Conferences in Research and Practice in
Information Technology

Volume 158

Data Mining and Analytics 2014 (AusDM'14)

,

Data Mining and Analytics 2014

Proceedings of the
Twelfth Australasian Data Mining Conference
(AusDM'14), Brisbane, Australia,
27–28 November 2014

Xue Li, Lin Liu, Kok-Leong Ong and
Yanchang Zhao, Eds.

Volume 158 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 2014. Proceedings of the Twelfth Australasian Data Mining Conference (AusDM'14), Brisbane, Australia, 27–28 November 2014

Conferences in Research and Practice in Information Technology, Volume 158.

Editors:

Xue Li
School of Information Technology and Electrical Engineering
Faculty of Engineering, Architecture and Information Technology
The University of Queensland
Brisbane, QLD 4072, Australia
Email: x.li@uq.edu.au

Lin Liu
School of Information Technology and Mathematical Sciences
Division of Information Technology, Engineering and the Environment
University of South Australia
Mawson Lakes Campus
Mawson Lakes, SA 5095, Australia
Email: lin.liu@unisa.edu.au

Kok-Leong Ong
School of Information Technology
Deakin University
Burwood, Victoria 3125, Australia
Email: kok-leong.ong@deakin.edu.au

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

Series Editors:
Vladimir Estivill-Castro, Griffith University, Queensland
Simeon J. Simoff, University of Western Sydney, 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 158.
ISSN 1445-1336.
ISBN 978-1-921770-17-3.

Document engineering by CRPIT, November 2014.

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

Table of Contents

Message from the General Chairs

Dear AusDM authors, participants, and sponsors,

We like to welcome you to the Twelfth Australasian Data Mining conference (AusDM'14). We are proud to host AusDM in Brisbane for the first time.

We are excited to have two internationally recognised keynote speakers. Both keynote speaker show extreme prolific career in data mining. Prof Peter Bartlett from the University of California, Berkeley will talk about efficient algorithms to solve decision making problems under uncertainty that are quite common in data mining. Prof Geoff McLachlan from the University of Queensland will attempt to bridge the gap between data mining and statistics, and will talk about statistical procedures that can be adopted to extract meaningful patterns from the data.

This year's conference had paper submissions in two categories, research and application, and we are pleased to have received a good number of high quality submissions which show the breadth and depth of data mining and analytics that happens in Australasian academics, and private, public and government organisations. Of the 53 submitted papers, 24 will be presented. Of these 18 are research track papers and 6 are industry papers.

To enrich the practical aspects of our conference program, we are pleased to offer a practical tutorial on the use of R in Data Mining to be given by Dr Yanchang Zhao, Senior Data Miner at the Department of Immigration and Border Protection Australia, and author of the Elsevier book "R and Data Mining: Example and Case Studies".

There are many people and organisations who have supported this year's conference. We thank all authors who have submitted papers, and we like to congratulate those who were successful. We also like to acknowledge all reviewers who have put significant efforts into careful assessment of the submitted papers. We like to thank the AusDM local chair and volunteers who supported the running of the conference.

Finally, we like to thank our sponsors for their support: Queensland University of Technology (Platinum Sponsor), the Australian National University and the Deakin University (Gold Sponsors).
We hope you enjoy AusDM'14 and your stay in Brisbane.

Yours Sincerely,

Richi Nayak
Queensland University of Technology, Brisbane

Paul Kennedy
University of Technology, Sydney

November 2014

Message from the Program Chairs

Welcome to the 12th Australasian Data Mining Conference (AusDM14), in Brisbane, Australia.

A total of 53 papers were submitted to the two conference tracks (research and application). After careful consideration 24 papers were selected for publication and presentation. AusDM follows a rigid blind peer-review and ranking-based paper selection process. Each paper was rigorously reviewed by at least three reviewers and up to five reviewers took part in providing the assessment of a papers merits.

We would like to thank all the authors who submitted their work to AusDM14. Without your contributions and support, it would not be possible for the conference to succeed. We will continue to extend the conference format to be able to accommodate more presentations.

