since true anomalies may be masked by such distributions. We declare a record to be an anomaly if either it exceeds a specified threshold in *anomalousness* or it exceeds a specified threshold in *suspicion*.

We developed a Java implementation as per the above. We note that data storage requirements are essentially linear in the number of particles used (which is fixed) and the number of distributions assigned within each particle (which increases over time, but as we shall see in the results is relatively bounded). Each particle is created with a single Gaussian distribution containing the first record to be processed (and the mean vector and variance matrix initialised accordingly).

For the training set, the subsequent data records are then fed to the algorithm one by one, and either added to existing distributions or a new distribution is created accordingly – no anomalies are reported during this phase. For the main data set, as each record is evaluated, if it is flagged as an anomaly (*i.e.* if its *anomalousness* or *suspicion* exceed the specified thresholds) then the record is marked in the log as such and the model is not updated.

If analyst feedback is being tested, records flagged as anomalous and which are false positives are represented to the model and the addition of the record to the model is forced, regardless of any anomaly indication.

Finally, once the run on the data set is complete, the entire data set is run through the model again with no updates to the model allowed, in order to maintain a static set of distributions. The distributions (or, if they are flagged as anomalous, the lack of an assigned distribution) for each record for each particle are logged. This is for visualisation of the distribution assignments of the model in order to understand the clustering of the data.

With regard to the details of the implementation, there are several points of note:

- During the Gibbs update of the parameters, there were issues associated with singularities due to, for example, a number of constant values in the data, arising from the fact that a lot of the data is actually integer values rather than real-valued. In order to overcome these problems, two arbitrary thresholds were introduced – one to restrict contribution to the Gibbs update to those distributions with more than a certain number of points, and the other to only those distributions with covariance matrix above a certain threshold (dynamically evaluated based on the dimension of the data).

- Sampling from the Wishart distribution naively involves a sum whose range is of size linear in the number of points in the distribution component. Since we are dealing with potentially unbounded size streaming data sets, this can quickly become computationally intractable. We use a theorem due to (Odell & Feiveson 1966) to express the Wishart distribution in terms of a sum of a set of chi-squared distributions of size the dimension of the data vectors instead. The chi-squared distributions may then be either sampled directly or, above a certain specified threshold, approximated by a normal distribution.

- In contrast to some of the other algorithms discussed in this work, the *anomalousness* threshold, since it is based on a probabilistic interpretation, can be set without recourse to trial and error with the data. This can be of significant benefit in an online scenario.

Algorithm	TPR	FPR	AR	FDR
GMM (s=0.9)	0.99	2.1E-03	5.1E-03	0.41
GMM (s=1.0)	0.24	6.0E-04	1.3E-03	0.46
GMM (s=0.9) + analyst feedback	0.99	3.1E-03	6.1E-03	0.51
GMM (s=1.0) + analyst feedback	0.10	8.6E-05	3.9E-04	0.22

Table 1: Results for GMM Algorithm

2.3 Results

The GMM algorithm was tested on the PREDICT data set with 10 particles and an *anomalousness* threshold of zero. *Suspicion* thresholds, s, of values 0.9 and 1.0 were trialled. The results are shown in Table 1.

The results show that the use of *suspicion* as an alternative trigger for flagging an anomaly is effective, in that, without this measure, many of the malicious records are missed (*i.e.* lower TPR value at $s = 1.0$) since they are assigned to sparse distributions created during the model's evolution – though we note that the FDR values remain similar.

It is observed that analyst feedback has a detrimental effect in the case $s = 0.9$. This might be expected as a number of new (therefore sparse) distributions will be created via analyst feedback, and the *suspicion* score for subsequent points assigned to such distributions will be higher. Conversely, analyst feedback improves the results in the case $s = 1.0$. With such a *suspicion* threshold, only *anomalousness* scores can trigger an anomaly being flagged, *i.e.* sparse new distributions no longer have an adverse impact on the scoring. The use of a *suspicion* threshold less than 1.0 will therefore depend on the desired outcome – if a high TPR is required, the analyst should consider lowering this threshold. But if benefits from analyst feedback (*e.g.* in terms of reduced FDR) are desired, the threshold may be left at 1.0.

Finally, we note that, as indicated in the previous section, the number of distributions remains relatively bounded. The average number of Gaussians generated without analyst feedback is 8, and with analyst feedback is 18.

3 Online k-Nearest Neighbour

3.1 Overview

k-NN is a simple classification and regression algorithm – either classifying a point based on the majority class of its k nearest neighbours, or assigning an output to a point based on the value of the k nearest neighbours (Altman 1992).

It can also be used in anomaly detection, with appropriate interpretation. For example, kNN-GAS is one of the outlier detection algorithms available in RapidMiner[1](Ramaswamy et al. 2000). These are for the non-online case – the authors propose a simple algorithm whereby points are ranked by the (sum of the) distance to the $k-$th nearest neighbour, and the top ranked points are declared outliers.

[1]https://github.com/Markus-Go/rapidminer-anomalydetection/

3.1.1 Online k-NN Classification

Considering the online case, (Forster et al. 2010) considers fine tuning of a pre-trained gesture recognition system based on feedback from a user (*correct* or *error*) regarding the validity of classification results. Such a system can accommodate a data distribution which is non-stationary, *e.g.* due to a change of sensor position, addition of new users, or user behavioural changes. To this end, they introduce the concept of weights assigned to each point in the training set. They then use an incremental online learning system for a weighted k-NN classifier, *c.f.* regression or anomaly detection.

The approach of (Forster et al. 2010) is as follows. Given positive integers k and m:

1. Training data is stored and all weights set to 1.
2. Classification of a new instance post-training is based on the largest sum of weights for the k nearest points from that class.
3. The user reports *correct* or *error*.
   (a) If *correct*, the point is added to the model, with the class found, and weight 1. The weights of the m nearest points of the same class are increased by some appropriate function – essentially weights are constrained to [0,2] to"prevent single instances biasing the classification".
   (b) If *error*, the weights of the m nearest points of that class are reduced. Any points with weights below a specified threshold are eliminated.

In addition, (Forster et al. 2010) introduce additional features to control the behaviour, largely due to the unconstrained number of points. k and m are adapted dynamically as the number of points changes.

3.2 Proposed Algorithm

In our case, we wish to develop a fully online approach to outlier detection with k-NN. However, the techniques of (Forster et al. 2010) (a) do not immediately extend to the case where an unbounded amount of data will be processed post-training, and (b) are for a classifier algorithm. We make the simplifying decision to keep our total number of stored points fixed. We do however adopt the weighting scheme used in (Forster et al. 2010). This allows for adaptation to non-stationary distributions and reinforces learning from the training set.

Our approach is simply to add each point in as it arrives, and remove an old point in the process. That selection will be based on weight – which is updated in the process, as per (Forster et al. 2010). We shall keep things simpler by using only one parameter (*i.e.* $k = m$), and, since our number of points is fixed, we do not need to dynamically modify this as we proceed.

Our proposed algorithm, for a given k and a threshold t, is then heuristically:

1. Initialise the set of points with the training data, and set all weights to 1.
2. Read in point x from the test data.
3. Find the k closest points and set x's score to the sum of the distances to these k points.

Algorithm	TPR	FPR	AR	FDR
OKNN	0.19	3.2E-04	8.9E-04	0.36
	0.26	6.3E-04	1.4E-03	0.45
	0.99	3.9E-03	6.9E-03	0.57
	1.00	6.1E-02	6.4E-02	0.95
OKNN + analyst feedback	0.19	1.9E-04	7.6E-04	0.25
	0.26	5.1E-04	1.3E-03	0.40
	0.53	2.2E-03	3.8E-03	0.58
	0.86	1.3E-02	1.5E-02	0.83

Table 2: Results for OKNN Algorithm

4. If score $< t$
   (a) Increase the weights of the k nearest points to x.
   (b) Replace the point of least weight with x (and set its weight to 1) (note: in the case of a tie, break the tie arbitrarily).
   (c) Decrease the weights of the k points nearest to the point that was replaced.
5. If score $>= t$, report x as anomalous and do not update the cohort of points.

The weight decrease function we use is simply weight w is updated to $w/2$. The weight increase function is w is updated to $2 * (w/(1 + w^2/4))$. This ensures weights remain in the interval (0,2).

Available space does not allow us to explore the various options considered. But we note that different options were used at various points in the algorithm, *e.g.* weighting distances by the weights of the points, not updating weights, *etc.*, and evaluated against our test data set. The above algorithm proved to be optimal among those tested on the sample data.

3.3 Implementation

Implementation is straightforward, and performance not an issue with the test data used. Optimisation of the various naive search techniques for discovering nearest points would result in a significant increase in performance. We leave this to future work.

One concern for this algorithm is that the weights of the points will accumulate at either 0 or 2, and the benefit afforded from the diversity of weights will be lost. Some simple analysis was done during initial testing in this work. However, a full investigation of the diversity of the weights, and any consequent need to consider a periodic/threshold triggered renormalisation of the weights and/or a change in the aggressiveness of the weight increase and decrease functions, is left to future work.

Consideration of use of metrics other than Euclidean is left as a topic for future research.

3.4 Results

Our online k-NN (OKNN) algorithm was run on the PREDICT data set with $k = 10$ and thresholds ranging from 10 to 300 (with equal threshold values used for the two sets of results). The results are shown in Table 2. We note that analyst feedback has a small effect on the results, reducing the AR at a given threshold, as expected, with a corresponding (beneficial) reduction in FDR.

4 Online Histogram-Based Outlier Score

4.1 Overview

In (Goldstein & Dengel 2012) the authors describe an outlier detection algorithm, HBOS, which has been open sourced in the RapidMiner anomaly detection extension. The basic idea is to fit a histogram to the data, and the anomaly score is then simply the log of the inverse of the corresponding histogram density – summed over all dimensions in the case of multiple dimensions. An overarching assumption is to ignore feature dependence, so that a univariate method may be used (and trivially extended to multiple features). This results in significant performance benefit, though at the cost of a possible loss of precision.

They propose two models: one with fixed width bins, and the other with dynamic width bins containing a fixed number of points (though the simplest approach is complicated by the presence of multiple points of the same value).

4.2 Proposed Algorithm

The challenge in trying to develop an online version of the HBOS (OHBOS) algorithm is to allow for future (as yet unseen) points when creating the bins. There are a couple of choices:

- Create bins dynamically as needed.
- Allocate a fixed set of bins based on the training set.

We adopt the simpler approach and allocate bins up front. However, we use a mixture of the static and dynamic bin-width approaches of (Goldstein & Dengel 2012) in order to allow for future points in currently sparse areas of the distribution. The basic idea then is to default essentially to a set of fixed width bins, but introduce additional bins in regions where the training data is dense. This means that we retain coverage in the sparser areas (at least if our fixed width is sufficiently small) and also limit the number of points in a bin in the denser areas.

Heuristically, the proposed algorithm is as follows. The training set is used for initial allocation of the set of bins. Given values for *threshold*, *resolution* (essentially order of magnitude for number of bins to allocate across the range of the training data set), *scale* (multiplier to add a margin around the data values spanned by the training data) and *numTrain* (the size of the training data set), for each dimension:

- Set default bin size to span of data for the *numTrain* training data set records / *resolution*.
- Allocate a size *scale* times the span of the training data in the empty region before the data and divide it into empty bins of the default size.
- Set maximum number of points per bin for the training data, *maxPoints*, to *numTrain* / *resolution*.
- Order the training data.
- Add the training data points to bins in order. The bins should either be of size the default size, calculated above, or appropriately smaller if the default step size contains more than the *maxPoints* points (special handling of cases of runs of constant values in the data means that, in that case, a bin may contain more than *maxPoints* points)

Algorithm	TPR	FPR	AR	FDR
OHBOS	0.29	1.3E-03	2.1E-03	0.59
	0.48	1.5E-03	2.9E-03	0.50
	0.99	6.6E-03	9.5E-03	0.69
	1.00	5.7E-02	6.0E-02	0.95
OHBOS + analyst feedback	0.28	2.7E-04	1.1E-03	0.242
	0.48	5.0E-04	1.9E-03	0.256
	0.55	3.7E-03	5.4E-03	0.691
	1.00	5.3E-02	5.6E-02	0.95

Table 3: Results for OHBOS Algorithm

- Allocate a size *scale* times the span of the training data in the empty region after the data and divide it into empty bins of the default size.

For each record in the test data:

- Evaluate its anomaly score by summing over the logarithmic score in each dimension computed as follows (using logs to control underflow):
  - Identify the bin in the given dimension into which the record falls.
  - Logarithmic score is - log (normalised number of points / width of bin), where normalised number of points is the number of points in the bin / total number of points.
- If score >= threshold, report anomaly to analyst.
- If score < threshold, add the point to its assigned bin in each dimension.

Note that, in addition to the issue noted above regarding the underlying assumption of the independence of the data dimensions, this approach requires a substantial representative sample of data for training. Also, because of the static approach chosen for bin allocation, it is not an ideal approach for non-stationary distributions. We leave addressing of such issues to further work.

4.3 Results

OHBOS was run on the PREDICT data set with $resolution = 10$, $scale = 10$, $numCal = 10000$ and values of *threshold* ranging from 10 to 100. At the upper end of the range of *threshold*, the only data points being flagged as outliers are those with an infinite anomaly score, *i.e.* those which fall into empty bins in at least one of the dimensions. Further increase in the *threshold* value beyond 100 will not have any effect. The results are shown in Table 3.

We note that analyst feedback can have a significant benefit on the FDR values, as would be expected for this simple algorithm. Points falling into empty bins generate infinite anomaly scores – hence populating more bins from analyst feedback during the run will have a significant impact on reducing scores of subsequent points by reducing the number of empty bins.

5 Online k-Means

5.1 Overview

k-Means clustering is a well-known simple technique which aims to divide the data set into a specified number k of partitions in such a way as to minimise the "within cluster sum of squares" (WCSS) $\sum_{i=1}^{k} \sum_{x \in S_i} ||x - \mu_i||^2$, where S_i is the ith partition of the data and μ_i is the mean of the data points in S_i.

The common algorithm is a simple refinement algorithm, alternating between assigning points to the nearest of the k cluster centres, and then updating each centre to the mean of its assigned points. Using at least a Euclidean distance, this is guaranteed to converge, though not necessarily to a global optimum. This is known typically as Lloyd's Algorithm (Lloyd 1982).

Initialisation of the cluster centres is typically done by random selection of data points. On the other hand, k-Means++ is an algorithm for choosing the initial values based on selection of points in proportion to probabilities assigned from their distance from already chosen points (Arthur & Vassilvitskii 2007). Motivation for this improvement comes from results such as in (Kanungo et al. 2002), where it is argued that "it is easy to construct situations in which Lloyds algorithm converges to a local minimum that is arbitrarily bad compared to the optimal solution.". (Arthur & Vassilvitskii 2007) shows that k-Means++, on the other hand, gives an expected $O(logk)$ approximation to the cost of the optimal clustering. However, we note that in the online case such point selection is not possible with streaming data. Optimal initial point selection is left for future work. For the current initial investigation, we simply initialise the cluster centres to the first distinct k points in the data.

(King 2012) considers the non-stationary case for online k-Means (OKM). The key point is that the WCCS cost function treats all data points from all times equally. However, in a non-stationary setting, one does not wish to give such equal weight to all time slices. Instead, King proposes the cost function, at time t modified with a weighting δ^{t-i} for some $\delta \in (0, 1)$, where t is the current time and i is the time from which the data point originated. King's proposed modification to Lloyd's Algorithm is simply to update μ_i to $\mu_i + (x - \mu_i)/\alpha$ for some $\alpha \in (1, \infty)$ given a new data point x, as opposed to $\mu_i + (x - \mu_i)/|S_i|$.

Our OKM algorithm is simply an implementation of that of (King 2012), with clusters initialised as discussed above. Anomaly identification is accomplished with a simple threshold on the distance of the point from the nearest cluster. We defer to future work analogs of (Arthur & Vassilvitskii 2007) regarding cluster initialisation and on dynamic selection of the number of clusters k.

5.2 Results

OKM was run on the PREDICT data set with 10 clusters and $\alpha = 1000$, with alert thresholds ranging from 1 to 100. The results are shown in Table 4. We note that analyst feedback has only a minor impact on the results.

6 Latent Dirichlet Allocation

6.1 Overview

LDA is a probabilistic graphical model that assumes a set of latent stochastic processes responsible for generating a set of observed categorical data. A natural setting for LDA is the modelling of corpora, where a datum is a document composed of a bag-of-words,

Algorithm	TPR	FPR	AR	FDR
OKM	0.19	7.7E-05	6.4E-04	0.12
	0.25	5.1E-04	1.3E-03	0.40
	0.31	1.3E-03	2.2E-03	0.58
	0.92	5.7E-03	8.4E-03	0.67
	0.98	3.3E-02	3.6E-02	0.92
	0.99	6.2E-02	6.5E-02	0.95
OKM + analyst feedback	0.19	7.7E-05	6.4E-04	0.12
	0.25	5.1E-04	1.3E-03	0.40
	0.31	1.3E-03	2.2E-03	0.57
	0.92	5.2E-03	7.9E-03	0.65
	0.98	3.0E-02	3.3E-02	0.91
	0.99	6.2E-02	6.4E-02	0.95

Table 4: Results for OKM Algorithm

and where the latent processes are topics with differing probability distributions over the vocabulary. A seminal paper on LDA is by (Blei et al. 2003).

The investigation of LDA for network traffic anomaly detection is a small but emerging research area. (Robinson 2010) implemented LDA in batch-mode, whereby the inferencing was performed via a (collapsed) Gibbs sampling on the entire dataset (some 150K records), and anomalousness was estimated using a (Hellinger) distance metric between all pairs of records. While promising results were reported on the VAST Challenge 2009 network intrusion data, the $O(N^2)$ nature of the anomalousness calculation and the batch-mode inferencing preclude it from the typical online setting.

(Cramer & Carin 2011) use LDA principally for identifying the latent topics in network traffic data to show that Bayesian modelling can approximate human observers well, however in conclusion do note that anomaly detection could be performed by using the (log-)likelihood of new records based on the model trained using normal data. (Ferragut et al. 2011) similarly used a distance metric (Kullback-Leibler divergence) between the topic mixture of a record and that of the average of all other records, in a leave-one-out fashion, and also performed inferencing in batch-mode, though preferring variational inference (expectation maximisation) over sampling. They showed promising results on a dataset of Transmission Control Protocol (TCP) connections entering their laboratory.

(Newton 2012) uses the first half hour of a TCP trace to and from his university network to train an LDA model using VB (presumably in batch-mode) and then estimates the (log-)likelihood of records in the next half hour to determine anomalies, with initial qualitative results showing promise. Finally, (Kasliwal et al. 2014) used data from the KDDCUP99 dataset to train an LDA model in conjunction with a genetic algorithm to detect anomalies with an accuracy of 88% and an FPR of 6%.

Unlike these previous attempts, the objective in our research was to develop an online, probabilistic-based, LDA model for streaming network traffic anomaly detection.

6.2 Mathematical Formulation

The observed data $\boldsymbol{X}$ is an $N \times M$ matrix where x_{ij} is a binary vector of length W, indicating which word (of W words) occupies the $j-$th word position in the i-th datum. We introduce random variables (a) $\boldsymbol{Z}$, a matrix where z_{ij} is also a binary vector, but of length K, indicating which topic (of K topics) is assumed to have generated the $j-$th word position in the i-th datum; (b) $\boldsymbol{\theta}$, an $N \times K$ matrix whose $i-$th row is a vector representing the topic proportions for the $i-$th datum; and (c) $\boldsymbol{\beta}$, an $K \times W$ matrix whose $k-$th row is a vector representing the word proportions for the $k-$th topic. Finally, $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$ are parameters govening the prior distributions for $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$.

The inference problem is to determine the posterior probability distribution $p(\boldsymbol{\theta}, \boldsymbol{Z}, \boldsymbol{\beta}|\boldsymbol{X}, \boldsymbol{\alpha}, \boldsymbol{\eta})$. Variational Inference approximates p with another distribution q conditioned on a set free parameters $\boldsymbol{\Omega}$ whose values are sought to make $q(\boldsymbol{\theta}, \boldsymbol{Z}, \boldsymbol{\beta}|\boldsymbol{\Omega})$ as close to $p(\boldsymbol{\theta}, \boldsymbol{Z}, \boldsymbol{\beta}|\boldsymbol{W}, \boldsymbol{\alpha}, \boldsymbol{\eta})$ as possible. To make this problem tractable we assume q is fully factorisable:

$$q = \prod_{k=1}^{K} q(\boldsymbol{\beta}_k|\boldsymbol{\lambda}_k) \prod_{i=1}^{N} \left(q(\boldsymbol{\theta}_i|\boldsymbol{\gamma}_i) \prod_{j=1}^{M} q(\boldsymbol{Z}_{ij}|\boldsymbol{\phi}_{ij}) \right)$$

and in LDA we assume that $\boldsymbol{\theta}_i$ and $\boldsymbol{\beta}_k$ follow Dirichlet distributions $\boldsymbol{\theta}_i|\boldsymbol{\alpha} \sim Dir(\boldsymbol{\alpha})$ and $\boldsymbol{\beta}_k|\boldsymbol{\eta}_k \sim Dir(\boldsymbol{\eta}_k)$ and that the topic assignment and word selection follow Categorical distributions $\boldsymbol{Z}_{ij}|\boldsymbol{\theta}_i \sim Cat(\boldsymbol{\theta}_i)$ and $\boldsymbol{X}_{ij}|\boldsymbol{Z}_{ij}, \boldsymbol{\beta} \sim Cat(\prod_{k=1}^{K} \boldsymbol{\beta}_k^{Z_{ijk}})$. The variational distributions are assumed to belong to the same families.

Analytically differentiating the resulting objective function and equating to zero allows one to derive a set of nonlinear, coupled equations for the optimal variational parameters, which could be solved by iteration. However, for streaming data it quickly becomes impractical since, as the dataset size N grows, so does the number of local parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\phi}$. Also, the global parameters $\boldsymbol{\lambda}$ would need to be re-optimised as each new datum arrives.

Instead of this batch-mode, we follow stochastic optimisation (Hoffman et al. 2013), whereby for each new datum we take a single step in a gradient ascent for $\boldsymbol{\lambda}$ and solve by iteration for the optimal local parameters for only a mini-batch B of the N documents. Furthermore, we follow (Amari 1982) and use the natural (vice Euclidean) gradient, so that:

$$\boldsymbol{\lambda}_k^{t+1} = \boldsymbol{\lambda}_k^t + \rho_k^t \left(\boldsymbol{\eta}_k - \boldsymbol{\lambda}_k^t + \frac{N}{B} \sum_{i=1}^{B} \sum_{j=1}^{M} \boldsymbol{x}_{ij} \phi_{ijk}^{OPT} \right). \tag{1}$$

For the gradient step-size ρ_k^t we follow (Ranganath et al. 2013) and employ an adaptive learning scheme. For the local variational parameters usual differentiation results in:

$$\phi_{ijk}^{OPT} = \frac{e^{F(\gamma_{ik}) + \sum_{v=1}^{W} x_{ijv} F(\lambda_{kv}) - F(\lambda_{k\cdot})}}{\sum_{k'=1}^{K} e^{F(\gamma_{ik'}) + \sum_{v=1}^{W} x_{ijv} F(\lambda_{k'v}) - F(\lambda_{k'\cdot})}} \tag{2}$$

and

$$\gamma_{ik}^{OPT} = \alpha_k + \sum_{j=1}^{M} \phi_{ijk} \tag{3}$$

where $\lambda_{k\cdot} = \sum_{v=1}^{W} \lambda_{kv}$.

Collectively, Equations (1 – 3) detail the online LDA model. To compute the anomalousness of each datum as it arrives, we estimate the probability of observing it based on the current state of the online LDA model. We use the current variational parameters for the vocabulary distribution $\boldsymbol{\lambda}$ along with the word indicators x_{ijv} of the incoming datum to compute the topic proportions variational parameters $\boldsymbol{\gamma}_i$

(using Equations (2,3)) and then normalise these to compute estimates $\hat{\theta}_{ik}$ and $\hat{\beta}_{kv}$ respectively. We then estimate the per-word log probability of the incoming datum as:

$$p_i = \frac{\sum_{j=1}^{M}\sum_{v=1}^{W} x_{ijv} \log\left(\sum_{k=1}^{K} \hat{\theta}_{ik}\hat{\beta}_{kv}\right)}{\sum_{j=1}^{M}\sum_{v=1}^{W} x_{ijv}}.$$

6.3 Implementation

For LDA, we focus only on the categorical features of *port_respondent*, *service*, *connection_state* and *history* (and *protocol* although there were only five non-TCP records in the dataset) to contrast with the numerical-based algorithms above, and with a view to subsequent ensembling (see our companion paper (Abraham et al. 2016)).

We developed a Python implementation of the above model by modifying the code at cs.princeton.edu/∼blei/downloads/onlineldavb.tar to incorporate the adaptive learning gradient step-size and the per-word log probability. The number of topics K and mini-batch size B were set at 10 and 5, respectively. The prior parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$ were set as constants $1/K$. The mini-batch comprised the incoming record and a random sample of $B-1$ records from the past data stream.

The model was learned on an initial stream of $10,000$ network traffic data, after which each incoming record was assessed against $p_i < mp^*$ where p^* was the median p_i value of the training data and where m is a user-supplied factor to control the sensitivity of the detector. If the incoming record was flagged as anomalous it was discarded from the model update, unless analyst feedback was assumed whereby a false positive would be included.

6.4 Results

One of the features of LDA is its ability to provide qualitative insight on the generative process underlying the network traffic. The learned topics corresponded well with protocol types such as Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP) and Secure Sockets Layer (SSL), supporting the validity of the approach. The results of applying the LDA model to the PREDICT dataset are given in Table 5. The impact of analyst feedback is mixed, with improved FPR (and lower AR) at the highest TPR levels only.