We thank all Program Committee members for their timely and high-quality reviews. Our Program Committee members have been pivotal to the success of this conference. On behalf of the entire organising committee, we express our appreciation to the committee for their cooperative spirit and extraordinary effort. It has been a true privilege to work with such a dedicated and focused team, many whom were also active in helping with the publicity of the conference. We also wish to extend our appreciation to any of the external reviewers relied upon by the Program Committee members. They have played a part of making this conference possible.

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

Yours Sincerely,

Xue Li
The University of Queensland, Brisbane

Lin Liu
University of South Australia, Adelaide

Kok-Leong Ong
Deakin University, Melbourne

Yanchang Zhao
Department of Immigration and Border Protection, Australia;
and RDataMining.com

November 2014

Conference Organisation

General Chairs

Richi Nayak, Queensland University of Technology
Paul Kennedy, University of Technology Sydney

Local Chair

Yue Xu, Queensland University of Technology

Program Chairs (Research)

Lin Liu, University of South Australia
Xue Li, University of Queensland

Program Chairs (Application)

Yanchang Zhao, Department of Immigration and Border Protection, Australia; and RDataMining.com
Kok-Leong Ong, Deakin University, Melbourne

Sponsorship Chair

Andrew Stranieri, University of Ballarat

Steering Committee Chairs

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

Other Steering Committee Members

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

Program Committee

Research Track

Shafiq Alam, University of Auckland, New Zealand
Adil Bagirov, Federation University Australia
Yee Ling Boo, Deakin University, Australia
Ling Chen, University of Technology Sydney, Australia
Xuan-Hong Dang, University of California at Santa Barbara, USA
Ping Guo, Beijing Normal University, China
Md Zahidul Islam, Charles Sturt University, Australia
Yun Sing Koh, University of Auckland, New Zealand
Siddhivinayak Kulkarni, Federation University Australia
Paul Kwan, University of New England, Australia
Robert Layton, Federation University Australia
Gang Li, Deakin University, Australia
Geng Li, ORACLE, USA
Huizhi Liang, Australian National University
Bing Liu, Children's Cancer Institute Australia for Medical Research, UNSW, Australia
Brad Malin, Vanderbilt University, USA
Md Marwan Md Fuad, University of Tromsø, Norway
Namita Mittal, MNIT Jaipur, India
Christine O'Keefe, CSIRO, Australia
Tom Osborn, University of Technology Sydney, Australia
Francois Poulet, University of Rennes 1 - IRISA, France
Md Geaur Rahman, Charles Sturt University/Bangladesh Agricultural University
Md Anisur Rahman, Charles Sturt University, Australia
Andrew Stranieri, Federation University, Australia
Siamak Tafavogh, University of Technology Sydney, Australia
Xiaohui Tao, University of Southern Queensland, Australia
Dinusha Vatsalan, Australian National University
Sitalakshmi Venkatraman, Federation University, Australia
Guandong Xu, University of Technology Sydney, Australia
John Yearwood, Federation University, Australia
Ting Yu, Transport for NSW, Australia
Mengjie Zhang, Victoria University of Wellington, New Zealand
Qiang Zhu, LinkedIn

Application Track

Chris Barnes, Australian Institute of Sport, Australia
Rohan Baxter, Australian Taxation Office, Australia
Shane Butler, Telstra, , Australia
Adriel Cheng, Defence Science and Technology Organization, Australia
Ross Farrelly, Datamilk
Vittorio Furlan, Accenture
Richard Gao, Department of Agriculture, Fisheries and Forestry, Australia
Warwick Graco, Australian Taxation Office, Australia
Lifang Gu, Australia Taxation Office, Australia
Yingsong Hu, Department of Human Services, Australia
Edward Kang, Emagine International
Luke Lake, Department of Immigration and Border Protection, Australia
Jin Li, Geoscience Australia
Kee Siong Ng, Pivotal
Wei Peng, Global Analytics Hub, Lenovo
Clifton Phua, SAS Institute
Martin Rennhackkamp, PBT Group
Rory Winston, Commerzbank AG
Andrew Wyer, Department of Immigration and Border Protection, Australia
Debbie Zhang, Australian Taxation Office, Australia
Ke Zhang, Department of Health and Ageing, Australia
Sam Zhao, ABARES, Australian Government

AusDM Sponsors

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

http://www.togaware.com

http://www.anu.edu.au

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

Queensland University of Technology
Brisbane Australia
http://www.qut.edu.au/

Keynotes

Keynote - I
Learning in Sequential Decision Problems

Prof. Peter Bartlett,
University of California, Berkeley, USA &
Queensland University of Technology, Brisbane

Abstract
Many problems of decision making under uncertainty can be formulated as sequential decision problems in which a strategy's current state and choice of action determine its loss and next state, and the aim is to choose actions so as to minimize the sum of losses incurred. For instance, in internet news recommendation and in digital marketing, the optimization of interactions with users to maximize long- term utility needs to exploit the dynamics of users. We consider three problems of this kind: Markov decision processes with adversarially chosen transition and loss structures; policy optimization for large scale Markov decision processes; and linear tracking problems with adversarially chosen quadratic loss functions. We present algorithms and optimal excess loss bounds for these three problems. We show situations where these algorithms are computationally efficient, and others where hardness results suggest that no algorithm is computationally efficient.

Biography
Peter Bartlett is a professor in Mathematics at the Queensland University of Technology and professor in Computer Science and Statistics at UC Berkeley. His research interests include machine learning, statistical learning theory, and adaptive control. He has served as associate editor of Bernoulli, Machine Learning, the Journal of Machine Learning Research, the Journal of Artificial Intelligence Research, the IEEE Transactions on Information Theory, and Mathematics of Control Signals and Systems, and on the editorial boards of Machine Learning, JAIR, and Foundations and Trends in Machine Learning. He has been professor in the Research School of Information Sciences and Engineering at the Australian National University, and
honorary professor at the University of Queensland. He was awarded the Malcolm McIntosh Prize for Physical Scientist of the Year in Australia in 2001, was an IMS Medallion Lecturer in 2008, and is an Australian Laureate Fellow and a Fellow of the IMS.

Keynote - II
Making Sense of a Random World through Statistics

Prof. Geoff McLachlan,
University of Queensland, Brisbane, Australia

Abstract
With the growth in data in recent times, it is argued in this talk that there is a need for even more statistical methods in data mining. In so doing, we present some examples in which there is a need to adopt some fairly sophisticated statistical procedures (at least not off-the-shelf methods) to avoid misleading inferences being made about patterns in the data due to randomness. One example concerns the search for clusters in data. Having found an apparent clustering in a dataset, as evidenced in a visualisation of the dataset in some reduced form, the question arises of whether this clustering is representative of an underlying group structure or is merely due to random fluctuations. Another example concerns the supervised classification in the case of many variables measured on only a small number of objects. In this situation, it is possible to construct a classifier based on a relatively small subset of the variables that provides a perfect classification of the data (that is, its apparent error rate is zero). We discuss how statistics is needed to correct for the optimism in these results due to randomness and to provide a realistic interpretation.

Biography
Professor Geoffrey McLachlan's has a personal chair in statistics in UQ's School of Mathematics and Physics, and a joint appointment with the Institute for Molecular Bioscience. His research interests are in: data mining, statistical analysis of microarray, gene expression data, finite mixture models and medical statistics. His current research projects in statistics are in the related fields of classification, cluster and discriminant analyses, image analysis, machine learning, neural networks, and pattern recognition, and in the field of statistical inference. He has published six monographs, 20 book chapters and more than 140 articles in peer-reviewed literature. He currently holds an ARC Professorial Fellowship and is a chief investigator in the ARC Centre of Excellence in Bioinformatics. As an ISI Highly Cited Author, Professor McLachlan is part of an elite club of less than half of one percent of all published researchers in the world. In December 2010, Professor Geoff McLachlan was awarded the Statistical Society of Australia's highest honour - the Pitman Medal - to recognise his "outstanding achievement in, and contribution to, the discipline of statistics".