The minimum FDR occurs at lower TPR levels. For example, if the analyst could handle an AR of around two-per-thousand, then on average, one of those two alerts would be a false alarm and the other would be an anomaly. If that rate was too fast, then it could be slowed, for example to around five-per-ten-thousand, but on average, four of those alerts would now be false alarms while the other would be an anomaly.

7 Conclusion

A variety of UML techniques were trialled, spanning clustering-based (k-Means), distance-based (k-NN), density-based (HBOS), and statistical-based (GMM and LDA). While the results presented here are limited to a single test set, in general we found differences in the degree of tuning required (e.g. k-Means and k-NN both need to define k and the decision threshold,

Algorithm	TPR	FPR	AR	FDR
LDA	0.03	4.0E-04	4.9E-04	0.83
	0.50	9.8E-04	2.5E-03	0.39
	0.72	2.6E-03	4.8E-03	0.55
	0.95	2.4E-02	2.7E-02	0.89
LDA + analyst feedback	0.28	1.1E-03	2.0E-03	0.57
	0.65	1.8E-03	3.7E-03	0.48
	0.72	2.2E-03	4.4E-03	0.51
	1.000	1.2E-02	1.5E-02	0.80

Table 5: Results for LDA Algorithm

Algorithm	TPR	AR	FDR	Score
GMM	0.99	5.1E-03	0.41	0.47
LDA	0.50	2.5E-03	0.39	0.33
OHBOS	0.48	2.9E-03	0.50	0.24
OKNN	0.19	8.9E-04	0.36	0.18
OKM	0.19	6.4E-04	0.12	0.26

Table 6: Summary Results for all Algorithms

and very much depend on the dataset and, moreover, the features chosen and their normalisation) and computational complexity (e.g. GMM, LDA, and to an extent k-NN required significantly more computing resources than HBOS or k-Means).

Each of the five algorithms investigated could correctly classify (virtually) all of the malicious traffic in the PREDICT dataset examined with sufficiently relaxed thresholds, or detect a proportion of them with more favourable values of FPR, AR and FDR. In practice, in real world data the number of true positives is unknown and it is the AR and FDR that have particular meaning. To estimate a 'sweet spot' for each method, we identified the single row in each of Tables (1 - 5) which maximises a score comprised of the product of the F_β-score (with $\beta = \frac{1}{3}$ to give more weight to FDR than TPR) and the Matthews correlation coefficient. The former metric balances TPR and FDR while the latter balances TPR and AR, and both are robust to class-imbalanced datasets. Table 6 displays the results (where the non-analyst feedback version of the algorithms was used). We note that the five algorithms partition into three regimes of TPR. At the highest TPR, GMM appears most suited though at the expense of relatively high AR and FDR – though these effects can be mitigated through the use of ensembling (see (Abraham et al. 2016)). Taking a more pragmatic approach, one may sacrifice TPR for improved efficiency via a lower FDR, *e.g.* use of LDA and OHBOS at mid-ranges of TPR, and OKNN and OKM at low-ranges of TPR.

Future work will focus on (a) forming a better understanding of how to incorporate analyst feedback in dynamic updating of UML-based algorithms; (b) utilising big data platforms to approach real-time performance (many of the algorithms are ripe for parallel processing); (c) enhanced feature engineering (both categorical and numerical) of computer network traffic data; and (d) algorithmic enhancements such as dynamic k computation for OKM or use of non-Euclidean distances for other algorithms.

References

Abraham, T., Gill, A. & Montague, P. (2016), Ensembles to control outlier detection rates in network traffic data, *in* 'Proceedings of the Fourteenth Australasian Data Mining Conference', CRPIT, Canberra, Australia.

Aitkin, M. & Wilson, G. T. (1980), 'Mixture models, outliers, and the EM algorithm', *Technometrics* **22**(3), 325–331.

Altman, N. S. (1992), 'An introduction to kernel and nearest-neighbor nonparametric regression', *The American Statistician* **46**(3), 175–185.

Amari, S.-I. (1982), 'Differential geometry of curved exponential families-curvatures and information loss', *The Annals of Statistics* **2**, 357–385.

Arthur, D. & Vassilvitskii, S. (2007), k-means++: The advantages of careful seeding, *in* 'Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms', Society for Industrial and Applied Mathematics, pp. 1027–1035.

Bhuyan, M. H., Bhattacharyya, D. K. & Kalita, J. K. (2014), 'Network anomaly detection: Methods, systems and tools', *IEEE Communications Surveys Tutorials* **16**(1), 303–336.

Blei, D. M., Ng, A. Y. & Jordan, M. I. (2003), 'Latent dirichlet allocation', *J. Mach. Learn. Res.* **3**, 993–1022.

Carvalho, C., Johannes, M. S., Lopes, H. F. & Polson, N. (2010), 'Particle learning and smoothing', *Statistical Science* **25**(1), 88–106.

Carvalho, C. M., Lopes, H. F., Polson, N. G., Taddy, M. A. et al. (2010), 'Particle learning for general mixtures', *Bayesian Analysis* **5**(4), 709–740.

Cramer, C. & Carin, L. (2011), Bayesian topic models for describing computer network behaviors, *in* '2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)', IEEE, pp. 1888–1891.

Ferragut, E. M., Darmon, D. M., Shue, C. A. & Kelley, S. (2011), Automatic construction of anomaly detectors from graphical models, *in* 'Computational Intelligence in Cyber Security (CICS), 2011 IEEE Symposium on', IEEE, pp. 9–16.

Forster, K., Monteleone, S., Calatroni, A., Roggen, D. & Troster, G. (2010), Incremental knn classifier exploiting correct-error teacher for activity recognition, *in* 'Machine Learning and Applications (ICMLA), 2010 Ninth International Conference on', IEEE, pp. 445–450.

Goldstein, M. & Dengel, A. (2012), 'Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm', *KI-2012: Poster and Demo Track* pp. 59–63.

Hoffman, M. D., Blei, D. M., Wang, C. & Paisley, J. (2013), 'Stochastic variational inference', *Journal of Machine Learning Research* **14**, 1303–1347.

Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R. & Wu, A. Y. (2002), A local search approximation algorithm for k-means clustering, *in* 'Proceedings of the eighteenth annual symposium on Computational geometry', ACM, pp. 10–18.

Kasliwal, B., Bhatia, S., Saini, S., Thaseen, I. & Kumar, C. (2014), A hybrid anomaly detection model using g-lda, *in* 'Advance Computing Conference (IACC), 2014 IEEE International', IEEE, pp. 288–293.

King, A. (2012), 'Online k-means clustering of nonstationary data', *Prediction Project Report* . **URL:** *http://ocw.mit.edu/courses/sloan-school-of-management/15-097-prediction-machine-learning-and-statistics-spring-2012/projects/MIT15_097S12_proj1.pdf*

Lloyd, S. (1982), 'Least squares quantization in pcm', *IEEE transactions on information theory* **28**(2), 129–137.

Merl, D. (2009), Advances in Bayesian model based clustering using particle learning, Technical report, Technical Report LLNL-TR-421078, Lawrence Livermore National Laboratory.

Murphy, K. P. (2007), 'Conjugate Bayesian analysis of the Gaussian distribution'. **URL:** *http://www.seas.harvard.edu/courses/cs281/papers/murphy-2007.pdf*

Newton, B. (2012), 'Anomaly detection in network traffic traces using latent dirichlet allocation'. **URL:** *http://www.cs.unc.edu/bn/BenNewtonFinalProjectReport.pdf*

Ngan, H. Y., Yung, N. H. & Yeh, A. G. (2012), Modeling of traffic data characteristics by dirichlet process mixtures, *in* '2012 IEEE International Conference on Automation Science and Engineering (CASE)', IEEE, pp. 224–229.

Odell, P. L. & Feiveson, A. H. (1966), 'A numerical procedure to generate a sample covariance matrix', *Journal of the American Statistical Association* **61**(313), 199–203.

Ramaswamy, S., Rastogi, R. & Shim, K. (2000), Efficient algorithms for mining outliers from large data sets, *in* 'ACM SIGMOD Record', Vol. 29, ACM, pp. 427–438.

Ranganath, R., Wang, C., Blei, D. & Xing, E. (2013), An adaptive learning rate for stochastic variational inference, *in* S. Dasgupta & D. Mcallester, eds, 'Proceedings of the 30th International Conference on Machine Learning (ICML-13)', Vol. 28, JMLR Workshop and Conference Proceedings, pp. 298–306.

Robinson, D. G. (2010), *Statistical language analysis for automatic exfiltration event detection.* **URL:** *http://www.osti.gov/scitech/servlets/purl/983675*

Scheper, C., Cantor, S. & Karlsen, R. (2009), Trusted distributed repository of internet usage data for use in cyber security research, *in* 'Conference For Homeland Security, 2009. CATCH '09. Cybersecurity Applications Technology', pp. 83–88.

Yamanishi, K., Takeuchi, J.-I., Williams, G. & Milne, P. (2004), 'On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms', *Data Mining and Knowledge Discovery* **8**(3), 275–300.

Ensembles to Control Outlier Detection Rates in Network Traffic Data

Tamas Abraham **Andrew Gill** **Paul Montague**

Defence Science and Technology Group
Edinburgh, Australia
Email: {tamas.abraham, andrew.gill, paul.montague}@dsto.defence.gov.au

Abstract

This paper describes an approach to find outliers in network traffic to provide operators of network perimeter defence software with observations to analyse for maliciousness. Outliers found by our scheme may not correspond to actual harmful traffic, though we aim to maximise this probability with appropriate feature selection. We rely on operator feedback to determine if an observation is useful, and use this to adjust subsequent findings.

We use machine learning algorithms on unlabelled network traffic data and ensembles to select candidates for evaluation by human operators. The rate at which observations are presented is controlled to suit operator availability. We discuss the constraints and assumptions made on the data and the operating environment, and present experimental results. Detailed analysis of individual system components is outside the scope of this paper.

Keywords: Outlier detection, Machine learning algorithms, Ensemble methods, Intrusion detection systems, Network security

1 Introduction

Network perimeter defence is a typical and often large-scale activity in government and business to protect infrastructure and data belonging to the organisation from outside influences/internal threats. Intrusion detection systems (IDSs) (Lunt et al. 1989) are a constantly evolving but mostly mature component of this effort aimed at identifying malicious activity within network traffic traversing through the organisation. Knowledge-based IDSs recognise known patterns amongst the data and alert when they are encountered. Behaviour-based IDSs build models of known good traffic and alert when deviations from the model are observed.

In this paper, we describe a behaviour-based system that finds outliers in network traffic that has already passed through existing IDSs used by the organisation. Its components (see Figure 1) include:

- Machine learning algorithms to identify outliers.
- Ensembles to combine results to locate outliers that are supported by multiple algorithms.

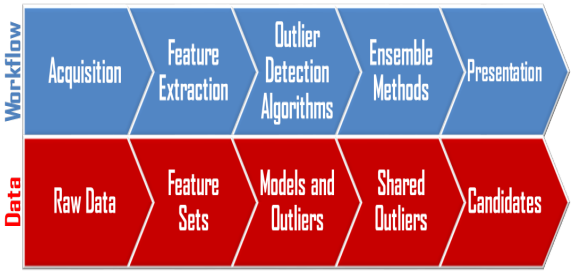


Figure 1: System workflow and data.

- Presentation of selected contents to an operator for evaluation and feedback into the system.

The result is a highly modular, extensible and parameterisable solution aimed at minimising (but not eliminating) human participation and maximising the potential for discovering unusual, potentially malicious traffic. Operator feedback can result in the filtering of subsequent benign findings, or the conversion of newly discovered harmful content into signatures for existing IDSs.

2 Background

Network intrusion detection systems, in particular those based on signatures (Roesch 1999), are robust and effective tools that block undesired traffic from reaching their intended destination. Knowledge-based IDSs have an inherent weakness of not being able to deal with previously unseen malicious traffic, leaving systems unprotected until a signature is created and deployed. Behaviour-based IDSs, on the other hand, are designed to catch unknown harmful traffic by detecting deviations from what is considered normal, presenting opportunities for improving the state-of-the-art despite a range of solutions already proposed (Tzur-David 2011).

At the core of behaviour-based IDSs are machine learning algorithms responsible for building models of standard activities and identifying anomalies. Outlier detection (Hawkins 1980) is a well-studied field particularly suited for this task, employing various statistical measures (based on distribution, density, distance and so on) to calculate the difference of an individual data point from others (Kriegel et al. 2010). Basing a solution on a single outlier detection algorithm, however effective it may be for a given data and feature set combination, can reduce the flexibility of a system when used across multiple data sets/domains. Ensemble learning (Opitz & Maclin 1999) can address this concern by considering results from multiple sources. Of particular interest to us are

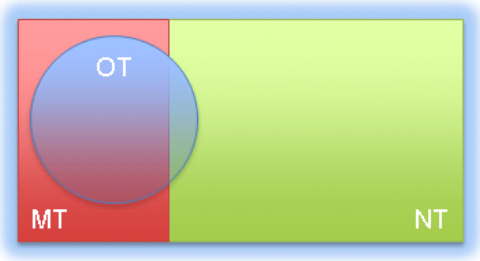


Figure 2: Relationship between traffic types.

methods that can be employed for unlabelled network data (Zimek et al. 2014), the focus of our study.