Contributed Papers

Automatic Detection of Cluster Structure Changes using Relative Density Self-Organizing Maps

Denny[1] **Pandu Wicaksono**[1] **Ruli Manurung**[1]

[1]Faculty of Computer Science, University of Indonesia, Indonesia
denny@cs.ui.ac.id, pandu.wicaksono91@ui.ac.id, maruli@cs.ui.ac.id

Abstract

Knowledge of clustering changes in real-life datasets is important in many contexts, such as customer attrition analysis and fraud detection. Organizations can use such knowledge of change to adapt business strategies in response to changing circumstances. Analysts should be able to relate new knowledge acquired from a newer dataset to that acquired from an earlier dataset to understand what has changed. There are two kind of clustering changes, which are: changes in clustering structure and changes in cluster memberships. The key contribution of this paper is a novel method to automatically detect structural changes in two snapshot datasets using ReDSOM. The method identifies emerging clusters, disappearing clusters, splitting clusters, merging clusters, enlarging clusters, and shrinking clusters. Evaluation using synthetic datasets demonstrates that this method can identify automatically structural cluster changes. Moreover, the changes identified in our evaluation using real-life datasets from the World Bank can be related to actual changes.

Keywords: temporal clustering, self-organizing maps, visualization.

1 Introduction

Clusters provide insights into archetypical behaviours across a population, for example from taxation records, insurance claims, customer purchases, and medical histories. A cluster is a set of similar observations of entities, but these observations are dissimilar to observations of entities in other clusters (Han et al. 2011). The process of assignment of these observations in a dataset into clusters based on similarity is called as cluster analysis (Jain et al. 1999). Clustering is an exploratory data analysis technique that aims to discover the underlying structures in data.

In order to respond to change, we need to be able to identify and understand change. One way to do so is by looking at changes of clusters in terms of their structure and memberships. This type of knowledge can help organizations develop strategies, such as in fraud detection and in customer attrition analysis. Moreover, analysts often have to understand change from two datasets acquired at two different points in time to adapt existing business strategies.

This paper presents a novel method that automatically detect changes in cluster structure from a clustering result $\mathcal{C}(\tau_1)$ obtained from dataset $\mathcal{D}(\tau_1)$ observed at time period τ_1 compared to clustering result $\mathcal{C}(\tau_2)$ obtained from dataset $\mathcal{D}(\tau_2)$, where $\tau_1 < \tau_2$. To understand what has changed, analysts need to relate new knowledge (often represented as models) acquired from a newer dataset $\mathcal{D}(\tau_2)$ to that acquired from an earlier dataset $\mathcal{D}(\tau_1)$.

In this paper, the clustering structure and the cluster assignment from a clustering result $\mathcal{C}(\tau_i)$ are defined as follow. The clustering structure of a clustering result is the shapes, densities, sizes, locations of each clusters, including similarity and distances between clusters. This structure also includes the partitioning of the data space $\mathbb{R}^d$ into Voronoi regions. On the other hand, the cluster assignments are assignments of each data vector in the dataset to a cluster. In other words, it is a partitioning of a set of data vectors into k non-overlapping and collectively exhaustive subsets.

In order to analyze clustering changes, this paper uses the ReDSOM method (Denny et al. 2010) to compare two Self-Organizing Maps $\mathcal{M}(\tau_1)$ and $\mathcal{M}(\tau_2)$ trained from two snapshot datasets $\mathcal{D}(\tau_1)$ and $\mathcal{D}(\tau_2)$. In Denny et al. (2010), changes in cluster structure are identified through visualizations by analysts. On the other hand, this paper aims to automatically detect structural cluster changes.

This paper compares SOMs since the SOMs capture the clustering structure of the underlying datasets. Changes between two related datasets can be discovered by comparing the resulting data mining models since each model captures specific characteristics of the respective dataset as in FOCUS framework (Ganti et al. 2002) and in PANDA framework (Bartolini et al. 2009). This approach is also called "contrast mining" or "change mining" (Boettcher 2011).