Our goal is to use outlier detection methods together with ensemble techniques to aid the work of network defence software operators. We consider our solution complementary to existing tools rather than in competition with them. The criteria that are relevant and specific to our scenario are discussed below.

2.1 Data

In our work, we assume that data has already been checked and cleaned by other software: we are looking for any residual malicious traffic that was missed by existing tools. A single additional confirmed find will therefore be an improvement on what was achieved with current defensive tools. The ramification of this is significant: it means that there is no minimum requirement on the sensitivity of the final output of our approach, since finding just one remaining malicious data record is a better outcome than before. Nevertheless, our goal will be to capture as much of the residual malicious traffic as possible.

We also assume that at least some of the residual malicious traffic can be differentiated from regular traffic as outliers from an established norm utilising machine learning techniques. In practice, this entails extracting appropriate data features that capture this difference. Note that we do not automatically treat outliers as malicious, but expect that if there are some, then those that are more markedly different from the norm will be more likely candidates. Finally, we assume that the malicious traffic is much smaller in volume than regular traffic in our data as a result of its prior cleaning by IDSs.

If we divide our data into two sets, the residual malicious traffic (MT, $|MT| \geq 0$) and normal traffic (NT, $|NT| > 0$), an outlier detection algorithm will produce a third set, outlier traffic (OT, $|OT| \geq 0$). We can summarise our assumptions above as follows: $|MT| << |NT|$ because of prior cleaning of the data; $MT \cap OT \neq \emptyset$ because we assume that at least some malicious traffic is found amongst outliers. Figure 2 illustrates a close-to-optimal relationship between the three traffic types, with the sizes of MT and OT exaggerated for better visualisation. We also expect that algorithms will generate outliers that are actually normal traffic, and some malicious traffic to be missed.

2.2 Labels

The data available to us for real-life operation contains no labels, leaving unsupervised techniques as candidates for outlier detection. Semi-supervised algorithms may also be used as results from the operator evaluation of candidate outliers are made available. Operator feedback is also useful to measure the effectiveness of our system, both individual algorithms and ensembles. We want to select methods that promise some level of success, and we can either do this by testing and fine-tuning on labelled data when available, or utilising feedback from live operations to assess the usefulness of components.

2.3 Real-time

One of the important considerations when defending a network is the timeliness of the reaction to threats, often requiring (near) real-time identification of malicious traffic. Such speed of discovery is best achieved by using online algorithms for outlier detection in our solution as opposed to batch processing of data.

To further accelerate the rate of processing, we propose to restrict the use of state information from data/traffic. This limits the type of features we can extract from our data. No high-level aggregate network information is collected, and we also leave the use of background knowledge for future experiments.

2.4 Operator resources

One of the drivers behind automating the processes in network security operations is the high cost and often low availability of human operators. Their time may be limited for evaluating excessive numbers of candidates of potentially malicious traffic. This suggests that our solution not only needs to be timely but the rate at which candidates are produced should also be adjustable to suit operator engagement levels.

We stated earlier that we do not need to capture all malicious traffic to be successful: we also cannot spend too many resources to find them. We therefore need to make sure that the candidates we offer to operators have a high likelihood of being interesting. This would be best achieved if the models generated by outlier detection algorithms produced outliers that corresponded mostly to malicious traffic. (Although, note that we do not qualify the level of risk posed by different malicious traffic.) To state it in supervised learning terms, this is the same as saying that in the candidate set we would like to have high precision, or equivalently, minimise the false discovery rate (FDR). In practice, we do not expect any single outlier detection algorithm to produce an acceptable model. Instead, we use ensembles to create result sets that generate candidates at a desired rate and combine results from individual algorithms to find the outliers most likely to be malicious.

2.5 Challenges

Having highlighted some of the considerations and difficulties we face, we pose the following questions to be considered by our implementation, along with the approaches we plan to address them (followed in parentheses):

- How do we decide what constitutes an outlier? (a feature and threshold selection problem)
- How can we ensure that the accuracy of individual algorithms is satisfactory? (an algorithm tuning problem)
- How can we best utilise results from multiple algorithms to improve the results from individual algorithms? (an ensemble problem)
- How do we keep the list of results to a size suitable for operators? (a result reduction problem)

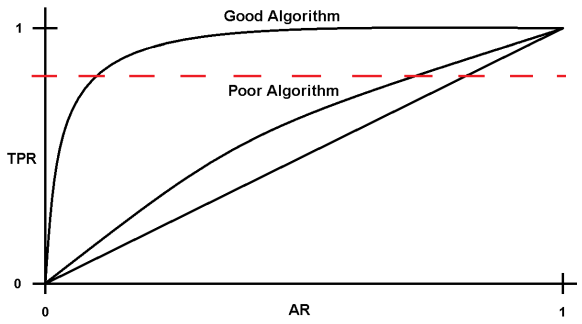


Figure 3: Alert rates of different algorithms.

A recent paper (Veeramachaneni et al. 2016) proposes a similar system to ours. It utilises outlier detection, ensembles, operator feedback, and includes supervised learning components, the use of complex features and batch processing to update models. Our approach misses some of these elements in order to achieve our (near) real-time goals.

3 Implementation

Given our assumptions in Section 2.1, we use multiple outlier detection algorithms to generate their version of OT. Each algorithm produces outliers at a prescribed alert rate (AR) using a selected set of features. We next examine the relationship between the resulting OT and MT, the residual malicious traffic (OT would ideally be equal to MT).

Each algorithm is likely to produce both false positives (false alarms, normal traffic shown as outliers) and false negatives (misses, malicious traffic that are not outliers). Our goal is to maximise the malicious records in individual results, even at the expense of large number of false alarms, as this yields ensembles that are also likely to include them. In other terms, we would like to achieve high sensitivity, or true positive rate (TPR), producing OTs that each cover much of MT, even if it means an associated high false positive rate (FPR), or overlap between OT and NT.

Identifying such high TPR algorithms can be difficult in an unsupervised learning environment. Growing the alert rate, i.e. producing more outliers, is likely to result in more hits, but also in an increased false alarm rate. Excessively growing the size of OT may thus be counter-productive.

For some algorithms, however, it may be possible to empirically establish an AR (the percentage of predictions amongst all records) where the TPR of predictions is adequate for processing in ensembles. Figure 3 illustrates how difficult finding such rates could be by depicting the TPR curves of different algorithms against a desired minimum (indicated by the dashed line). Random guessing, shown as the diagonal line, and poor algorithms would be insufficient contributors when the goal is to avoid producing large amounts of false positives. Later in the paper we will discuss some of the other factors that may influence the success rate of outlier detection algorithms.

3.1 Features

Feature selection has been an integral part of preprocessing data along with other tasks such as dealing with errors, missing data, noise and so on (Frawley et al. 1992). Increases in data sizes and data dimensionality (i.e. the number of features) make it even more important to match features with learning tasks, resulting in the engineering of features gaining more focus (Guyon et al. 2006).

Network data features have been studied extensively (Auld et al. 2007, Williams et al. 2006). The number of features generated by various proposals can run into the hundreds, even thousands, which gives sufficient choice for selecting appropriate subsets for processing. In our work, we avoid some of the more complex features to keep our online learning activity timely (see Section 2.3), leaving us to exclude otherwise useful aggregate features used in other systems (Veeramachaneni et al. 2016).

We do, however, include basic statistics from individual connections like transmission sizes and information from packets and protocol headers. Content is not used, although some meta-data may be converted into features. Importantly, we exploit both quantitative (numeric) and qualitative (categorical) features through algorithms specifically suited for each feature type and through their ensemble combination.

We also consider keeping the number of features small for individual algorithms. This not only helps with the speed of evaluation, but for some algorithms, it can reduce the noise introduced by the addition of more features. Existing approaches like those above often also propose smaller feature sets for their analysis. These sets have many common elements, and they served as good points of comparison for the features in our work. We also created new features to suit specific outlier detection algorithms in our collection. Where possible, we analysed the performance of individual features using off-line sample data sets and removed less important ones to optimise feature sets. Unfortunately, a detailed discussion of feature set results is beyond the space limits of this paper.

3.2 Algorithms

We put no limit on the number of different algorithms contributing towards candidate generation in our solution. The same algorithm may be used multiple times with different feature sets and parameters. We do want our algorithms to satisfy some requirements, however, like the ability to evaluate near real-time, produce a quantitative measure for outlierness and incorporate feedback from operators. Online algorithms (Bottou 1998), which train with an initial set of streaming data and optionally update models as data is evaluated, are best suited for these purposes. We produced a small set of online algorithms from existing solutions to incorporate into our system. Of these, a version of Latent Dirichlet Allocation (LDA) (Blei et al. 2003) was used mostly with categorical attributes, whilst numerical attributes were processed with Gaussian Mixture Models (GMM) (Murphy 2012) and online variants of other well-known machine learning algorithms such as k-Means and k-Nearest Neighbors (Gill & Montague 2016).

For further comparison, we experimented with off-line algorithms from the Anomaly Detection Extension (ADE) (Amer & Goldstein 2012) of the popular RapidMiner (Mierswa et al. 2006) data mining toolkit. This extension contains implementations of a number of well-known outlier detection algorithms, such as INFLO, COF, LoOP, LOCI, and some others like HBOS and rPCA, all operating in batch-mode.

Figure 4 shows the relative performance of some of the algorithms[1] we experimented with on sample

[1]The algorithm names, and in later figures, the ensemble components, are not significant for demonstrating our points, although for our experiments with real data we used our online set of algo-

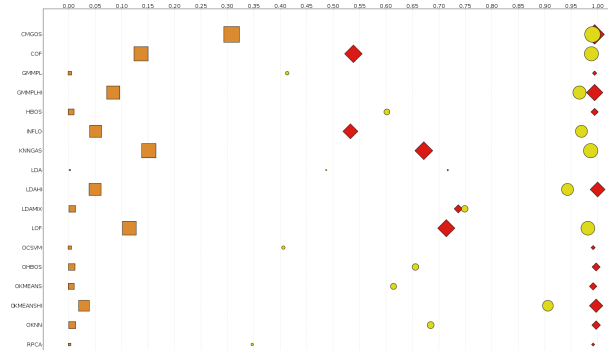


Figure 4: False positive (square), true positive (diamond) and false discovery (circle) rates (0 to 1 on x-axis) of various outlier detection algorithms (y-axis).

labelled data sets. The three different shapes represent the false positive (orange square), the true positive (red diamond) and false discovery (yellow circle) rates, with the size of the shapes indicating the number of predictions made by each algorithm. There are substantial differences between the results achieved, and in our early experiments we used figures like this to identify algorithms that were better suited for our purposes. The more desirable algorithms in the figure are those with their TPR diamond showing towards the extreme right, close to a 100% hit rate, and an FDR circle far to their left, indicating a high proportion of outliers as true positives. Note that smaller-sized shapes often satisfy this best, since fewer predictions at high TPR tend to include fewer false positives and thus yield a lower FDR.

3.3 Ensembles

Ensembles have been used in machine learning to improve on predictions in classification (Rokach 2010) and in unsupervised learning (Vega-Pons & Ruiz-Shulcloper 2011). Of the various ensemble approaches, we chose to combine results from multiple algorithms that performed their learning activity on the same data in an independent model-centric way, rather than sequentially (Aggarwal 2013). This ensures that individual algorithms have a degree of freedom in selecting their own parameters, relevant features, scoring mechanism and method to incorporate feedback from operators. Using different learning approaches can produce uncorrelated predictions and increase the coverage of candidate data in the combined outlier sets. By selecting appropriate ensemble combination methods, specific user goals may be targeted, such as the reduction of false positives. Load-sharing by dividing up features between algorithms can be used to satisfy operating speed requirements.

Ensemble member algorithms are expected to score every record to provide a means to determine whether an individual record is an outlier or not. This score, along with additional knowledge such as a threshold, could be used by the ensemble process to calculate the combined outlier likelihood of a record. Often, a common representational format is used to reconcile scores from various algorithms. Score types, methods to transform, normalisation and various combination techniques are to be considered (Kriegel et al. 2011). For the latter, voting, score averaging and using weights to scale the contribution of individual algorithms are available options. Complex fusion methods such as belief functions and knowledge behaviour spaces can be considered. One of the simplest examples is vote accumulation, the counting of the number of algorithms in the ensemble that flag a record as an outlier.

In online learning, where individual models may be updated as new data is evaluated, outlier threshold values may shift over time. This makes it more practical for each algorithm to determine the outlier status of a record and pass it to the ensemble process, rather than deferring this decision to the ensemble.

rithms, i.e. modified LDA, GMM, k-Means, k-NN and HBOS.

4 Experiments

We performed various experiments to test and verify the design decisions we made in our proposed solution. First, we checked that individual components performed as intended (e.g. using data sets from the UCI Machine Learning Repository), then made sure that the system as a whole was able to deliver candidate predictions using ensembles. Once this was established, we tested with additional data sets.

This section describes two separate experiments. In the first, we used labelled data in order to verify our methodology and to learn about the performance of our solution. Labels were used to gain some a priori knowledge about the data and to verify our results, but not for training and scoring. For this task, we selected data available in the Protected Repository for the Defense of Infrastructure against Cyber Threats (PREDICT) (Scheper et al. 2009) data repository. In the second experiment, we used unlabelled data to test our capability in a deployment-like situation, consisting of a large set of captured network traffic passing through an organisational boundary.

4.1 Feature Extraction

From data that was represented as raw network traffic in pcap (packet capture) format, we extracted features using Bro (Paxson 1998). Whilst Bro is primarily a network security monitoring solution, we only used its feature extraction capability to source our data attributes. We used some of the default features as provided by the software; for others we wrote our own Bro scripts. We created additional features by post-processing some of Bro's output (e.g. binning, extracting substrings).

Features were extracted for various application protocols as well as general TCP/IP connections, letting us experiment with different feature sets. Of these features, we selected subsets with categorical attributes to suit LDA and numerical attributes for the remaining algorithms. Separate sets were chosen for TCP/IP connection data and HTTP traffic. For our labelled data set, we used Weka (Holmes et al. 1994) to evaluate features and chose ones nominated by its various attribute selection algorithms.

4.2 Training

Our design calls for the use of online algorithms on unlabelled data. In practice, this means that after an initial period of training, subsequent records are assigned a score to indicate how different they are from the norm. This value is compared to a threshold, calculated during training, to establish whether the record is to be an outlier candidate. If the record is deemed normal, it may also be used to enhance the existing model.

The strategy to determine and update the outlier threshold is left to the individual algorithms. For example, if the assumption is that the training data is

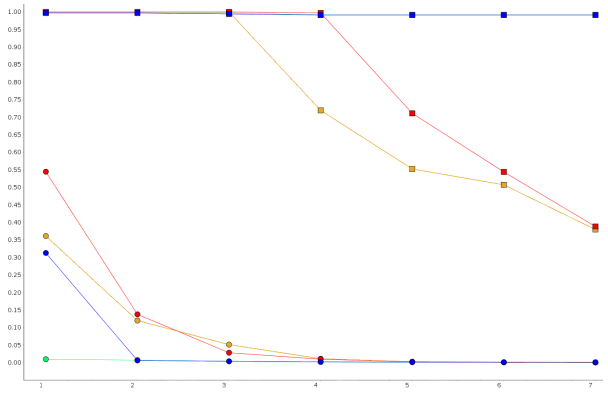


Figure 5: True (top group of lines) and false (bottom group) positive rates (y-axis) of selected ensembles.

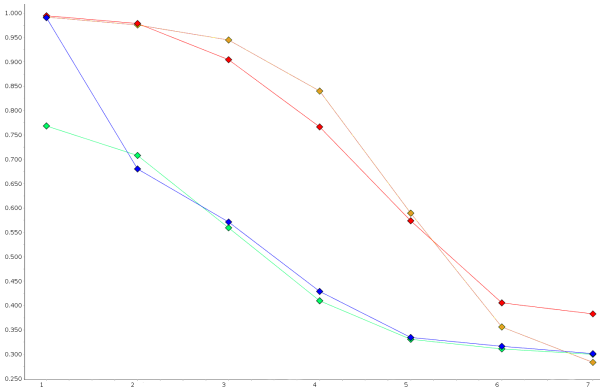


Figure 6: False discovery rates (y-axis) of ensembles versus the contributing algorithms count (x-axis).

normal, then the maximum score seen during training can be elected as the threshold. If we can approximate the percentage of malicious content, we can choose a cut-off value that corresponds to this percentage in the set of scores encountered during training. Statistical analysis of training scores can also be used to select thresholds. For example, if a small set of scores are found to be sufficiently higher than the rest, the minimum value of this set may be nominated. Finally, if we have no knowledge of the balance of normal and malicious content in the data, we may simply choose a method that generates candidates for ensemble processing at a desired alert rate.

We validated training on sample data sets. Because the size of the data during these tests was known a priori, training could be done both on a fixed number of records or on a percentage of the sample size. In real-world situations, a recommended approach would be to train for a certain amount of time instead. If the volume of traffic in a given time period is approximately known, size-based training may still be a suitable alternative.

4.3 Ensemble Combinations

We experimented with various techniques to produce ensembles as described in Section 3.3. We found that vote accumulation performed similarly when compared against more complex variants, and the resulting simplicity and speed made this a desirable candidate for our evaluation purposes. This meant that we tasked our online learning algorithms with delivering simple 'yes' or 'no' decisions to state the outlier status of each record to the ensemble process.

We use the following method to calculate statistics for ensembles. Let the size of ensemble E be n and the size of the test data sample S be m. Each algorithm in the ensemble individually flags every record as either normal or as an outlier. With each record $r_i \in S$, $i = 1, \ldots, m$, we associate a count c_i: the number of algorithms in the ensemble that nominate the record as an outlier, $c_i \in \{0, \ldots, n\}, \forall i$. For each $k \in \{1, \ldots, n\}$ we then produce a set O_k by finding records $r_i \in S$ with $c_i \geq k$, $i = 1, \ldots, m$, and nominate them as outlier candidates at level k for the ensemble. The remaining records in S at each level k will be designated normal traffic, even if they had multiple (but less than k) nominations, $N_k = S \setminus O_k$. Thus, for each ensemble of size n we compute n sets of new statistics for evaluation purposes.

As we increase the requirement for more algorithms to vote a record an outlier, we potentially decrease the number of qualifying candidates, expressed as $O_{k+1} \subseteq O_k$. Since this results in a non-increasing (monotonically or weakly decreasing) sequence of candidate set sizes, global indicators such as true and false positive rates will also follow the same pattern. With the appropriate selection of outlier detection algorithms, the latter can be made to diminish quicker than the former and thus the false discovery rate will improve to our benefit. Furthermore, the decrease in candidate set sizes at higher k can be taken advantage of when targeting a desired alert rate.

4.4 Labelled Data Tests

The data in PREDICT contains labelled traffic, which we used to simulate MT and NT, the residual malicious and normal traffic of our presumed scenario. We tested on several sample sizes and different data types (connection and HTTP data), various features and multiple malicious content percentages, ranging from very low ($<<$ 1%) to up to 10%. We applied our outlier detection algorithms to this data in an unsupervised fashion, generating models and outlier sets (OT, see Figure 2) without the use of labels. We could then evaluate the performance of the detection algorithms and their ensembles by observing the positive and negative labels in the resulting OT.

The labelled experiments helped us:

- Check outlier candidates nominated by individual algorithms against actual malicious traffic. (tune individual algorithms)
- Verify the effectiveness of ensemble combinations. (choose ensemble combination method)
- Investigate the relationship between ensemble results and malicious records. (analyse data)
- Develop ensemble strategies. (judge alert rates)

4.5 Observations from Labelled Tests

Re-examining Figure 4, we can make some statements on individual algorithms we used in our labelled data experiments. Some of the algorithms performed remarkably well, identifying a high percentage of malicious traffic as outliers and reporting with few errors. Other algorithms managed to maintain high sensitivity (TPR) but at the expense of reporting high levels of normal traffic as outliers. For almost all the samples we used from the PREDICT data set, our modified online algorithms from Section 3.2 belonged to one of these two categories. Some of the other algorithms we tested from the ADE in RapidMiner,

Table 1: Candidate outlier set statistics for various ensembles on a sample with 353 malicious records

Ensemble	$\|O_1\|$:TP	$\|O_2\|$:TP	$\|O_3\|$:TP	$\|O_4\|$:TP	$\|O_5\|$:TP	$\|O_6\|$:TP	$\|O_7\|$:TP
A (2 members)	23956:197	8427:181	-:-	-:-	-:-	-:-	-:-
B (2 members)	1189:353	419:252	-:-	-:-	-:-	-:-	-:-
C (3 members)	36712:352	736:351	507:350	-:-	-:-	-:-	-:-
D (4 members)	42542:254	14059:249	6050:188	1408:176	-:-	-:-	-:-
E (6 members)	37030:352	1101:352	808:351	607:350	518:350	502:350	-:-
F (6 members)	31877:353	3267:353	786:353	567:352	291:188	179:134	-:-
G (7 members)	1520:352	1205:352	799:352	593:350	523:350	508:350	500:350
H (7 members)	64195:353	16524:353	3692:353	1507:352	589:251	323:192	222:137

however, were not as successful. A few examples for this can be seen in Figure 4 where the red diamonds (TPRs) are towards the middle on the x-axis, indicating low hit rates. Overall, for this data set, we were able to achieve a high level of alignment between the results from our outlier detection algorithms and the malicious traffic present in the data. This helped us decide on which algorithms to use in ensembles and on the features to employ for individual algorithms.

Figures 5 and 6 illustrate how result characteristics change when some of these algorithms are combined together in ensembles. To show this, we chose a representative set of four ensembles of seven algorithms, their data displayed on coloured lines. Using the method in Section 4.3 to generate outlier candidate sets $O_k, k = 1, \ldots, 7$, we calculated separate statistics for each k along the x-axis. Figure 5 displays TPRs (top of the figure) and FPRs (bottom of the figure) along the y-axis, Figure 6 displays FDRs. In Figure 5, the values decrease as k increases as expected (see our discussion in Section 4.3). For a couple of ensembles, the TPR remains high even at the extreme end, indicating that all seven algorithms were highly effective at finding most of the malicious content as outliers. For the other two ensembles, the lines start to dip as we move towards higher k, which indicates the behaviour when the intersection of hits found by less effective member algorithms becomes smaller. False positive rates, on the other hand, show an early and marked drop in all cases and continue to decrease as the required number of contributing member algorithms rises.

The positive effect of ensemble combination can be seen in Figure 6. It shows that as k increases the FDR dips to levels that we may consider acceptable in candidate sets recommended to operators. Throughout our tests, we consistently saw similar results when creating ensembles for various data samples.

What the figures do not show, is the rate of decrease in the size of O_k as k increases. We often observed an order of magnitude decrease in the candidate outlier set size from k to $k+1$ for low k. Table 1 lists some combinations generated for one of our tests with 353 malicious records in the data sample. Each column shows the number of candidates generated for various k and the actual hits as ($|O_k|$:TP). The numbers support our earlier observation about decreasing prediction set sizes, and also reveal important information about the performance of member algorithms. One interesting observation is that while our algorithms may make many false predictions, these mistakes are typically different across algorithms. As k increases, this results in fewer false alarms as more algorithms are required to agree, giving an improved, smaller FDR.

The statistics in Table 1 include some extremes. Ensembles *A* and *D* show combinations of poorly performing algorithms with results not fit to be presented to an operator as the ratio of true positives to predictions is too low. For ensemble *A*, the TPR of the two member algorithms combined is below 56% (197/353), which is poor. In contrast, ensembles *C*, *E* and *G* contain high-performing algorithms that produce almost all malicious records even at maximum k. The latter two also make less than one mistake in two from $k \geq 4$, which is excellent. Ensembles *B*, *F* and *H* represent the more commonly observed experimental outcomes. Most member algorithms perform well with high individual sensitivity, whilst making an acceptable amount of false predictions. For higher k the TPR of these ensembles start to drop, yet their FDR continues to improve.

The decrease in candidate set sizes and the improved FDRs at higher k suggest that we should be able to satisfy another one of our project requirements, that of a desired alert rate. The examples of Table 1 show that with an appropriate choice of algorithms, ensemble size and k, a suitable AR with an acceptable FDR can be achieved.

4.6 Alert Rate Experiment

Our final study with labelled data was to mimic the operation of our system in real-life. In this scenario, we deployed our outlier detection algorithms with no knowledge about the data, other than the optional operator feedback. There were no labels, no indication of what percentage of the data was malicious and what the best features would be for detection. Our insights from previous experiments can hint at what features we should use and which of the algorithms are best suited to detect outliers in our test data, but there is no guarantee that they will be valid across all data sets. The knowledge that we relied heavily on, however, was that more predictions would produce more hits and that ensembles reduce the FDR and can yield an acceptable hits versus false alarms ratio.