Most temporal clustering algorithms consider clustering of sequences of events or clustering of time series (Antunes & Oliveira 2001, Roddick et al. 2001, Roddick & Spiliopoulou 2002). The goals of this paper are different to the aims of time series clustering and clustering of sequences. Sequence clustering aims to group objects based on sequential structural characteristics. Time-series clustering, on the other hand, aims to cluster individual entities that have similar time-series patterns to discover and describe common trends in time series (Roddick & Spiliopoulou 2002). In contrast, this paper considers the clustering of observations of entities at points in time, and compares the clustering structures from *snapshot datasets* derived from *longitudinal data*. A snapshot dataset $\mathcal{D}(\tau_j)$ contains an observation of each entity $\mathcal{I}_i$ at one time period τ_j from a longitudinal data.

Entity $\mathcal{I}_i$ is defined as a *subject of interest*. For example, an entity can be a tax payer, a country, or a customer. An observation of entity $\mathcal{I}_i$ at time period τ_j is the measurements of entity $\mathcal{I}_i$ at time period τ_j based on attributes/features. These measurements are represented as data vector $\mathbf{x}_i(\tau_j)$. Longitudinal data, often referred to as *panel data*, are collection of repeated observations $\mathbf{x}_i(\tau_1), \ldots, \mathbf{x}_i(\tau_t)$ at multiple time periods $\tau_1, \ldots, \tau_t$ that track the same type of information/observation on the same set of entities $\mathcal{I}_i$ (Diggle et al. 1994). In other words, there are a number of observations associated for the same entity. An example of longitudinal data is various indicators for each country that are collected regularly by the World Bank. From these data, the snapshot datasets are the welfare condition of countries that are observed in the 1980s as one snapshot and the 1990s as the subsequent snapshot. These two observations of a country are represented as two data vectors $\mathbf{x}_i(\tau_1)$ and $\mathbf{x}_i(\tau_2)$. Analysis of change from longitudinal data leads to quite a different approach to the process of cluster analysis.

The remainder of the paper is organized as follows. The next section discusses related works in temporal cluster analysis. Sections 3 then reviews structural cluster changes detection and the relative density definition. The contribution of this paper is discussed in Section 4. Section 5 discusses our experiments on the threshold parameters used in our algorithm. The application of the algorithm with synthetic amd real-life datasets is then discussed in Section 6. Conclusions and future work are provided in Section 7.

2 Related Works

Existing methods can be differentiated based on the types of data they can handle, which are: data stream, partitioned dataset, snapshot longitudinal, univariate time series, and trajectories. Clustering snapshot datasets has not received much attention in temporal clustering. Research has focused mostly on clustering of sequences, time series clustering, data stream clustering, and trajectory clustering. The ReDSOM method clusters snapshot datasets and can contrast the clustering results between two snapshots (Denny et al. 2010). This method can also be used for multivariate time series data once transformed into snapshot datasets. However, ReDSOM does not detect structural changes automatically.

In identifying structural changes, there are a number of different approaches. MONIC (Spiliopoulou et al. 2006) and MClusT/MEC (Oliveira & Gama 2010) define clusters as set of objects. Therefore, structural changes is defined based on overlap of cluster members between two clusters of two time periods. Furthermore, MONIC uses ageing of observations to monitor evolutions of clusters which is appropriate for clustering data stream. In contrast, this paper uses snapshots datasets to analyze changes of clustering over time. MONIC+ (Ntoutsi et al. 2009) tries to generalize MONIC to include more cluster types which are clusters as geometrical objects and as distribution. When clusters are defined as geometrical objects, cluster overlap is defined by intersection of area of the two clusters. It is not clear how area is defined in MONIC+, especially for high dimensional datasets. On the other hand, this paper defines cluster overlap by the intersection of the Voronoi region of two clusters. Adomavicius & Bockstedt (2008) uses between-cluster distances to detect structural changes. Kalnis et al. (2005) uses moving cluster, which is defined based on a set of objects and spatial location. ReDSOM and Aggarwal (2005) use kernel density estimation to detect structural changes. However, the work in Aggarwal (2005) is designed for stream clustering and the concept of velocity density estimation is limited to one attribute. ReDSOM, on the other hand, is used to analyze multivariate snapshot datasets by comparing density estimation and communicate the results using visualization. Recently, Held & Kruse (2013) presented a method based on MONIC to visualize the dynamics of cluster evolution. The method were extended to detect cluster rebirth, which is a missing cluster in the previous time period that is emerged again.