The aim of this experiment was to make our system generate alerts at some prescribed rate. This rate can be a function of the amount of data being processed, e.g. the operator may wish one record out of 100,000 to be offered for evaluation, or it may be based on elapsed time, such as requiring one alert per hour. To satisfy this, the AR of individual algorithms, which can be tuned, and the effect the ensemble combination of algorithms has on the size of the prediction sets O_k need to be considered. The ensembles of Table 1 show that enough variety can be achieved to ensure that for large n there is likely a $k \in \{1, \ldots, n\}$ where the size of the outlier prediction set versus the size of the processed data could approximate a desired

Table 2: Illustrative AR experiments

Experiment		AR	FDR
A	LDA	0.00031	0.575
	GMM	0.00068	0.24138
	Ensemble	0.00003	0.5
B	LDA	0.00371	0.46822
	GMM	0.00159	0.59113
	Ensemble	0.00108	0.39855
C	LDA	0.00774	0.64162
	GMM	0.00469	0.41304
	Ensemble	0.00425	0.35120
D	LDA	0.02497	0.88892
	GMM	0.04794	0.94244
	Ensemble	0.02277	0.87888

ratio, preferably as $k \to n$. If at $k = n$ we are still producing too many predictions, random sampling could be used to further reduce the number of predictions while preserving the hits versus false alarm ratio of O_n. If at $k \approx n$ the size of O_k is too small, increasing the alert rates of member algorithms can boost the size of O_k to reach the preferred ratio.

Table 2 shows results from one of our alert rate experiments. In these tests, we used two algorithms, LDA and GMM, and their ensemble with $k = n = 2$. For both algorithms, we generated predictions at four comparable ARs (experiments A to D), then combined their results into a simple 2-ensemble. Note that the ARs of the individual algorithms are not exactly the same in any of the experiments, but close: in practice, it was difficult to produce an identical rate for each algorithm. As expected, the table shows reduced ensemble ARs as disagreements between the predictions of member algorithms reduce the candidate outlier sets. We note that the ensemble FDR is not necessarily the best (see Experiment A), but it is always better than the worst individual member FDR (and generally better than most as we found in other experiments). When we cannot verify the performance of individual algorithms and identify which ones are better, an ensemble that produces improved results than the worst algorithm(s) seems a safer option than selecting an algorithm randomly.

A notable observation from Table 2 is that the FDR of member algorithms is not always monotonically decreasing with their AR. For example, the FDR of GMM in Experiment B is higher than in Experiment C despite having a lower AR. This can be explained by the relationship between the FDR, the AR and the sensitivity of the algorithm at the chosen AR. When the AR is such that the number of predictions is close to or lower than the size of the malicious traffic, the FDR is affected by the sensitivity of the algorithm, and this can result in the the observed non-monotonic behaviour. At higher ARs where the number of predictions is much larger than the size of the malicious traffic, the FDR increases as expected due the number of additional false positives included.

4.7 Unlabelled Data Test Results

Our alert rate experiments showed that it was possible to achieve a desired prediction rate using ensembles of unsupervised outlier detection algorithms. Furthermore, on labelled data sets we managed to produce alerts that had a high proportion of malicious content. In our second set of tests, we aimed for similar outcomes on unlabelled data. For these experiments, we needed to manually analyse our outlier predictions to understand our findings.

The unlabelled data in these tests is different from the PREDICT data. It contains traffic passing through organisational boundaries which has been filtered for malicious activity and should contain only normal instances, and therefore lacks some of the variety found in PREDICT. Nevertheless, we used features that proved the most useful in our labelled data tests, and chose various subsets for investigation. We created ensembles of sizes 5 and 6 (see Footnote 1 for the list of algorithms used), as they have proven sufficient to produce acceptable FDRs in labelled tests.

The conclusions we draw from this second set of experiments[2] are promising. By varying the alert rates of individual algorithms and creating multiple outlier prediction sets for our ensembles, we were able to achieve our goal of producing a prescribed alert rate. The decrease between the sizes of the prediction sets O_k as k moved towards the ensemble size n was, however, more marked. For example, in some tests we were not able to produce candidates for our 6-ensemble at $k = 6$, i.e. $|O_6| = 0$ – we could not get all six algorithms to agree on a single outlier candidate, even at high alert rates. This indicates that the algorithms are not as effective at finding outliers in this data using the same features as for PREDICT, or it may also imply that the malicious data in PREDICT may have been too easy to detect.

When analysing the results from tests on TCP/IP connection data, we found several types of outliers that were detected by our algorithms. We characterised some as port reuse, large transfers (emails or downloads), and incomplete sessions. Whilst the records we found were true outliers as defined by our chosen feature sets, they were determined to be non-malicious. Our HTTP tests produced similar sets of outliers. We have identified some issues within these results, such as inappropriate websites among large transfers. We also found that in these tests the types of outliers corresponding to these issues were more frequent than others, a characteristic we also saw in our labelled tests. This meant that we could provide the operator with a higher ratio of actual malicious traffic at our prescribed AR. Whilst outliers of certain types that occur in larger numbers are not guaranteed to be more likely malicious than less frequent ones, finding them early gives the operator the opportunity to filter them out quickly. This increases the likelihood of subsequent results containing more of the previously less frequent and potentially more interesting outliers.

5 Conclusions

In this paper, we described a system to produce alerts on network data using outlier detection techniques. The alerts are offered for expert evaluation to identify residual malicious traffic that may have been missed by other detection tools. Feedback from the operator and ensembles are used to control the rate of alerting and to improve the ratio of relevant outliers in the predictions offered.

We tested on labelled data to select and tune our methods, and were able to satisfy our aims of a desired AR and of producing highly relevant outlier content. Outlier predictions often corresponded to malicious records, an outcome that supported our assumption that malicious data can be captured as outlier traffic. Our tests on unlabelled data were more ambiguous. We satisfied alert rate and dominant outlier ratio requirements, but our predictions contained few obvi-

[2] Of the ca. 200 million records in the data set we random sampled multiple 1.5M and 300K test sets

ously malicious records. This was expected due to the data being cleaned prior by IDSs and attests to their effectiveness. Furthermore, studying outliers can be of benefit even with no malicious data present. Outliers can, for example, indicate configuration issues or the faulty operation of an appliance. We believe that at a suitable AR, the evaluation burden placed on the operator can be justified by the potential benefit offered by finding new threats or previously unknown network issues among the network traffic outliers.

We confirmed during the development of our system that feature selection is vital. They define the outliers, thus in order to capture malicious traffic, features that distinguish malicious data from normal must be included. Finding these features, however, is not trivial. Our system makes it possible to use many different feature sets to produce results prior to ensemble processing. We could therefore cover a large gamut of the possible feature space but we are yet to test the benefit of such a broad approach. We also need to look at the inclusion of higher level aggregate features and the integration of background knowledge, both that of human experts and existing knowledge bases. This could also result in improved results but we need to evaluate how their incorporation may be permitted by the timeliness requirement placed on our system.

References

Aggarwal, C. C. (2013), 'Outlier Ensembles', *SIGKDD Explorations* **14**(2), 49–58.

Amer, M. & Goldstein, M. (2012), Nearest-Neighbor and Clustering based Anomaly Detection Algorithms for RapidMiner, *in* S. Fischer & I. Mierswa, eds, 'Proceedings of the 3rd RapidMiner Community Meeting and Conference', Shaker Verlag GmbH, Budapest, Hungary, pp. 1–12.

Auld, T., Moore, A. & Gull, S. (2007), 'Bayesian Neural Networks for Internet Traffic Classification', *IEEE Transactions on Neural Networks* **18**(1), 223–239.

Blei, D. M., Ng, A. Y. & Jordan, M. I. (2003), 'Latent Dirichlet Allocation', *Journal of Machine Learning Research* **3**, 993–1022.

Bottou, L. (1998), Online Algorithms and Stochastic Approximations, *in* D. Saad, ed., 'Online Learning and Neural Networks', Cambridge University Press, Cambridge, UK, p. 35. revised, Oct 2012.

Frawley, W. J., Piatetsky-Shapiro, G. & Matheus, C. J. (1992), 'Knowledge Discovery in Databases: An Overview', *AI Mag.* **13**(3), 57–70.

Gill, A. & Montague, P. (2016), Online Machine Learning Algorithms for Outlier Detection in Network Traffic Data, *in* 'Proceedings of the Fourteenth Australasian Data Mining Conference', CRPIT, Canberra, Australia.

Guyon, I., Gunn, S., Nikravesh, M. & Zadeh, L. A. (2006), *Feature Extraction: Foundations and Applications*, Springer Science & Business Media.

Hawkins, D. M. (1980), *Identification of Outliers*, Monographs on Statistics and Applied Probability, 1st edn, Chapman and Hall, London; New York.

Holmes, G., Donkin, A. & Witten, I. H. (1994), WEKA: A Machine Learning Workbench, *in* 'Proceedings of the Second Australian and New Zealand Conference on Intelligent Information Systems', IEEE, Brisbane, Australia, pp. 357–361.

Kriegel, H.-P., Kröger, P., Schubert, E. & Zimek, A. (2011), Interpreting and Unifying Outlier Scores, *in* 'Proceedings of Eleventh SIAM International Conference on Data Mining', SIAM / Omnipress, Mesa, Arizona, USA, pp. 13–24.

Kriegel, H.-P., Kröger, P. & Zimek, A. (2010), 'Outlier Detection Techniques, SMD2010 Tutorial Presentation'.

Lunt, T., Jagannathan, R., Lee, R., Whitehurst, A. & Listgarten, S. (1989), Knowledge-based Intrusion Detection, *in* 'Proceedings of the Annual AI Systems in Government Conference', IEEE, Washington, DC, pp. 102–107.

Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M. & Euler, T. (2006), YALE: Rapid Prototyping for Complex Data Mining Tasks, *in* 'Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', ACM, Philadelphia, PA, USA, pp. 935–940.

Murphy, K. P. (2012), *Machine Learning: A Probabilistic Perspective*, Cambridge, MA.

Opitz, D. W. & Maclin, R. (1999), 'Popular Ensemble Methods: An Empirical Study', *Journal of Artificial Intelligence Research* **11**, 169–198.

Paxson, V. (1998), Bro: A System for Detecting Network Intruders in Real-Time, *in* 'Proceedings of the 7th USENIX Security Symposium', USENIX Association, San Antonio, Texas.

Roesch, M. (1999), Snort: Lightweight Intrusion Detection for Networks, *in* D. W. Parter, ed., 'Proceedings of the 13th Conference on Systems Administration (LISA-99)', USENIX, Seattle, WA, USA, pp. 229–238.

Rokach, L. (2010), 'Ensemble-based Classifiers', *Artificial Intelligence Review* **33**(1-2), 1–39.

Scheper, C., Cantor, S. & Karlsen, R. (2009), Trusted Distributed Repository of Internet Usage Data for Use in Cyber Security Research, *in* 'Proceedings of the Cybersecurity Applications & Technology Conference For Homeland Security', IEEE, pp. 83–88.

Tzur-David, S. (2011), Network Intrusion Prevention Systems: Signature-Based and Anomaly Detection, Ph.D., The Hebrew University of Jerusalem, Jerusalem, Israel.

Veeramachaneni, K., Arnaldo, I., Korrapati, V., Bassias, C. & Li, K. (2016), AIˆ2: Training a Big Data Machine to Defend, *in* '2016 IEEE 2nd International Conference on Big Data Security on Cloud', IEEE, New York, NY, USA, pp. 49–54.

Vega-Pons, S. & Ruiz-Shulcloper, J. (2011), 'A Survey of Clustering Ensemble Algorithms', *International Journal of Pattern Recognition and Artificial Intelligence* **25**(03), 337–372.

Williams, N., Zander, S. & Armitage, G. (2006), 'A Preliminary Performance Comparison of Five Machine Learning Algorithms for Practical IP Traffic Flow Classification', *SIGCOMM Comput. Commun. Rev.* **36**(5), 5–16.

Zimek, A., Campello, R. J. & Sander, J. (2014), 'Ensembles for Unsupervised Outlier Detection: Challenges and Research Questions a Position Paper', *SIGKDD Explorations* **15**(1), 11–22.

Industry Showcases

Application of Allocating Pattern Mining in Commercial Banking: A Novel Approach for Customer-Product Cross-Selling

Xuan Yang[1], Pengpeng Hao[1], Qian Gao[2], Jun Zhang[1], and Yanbo J. Wang[1]

[1] China Minsheng Banking Corp., Ltd.
No. 2, Fuxingmennei Avenue, Xicheng District, Beijing 100031, China
{yangxuan2, haopengpeng, zhangjun37, wangyanbo}@cmbc.com.cn

[2] Institute of Finance and Banking, Chinese Academy of Social Sciences
No. 5, Jianguomennei Dajie, Dongcheng District, Beijing 100732, China
gaoqian2002@yahoo.com.cn

Abstract

Nowadays, with the liberalisation of interest rate and development of "FinTech", Chinese commercial banks are suffering from serious customer churns, particularly, high-valued customers in private banking. According to empirical experiences of the Chinese banking industry, customers who have more than eight financial products have a lower probability of churning. As a result, cross-selling has been a key issue for Chinese commercial banks to retain their customers. In this paper, we mainly focus on the high-valued customers of Chinese private banks. With the real data of the latest years, the **AL**locating **P**atterns **M**ining (ALPM) technique was applied to analyse the wealth allocation rules of high-valued customers, and cross-selling suggestions were proposed in order to offering them appropriate financial products, finally improving high-valued customers' loyalty of the private bank.

ALPM, which derived from Association Rule Mining (ARM) technique, was firstly introduced by Wang *et al.* in 2008. An Association Rule (AR) is a common knowledge model in data mining that describes an implicative co-occurring relationship between two disjoint sets of binary-valued transaction database attributes (items), expressed in the form of an "antecedent $\Rightarrow$ consequent" rule. A variant of the AR is the Weighted Association Rule (WAR). An ALlocating Pattern (ALP) is a special form of WAR, where each rule item is associated with a weighting score between 0 and 1, and the sum of all rule item scores is 1. Such rules can not only indicate the implicative co-occurring relationships between two (disjoint) sets of items in a weighted setting, but also inform the "allocating" relationship among rule items. ALPs can be demonstrated to be applicable in marketing and possibly a surprising variety of other areas/situations.

In this case, we collected real data of a Chinese private bank. Based upon our early research results, an important segment of high-valued customers was defined as Financial Products Oriented Customers, whose features could be summarised as follows: (1) large proportion of financial products while small proportion of deposit; (2) purchased short-term financial products; (3) each single purchased financial product which amount is usually less than 3 million RMB. Since customers of this group have a huge demand for purchasing financial products and conducting wealth management, this ALPM technique was applied to them for following purposes: (i) exploring comparatively stable wealth allocating rules of high-valued customers; and (ii) offering accurate cross-selling suggestions based upon rules explored to improve the number of products that high-valued customers bounded. As for the products discussed in this paper, we mainly classified them into three categories: deposit, financial products and others (i.e., fund, treasury bonds and insurance). Since deposit could not only save risk capital but also make profit through loan-deposit spreads for commercial banks, it was considered an much easier way to improve operating performance compared with financial products and others. As a result, our purpose (ii) could be further illustrated as that (iii) we would like to improve the proportion of deposit of certain high-valued customers without decreasing their total assets managed by the private bank.

After clarifying our purposes and scope of research, our study was conducted by following steps:

Step 1: **Data Pre-processing.** Data of customers was transformed into one-sum weighted forms in order to fit typical format required in ALPM for further treatment. Then we transferred products' names into numbers, "1" for deposit, "2" for financial products, and "3" for others.

Step 2: **ALPM Application.** With "Support = 0.2%, Confidence = 25%", ALPM was applied to the data which has been treated. All evaluations were obtained using a JAVA application program. The experiments were run on a 3.00 GHz Pentium (R) Dual-Core CPU with 1.96 GB of RAM running under the *x*86 Windows Operating System.

Step 3: **ALP Explanation.** After successive experiments, the key rules revealed by the data were listed below:

	Antecedent	Consequent	Confidence
Rule1	**2(0.77) 3(0.14)**	**1(0.09)**	**0.25**
Rule2	1(0.04) 3(0.01)	2(0.95)	0.29
Rule3	1(0.04) 3(0.02)	2(0.94)	0.29
Rule4	1(0.06) 3(0.04)	2(0.90)	0.47
Rule5	1(0.06) 3(0.01)	2(0.93)	0.63
Rule6	1(0.04) 3(0.14)	2(0.82)	0.40
Rule7	1(0.03) 3(0.08)	2(0.89)	0.50
Rule8	1(0.07) 3(0.02)	2(0.91)	0.30

According to our purpose (iii), we only focus on **Rule1**, which means if a customer allocate about 77% of his/her money to financial products and 14% to others, then he/she has 25% probability to invest about 9% to deposit.

Step 4: Customer-Product Cross-Selling. 8 customers of 106 was selected for this rule, and 5 have responded. By offering them accurate cross-selling suggestion, 5 customers have improved their deposit significantly, making deposit amount increasing by 9.4 million RMB in 6 months.

***Keywords*:** Allocating Pattern Mining, Association Rule Mining, Commercial Banking, Cross-Selling, Private Bank.

Application of Graph Mining in Corporate Banking: A Novel Approach for Customer Acquisition and Development

Yanbo J. Wang and Yonghong Yang

China Minsheng Banking Corp., Ltd.
No. 2, Fuxingmennei Avenue, Xicheng District, Beijing 100031, China
{wangyanbo,yangyonghong}@cmbc.com.cn

Abstract

China's economy has entered a new normal status, and the interest rate liberalisation is becoming accelerated. Under such a background, the competition in domestic banking industry is becoming fierce. In particular, commercial banks should put an emphasis on the acquisition and development of new customers. However, for the purpose of attracting new customers in corporate banking, domestic commercial banks still mainly rely on customer managers' "point-to-point" strategy, but have not developed a "batch" mode yet. Particularly, they have failed to show a resultant force brought by the trade or industry. Obviously, the single-customer ("point-to-point") development mode generally displays higher cost but lower efficiency when compared to the "batch" development mode.

In this paper, we propose a new "batch" method, based on graph mining, for the acquisition and development of new customers in corporate banking. In fact, with long-term data accumulation, the banks have collected a large amount of customer data, especially such data that reflects the transaction and transfer behaviours between the customers. Using a graphical technique to construct a customer transaction network (i.e., a kind of social network) could help to characterise the transaction and transfer behaviours of the customers, and enable the banks to explore upstream and/or downstream enterprises in the industrial chain. Such enterprises in the chain who do not have an account in this bank (e.g., the bankrolls are transferred from this bank to other banks, or transferred from other banks to the current bank) are identified, in a "batch" manner. In addition, the graph model can also be applied to quantitatively analyse the influence of enterprise customers on the network so that the "core nodes" in the network could be further identified.

Taking a domestic commercial bank as a study, we extracted relevant information — the information about over 500 thousand total customers in corporate banking, from the bank's EDW (Enterprise Data Warehouse). Each customer was considered as a centre, and a directed graph on an annual basis was constructed to depict a social network between this customer and other customers from either this or other banks for "Force-directed Layout" visualisation purpose, in which each node represents a customer, each directed edge represents the transfer behaviour between two customers, and the thickness of an edge reflects the cumulative transfer amount within a year. In order to effectively identify the "core" enterprise in the industrial chain, the conventional approach of checking the counterparties at one level (i.e., directed transaction customers) is insufficient. We construct a multi-level (five-level) transaction chain graph.

From the transaction chain graph, we can compute the transaction chain level, the number of involved enterprises, the transaction amount, the number of enterprises opening accounts in this bank, the number of enterprises opening accounts in other banks, the region distribution of counterparties, and the transaction in each region. This information provides the bank with a business opportunity of attracting potential customers from the upstream and/or downstream counterparties. Of course, not every customer from other banks is our target. Therefore, we propose a graph mining model to first compute and then sort the new customer referral indices of potential customers. By doing so, the bank's first-line market staffs could have a better idea regarding the determination of their customer priority list. The formula of the "new customer referral index" $Recommend\ (i)$ is given as follows:

$$\boldsymbol{R}(i) = \left(\frac{D_1(i) - \min(D_1)}{\max(D_1) - \min(D_1)} + \frac{D_2(i) - \min(D_2)}{\max(D_2) - \min(D_2)}\right) * 0.5 * 100$$

$$R(i) \in [0,100]$$

Where

- $D_1\ (i) = \ln\ (degree\ (i) + 1)$ represents the directed influence of enterprise i, and $degree\ (i)$ is the degree of enterprise i. The larger the degree of an enterprise is, the greater the directed influence of the enterprise is in the network.
- $D_2\ (i) = \ln\ (1 + \Sigma_j\ (AMT\ (i, j)\ *\ degree\ (j)))$ represents the influence of counterparties of enterprise i (i.e., the indirect influence of enterprise i); $AMT\ (i, j)$ is the annual transaction amount between enterprise i and its counterparty j, and $degree\ (j)$ is the degree of counterparty j of enterprise i. If the degree of a counterparty of enterprise i is larger, and the corresponding transaction amount is larger, then the indirect influence of enterprise i is greater as well.

The new customer acquisition and development method proposed in this paper, is deployed through the bank's CRM System. The system could effectively submit the recommended new customer list to relevant customer managers. Since the running of relevant functional modules, the system has produced huge gains in the practical application. Specifically, the system had successfully recommended 12,112 new customers within the first 8 months of being used, with a total of 40.3 billion RMB deposit.

Keywords: Corporate Banking, Customer Acquisition, Force-directed Layout, Graph Mining, Social Network.

Ribbon matching: an experimental approach to linking unstructured data

Garry A. Mitchell
PO Box 8, Jindera, NSW, 2642
garry.mitchell@bigpond.com

Abstract

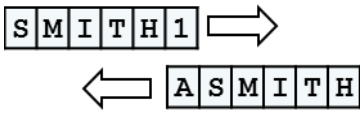
The world is awash with data. Considerable opportunities exist to coalesce external data with existing organizational data to enhance analysis and decision making. The challenge in this coalescence is linking external data to the internal data. Traditionally, linkage is determined by seeking matches in like field types that exhibit some ability to discriminate entities. To enable this comparison between fields, considerable effort is put into identifying field types and then cleansing and formatting the contained data. With growth in sourcing unstructured external data, such as web pages downloaded using web-scraping techniques, the difficulties in identifying, cleansing and matching data increase.

In an attempt to overcome these difficulties a new experimental ribbon matching technique has been developed and is now being tested and refined.

The basic principle of the approach is to create two strings and compare them, character by character, as they conceptually slide past each other. There are some similarities to the sliding windows string match approach except there is no need to know a priori what is being looked for – any matching cluster of characters contained anywhere in the strings will be identified.

· The first string is created by extracting appropriate external data, stripping it of all punctuation marks, symbols, capitalisation and spaces and then concatenating it into one continuous string. The second string is created from internal data holdings using all fields that have a degree of uniqueness for matching purposes. This data is also stripped of all punctuation marks, symbols, capitalisation and spaces and then concatenated into one continuous string. Conceptually these two strings are then slid against each other comparing overlapping characters at each step. This is simply illustrated in Figure 1. Adjoining matched characters are counted as match clusters of differing sizes.

A count can be maintained of the frequency, size and content of cluster matches for all matching clusters identified at each iteration of the comparison. This information can be used to indicate the likelihood that the identity related to each string matches. The external string can be run against many strings from internal data to



identify any matches.

Figure 1: Illustration of sliding strings

To reduce the iterations in this comparison process each string undergoes a simple two step rearrangement which creates two long strings which are then directly compared. These are referred to as ribbons.

A simple experimental prototype was built using VBA & Excel which was used to find matches in limited sets of data thereby proving the concept. Further experimentation was undertaken using Python and now the applications runs in Spark on a Hadoop cluster.

Two distinct components have been developed and are now being refined. Firstly there is the actual ribbon match engine which undertakes the comparison of strings and returns frequency, size and content of matching clusters. The second component is the scoring engine which takes the output from the ribbon match engine and calculates scores which indicate the likelihood that there is a match between ribbons.

The current version runs match searches against a table of approximately 39 million strings. Results to date are very encouraging with correct matches being present in the top ten scores in greater than 99% of cases run. Several opportunities to increase the ribbon engine speed and scoring engine accuracy have been identified and will form the basis of further experimentation.

The primary weakness of ribbon matching is the high processing overhead required when running large sets of input strings. The primary strength lies in the ease with which it deals with unstructured data input and therefore less need for cleansing and preparation of input data.

Aside from providing an alternative approach to conventional record linkage approaches an unexpected opportunity has arisen from this work. Because the ribbon matching process identifies clusters of similarity between input strings it is able to identify relationships, both formal and informal, that exist between the entities represented by these strings. This information can be used to map relationship networks that exist within the input populations.

Discovering temporal operational modes in an industrial chemical process identifying their driving factors with symbolic data mining

David Stirling[1], Paul Zulli[1] and Sheng Chew[2]

[1] Faculty of Engineering and Information Sciences
University of Wollongong
Email: stirling@uow.edu.au , paulz@uow.edu.au

[2] BlueScope Steel Limited
Email: Sheng.Chew@bluescopesteel.com

Abstract

Maximising the life of costly assets, such as a large industrial reactor, i.e. an ironmaking blast furnace (BF), is a significant issue to the company concerned. Extending, or rather not contributing to any reduction of the typical campaign life of a BF is commonly achieved through careful selection and control its feed inputs, the raw materials, their physical and chemical quality and characteristics, as well as appropriate operator interventions such as burden distribution changes, adjustments to the practices for energy input, and casting or tapping the resulting liquid metal and slag.

One of a number of key metrics that is utilised by operational staff in determining the health, or otherwise, of the BF process, as well as to monitor and influence its behaviour, is the Stave Heat Load (SHL). Staves are generally blocks of cast iron, or other material, that line and insulate the shell surface of a BF, and through which a series of embedded water-cooling pipes are distributed. The heat absorbed by the circulating water passing through various collections of staves is recorded at external heat exchangers. Generally, controlling the SHL is critically important because of the long- and short-term consequences to the BF process. For the long-term, the BF life (campaign) may be prolonged if both the magnitude and the variability of SHL are controlled. For the short-term, fuel consumption and product quality may be improved by maintaining SHL as low as possible.

On a number of occasions during a previous campaign of over ten years, the BlueScope No. 5 Blast Furnace (BF5) at Port Kembla N.S.W., experienced several sustained periods of abnormally high levels of SHL, these being beyond the aim total heat load (1500 GJ/day). The underlying influences and driving factors behind these variations, as well as the appropriate remedial strategies necessary to mitigate such occurrences were not well understood. Historically, the levels of SHL and its variability are often very difficult to diagnose and troubleshoot because a vast range of operational parameters can contribute to influence the dynamic nature of the BF process, i.e. there may be several likely individual causes, or combinations of possible factors in differing contexts or periods of operation, that may be the major causal driver.

Typically, all process controls and adjustments are carefully regulated, monitored and managed by operational personnel. Over many years, a significant body of knowledge has been amassed regarding certain well-understood modes of the process behaviours. The driving influences in such cases can be typically adjusted so as to mitigate any observed and undesirable trends, or to promote favourable ones. However, on several occasions an apparently familiar modal trend in the SHL may not respond to previously successful remedial strategies. In such cases, it is generally assumed that certain inadvertent and unforeseen changes in the contextual processes may have lead to new behaviours (or Concept Drift), being manifested as higher, lower or more variant SHL trends.

A series of machine learning and data mining approaches were utilised to provide an objective means of identifying a more holistic structure of modal patterns for this large, complex, multi-factorial and highly coupled process. Apart from the key output metric of the SHL, there were five main groups of attributes associated with the BF, each reflecting a range of dynamic or temporal properties such as: the quantity and quality of the consumed feed materials and their distribution within the BF, the fuel and energy rate utilised, the overall stability of the process, and the rate and quantity of extracted molten metal. All of these in turn entail some 150 further individual variables that were distributed amongst several associated data repositories. Here, the intended approach was to mine the more expansive BF data to ultimately identify the driving factors associated with significant process modalities of particular interest.

A range of techniques were employed, such as thematic and glyph visualisations, transformations, multi-dimensional scaling of the data, as well as symbolic modelling and analysis, in order to generate final sets of rules of the driving features behind each specific SHL modality.

Through a number of data mining (machine learning) trials on the BF campaign data, a number of key influencing factors were identified as being significant in maintaining an aim total SHL of 1500 GJ/-day. Several influential factors, previously overlooked, were also identified, including certain combinations of raw materials (e.g. percent pellet in the ore base) and various types of burden distribution (coke placement near the wall), all of which demonstrated a significantly high accuracy in repeatedly identifying the nine super-states of SHL, ranging from low to very high. These outcomes and new understandings have subsequently been added to the ongoing process improvement knowledge of the company.

Keywords: *Gaussian Mixture Models, Modalities, Link analysis, Data Mining, Stave Head Loads.*

Using Analytics to Improve Government Services

Rohan Baxter
Data Science and Special Purpose Acquisition
Smarter Data
Australian Taxation Office
PO Box 9977, Civic Square, 2608 ACT
rohan.baxter@ato.gov.au

Abstract

This talk describes how analytics methods and models were used to contribute to an improvement in the client and staff experience for the processing of individual income tax returns. The improvements could be measured in terms of improved processing times, fewer inbound client calls asking about progress of returns and fewer complaints. Key contributions to the improved services included the use of an integrated view of the client experience allowing the targeting of key irritants. The key irritants were then addressed using automation via analytics models, and targeted push messages to clients.

Keywords: Integrated Views, Automation, Predictive Modelling

Data Discovery at the ATO

Warwick Graco, Tony Nolan, Stewart Turner and Hari Koesmarno
Data Science and Special Purpose Acquisition
Smarter Data Program
Email: Warwick.graco@ato.gov.au

Abstract: This showcase covers the work done by the Data-Discovery Team at the Australian Taxation Office (ATO). This includes projects currently being carried out and proposed ones in the future.

The current focus of the Data-Discovery Team is on assisting Risk and Intelligence staff and Client-Experience owners to understand the composition of various populations. Examples include individuals, small-business and not-for-profit markets. The aim is to find the natural clusters in each target population.

The team is concentrating on the development of a four-stage clustering and knowledge-capture algorithm and that will be supplied as an application (App) written in R. It will have a Shiny frontend. The stages of the App include (1) identifying the discriminatory as distinct from the masking features used to find clusters (2) using the discriminatory features to cluster the target population (3) applying induction learning to work out the membership rules of each cluster and (4) capturing the insights that staff provide from examining the cases found in the clusters. These insights will be converted to business rules and applied to the target population to identify similar cases.

Users will use point and click operations with a mouse to operate the App. The App will help staff who analyse data for risk and intelligence and for client-experience purposes. The Data-Discovery Team will provide technical and training support to staff that use the App and will continue to develop new capabilities. This will enable users to perform the discovery function and help them to gain insights from the data. It should also provide increased job satisfaction from performing this activity.

Other tools developed and applied by the Data-Discovery Team include the use of capstone analysis and cohort analysis. The former rolls up both quantitative and qualitative measures using a compression algorithm to provide a two-dimensional grid showing where members of a target population lie compared to their peers. This assists to identify what is commonly called 'needles in a haystack' or cases that are well camouflaged and difficult to detect. The latter provides fine-grained population segments that consist of cases that share the same defining attributes such as customers who live in the same geographic area, are in the same income range and purchase the same retail products.

Future challenges for the Data-Discovery Team include developing a (1) column clustering algorithm to find patterns buried in the noise in data and to manage columns that have sparse data entries and (2) a digital-fingerprinting solution to identify each citizen's unique digital signature and what it is coupled to when it comes to different behaviours. Attention will also be given to using smart data to discover patterns in data and to produce classification and prediction models. This data is enriched with meta-knowledge and each data point is like a radio beacon that sends signals about what it contains in the way of useful and relevant information.

Author Index

Recent Volumes in the CRPIT Series

ISSN 1445-1336

Listed below are some of the latest volumes published in the ACS Series *Conferences in Research and Practice in Information Technology*. The full text of most papers (in either PDF or Postscript format) is available at the series website `http://crpit.com`.

Volume 147 - Computer Science 2014
Edited by Bruce Thomas, University of South Australia and Dave Parry, AUT University, New Zealand. January 2014. 978-1-921770-30-2.
Contains the proceedings of the Australian System Safety Thirty-Seventh Australasian Computer Science Conference (ACSC 2014), Auckland, New Zealand, 20 – 23 January 2014.

Volume 148 - Computing Education 2014
Edited by Jacqueline Whalley, AUT University, New Zealand and Daryl D'Souza, RMIT University, Australia. January 2014. 978-1-921770-31-9.
Contains the proceedings of the Sixteenth Australasian Computing Education Conference (ACE2014), Auckland, New Zealand, 20 – 23 January 2014.

Volume 149 - Information Security 2014
Edited by Udaya Parampalli, University of Melbourne, Australia and Ian Welch, Victoria University of Wellington, New Zealand. January 2014. 978-1-921770-32-6.
Contains the proceedings of the Twelfth Australasian Information Security Conference (AISC 2014), Auckland, New Zealand, 20 – 23 January 2014.

Volume 150 - User Interfaces 2014
Edited by Burkhard C. Wünsche, University of Auckland, New Zealand and Stefan Marks, AUT University, New Zealand. January 2014. 978-1-921770-33-3.
Contains the proceedings of the Fifteenth Australasian User Interface Conference (AUIC 2014), Auckland, New Zealand, 20 – 23 January 2014.

Volume 151 - Australian System Safety Conference 2013
Edited by Tony Cant, Defence Science and Technology Organisation, Australia. May 2013. 978-1-921770-38-8.
Contains the proceedings of the Australian System Safety Conference (ASSC 2013), Adelaide, Australia, 22 – 24 May 2013.

Volume 152 - Parallel and Distributed Computing 2014
Edited by Bahman Javadi, University of Western Sydney, Australia and Saurabh Kumar Garg, IBM Research, Australia. January 2014. 978-1-921770-34-0.
Contains the proceedings of the Twelfth Australasian Symposium on Parallel and Distributed Computing (AusPDC 2014), Auckland, New Zealand, 20 – 23 January 2014.

Volume 153 - Health Informatics and Knowledge Management 2014
Edited by Jim Warren, University of Auckland, New Zealand and Kathleen Gray, University of Melbourne, Australia. January 2014. 978-1-921770-35-7.
Contains the proceedings of the Seventh Australasian Workshop on Health Informatics and Knowledge Management (HIKM 2014), Auckland, New Zealand, 20 – 23 January 2014.

Volume 154 - Conceptual Modelling 2014
Edited by Georg Grossmann, University of South Australia and Motoshi Saeki, Tokyo Institute of Technology, Japan. January 2014. 978-1-921770-36-4.
Contains the proceedings of the Tenth Asia-Pacific Conference on Conceptual Modelling (APCCM 2014), Auckland, New Zealand, 20 – 23 January 2014.

Volume 155 - The Web 2014
Edited by Stephen Cranefield, University of Otago, New Zealand, Andrew Trotman, University of Otago, New Zealand and Jian Yang, Macquarie University, Australia. January 2014. 978-1-921770-37-1.
Contains the proceedings of the Second Australasian Web Conference (AWC 2014), Auckland, New Zealand, 20 – 23 January 2014.

Volume 156 - Australian System Safety Conference 2014
Edited by Tony Cant, Defence Science and Technology Organisation, Australia. May 2014. 978-1-921770-39-5.
Contains the proceedings of the Australian System Safety Conference (ASSC 2014), Melbourne, Australia, 28 – 30 May 2014.

Volume 158 - Data Mining and Analytics 2014
Edited by Xue Li, University of Queensland, Lin Liu, University of South Australia, Kok-Leong Ong, Deakin University and Yanchang Zhao, Department of Immigration and Border Protection, Australia and RDataMining.com, Australia. November 2014. 978-1-921770-17-3.
Contains the proceedings of the Twelfth Australasian Data Mining Conference (AusDM'14), Brisbane, Australia, 27–28 November 2014.

Volume 159 - Computer Science 2015
Edited by David Parry, AUT University, New Zealand. January 2015. 978-1-921770-41-8.
Contains the proceedings of the 38th Australasian Computer Science Conference (ACSC 2015), Sydney, Australia, 27 – 30 January 2015.

Volume 160 - Computing Education 2015
Edited by Daryl D'Souza, RMIT University and Katrina Falkner, University of Adelaide, Australia. January 2015. 978-1-921770-42-5.
Contains the proceedings of the 17th Australasian Computing Education Conference (ACE 2015), Sydney, Australia, 27 – 30 January 2015.

Volume 161 - Information Security 2015
Edited by Ian Welch, Victoria University of Wellington, New Zealand and Xun Yi, RMIT University, Australia. January 2015. 978-1-921770-43-2.
Contains the proceedings of the 13th Australasian Information Security Conference (AISC 2015), Sydney, Australia, 27 – 30 January 2015.

Volume 162 - User Interfaces 2015
Edited by Stefan Marks, AUT University and Rachel Blagojevic, Massey University, New Zealand. January 2015. 978-1-921770-44-9.
Contains the proceedings of the 16th Australasian User Interface Conference (AUIC 2015), Sydney, Australia, 27 – 30 January 2015.

Volume 163 - Parallel and Distributed Computing 2015
Edited by Bahman Javadi, University of Western Sydney and Saurabh Kumar Garg, University of Tasmania, Australia. January 2015. 978-1-921770-45-6.
Contains the proceedings of the 13th Australasian Symposium on Parallel and Distributed Computing (AusPDC 2015), Sydney, Australia, 27 – 30 January 2015.

Volume 164 - Health Informatics and Knowledge Management 2015
Edited by Anthony Maeder, University of Western Sydney, Australia and Jim Warren, University of Auckland, New Zealand. January 2015. 978-1-921770-46-3.
Contains the proceedings of the 8th Australasian Workshop on Health Informatics and Knowledge Management (HIKM 2015), Sydney, Australia, 27 – 30 January 2015.

Volume 165 - Conceptual Modelling 2015
Edited by Motoshi Saeki, Tokyo Institute of Technology, Japan and Henning Kö, Massey University, New Zealand. January 2015. 978-1-921770-47-0.
Contains the proceedings of the 11th Asia-Pacific Conference on Conceptual Modelling (APCCM 2015), Sydney, Australia, 27 – 30 January 2015.

Volume 166 - Australasian Web Conference 2015
Edited by Joseph G. Davis, University of Sydney, Australia and Alessandro Bozzon, Delft University of Technology, The Netherlands. January 2015. 978-1-921770-48-7.
Contains the proceedings of the 3rd Australasian Web Conference (AWC 2015), Sydney, Australia, 27 – 30 January 2015.

Volume 167 - Interactive Entertainment 2015
Edited by Yusuf Pisan, University of Technology, Sydney, Keith Nesbitt and Karen Blackmore, University of Newcastle, Australia. January 2015. 978-1-921770-49-4.
Contains the proceedings of the 11th Australasian Conference on Interactive Entertainment (IE 2015), Sydney, Australia, 27 – 30 January 2015.