The SOM-based methods to analyze cluster using multiple temporal datasets can be categorized into three approaches: chronological, temporal, and sequential cluster analysis. *Chronological cluster analysis* produces a different SOM and visualizations for each time period (Skupin & Hagelman 2005). *Temporal cluster analysis*, on the other hand, trains a single SOM with combined data from all time period (Skupin & Hagelman 2005). The limitation of this approach is its inability to detect structural clustering changes. *Sequential cluster analysis*, such as SbSOM (Fukui et al. 2008), trains a single SOM with sequential data. This paper uses chronological cluster analysis approach (Denny & Squire 2005). Given two snapshot datasets $\mathcal{D}(\tau_1)$ and $\mathcal{D}(\tau_2)$, the maps $\mathcal{M}(\tau_1)$ and $\mathcal{M}(\tau_2)$ are trained using their respective datasets. When training map $\mathcal{M}(\tau_2)$, the map $\mathcal{M}(\tau_1)$ is used as the initial map to preserve the orientation of the trained map.

3 Structural Cluster Changes Detection using ReDSOM

In ReDSOM, changes of clustering structure are identified from changes of density estimations at the same location over time. The plot of density estimation can provide useful characteristics in the data, such as skewness, multi-modality, and clustering structure. Density estimation is the construction of an estimate $\hat{f}(x)$ of an unobservable underlying density function $f(x)$ based on observed data (Silverman 1986). This estimation is also ideal to present the data to non-mathematicians, as it is fairly easy to understand (Silverman 1986). Dense regions in data space are good candidates for clusters. Conversely, very low density regions most likely contain outliers. Therefore, clusters can be found by density estimates, such as in the DENCLUE algorithm (Hinneburg & Keim 2003).

Kernel density estimation (KDE) is a non-parametric method to estimate the probability density function of the observed data at a given point (Silverman 1986). KDE is a non-parametric method, as it does not make assumption about the distribution of the observed data. This method is also known as *Parzen-Rosenblatt* window method (Parzen 1962, Rosenblatt 1956).

The approximation of density estimation can be calculated faster using prototype vectors produced by a VQ method (Hinneburg & Keim 2003, Macqueen 1967). Since SOM is also a VQ method, the prototype vectors of map $\mathcal{M}$ that is trained on dataset $\mathcal{D}$ can be used to estimate density of dataset $\mathcal{D}$. Data vector $\mathbf{x}_i \in \mathcal{D}$ can be approximated/represented using the closest prototype vector $\mathbf{m}_{b_i}$ with the accuracy measured by quantization error. In other words, the data space $\mathbb{R}^d$ is partitioned into $|\mathcal{M}|$ Voronoi regions without leaving any gaps or overlaps. A Voronoi region VR_j of a prototype vector $\mathbf{m}_j$ or cluster $C_j \in \mathcal{C}$ is

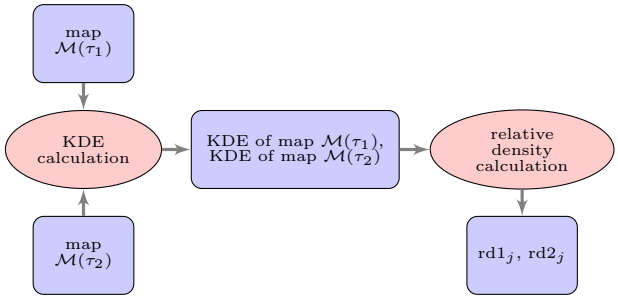


Figure 1: Relative density calculation

defined as the set of all points in $\mathbb{R}^d$ that are closest to the cluster centroid/prototype vector $\mathbf{m}_j$. Each partition contains the data vectors that are the nearest to its partition prototype vector compared to other prototype vectors on the map (Voronoi set).

This research uses two-level clustering that uses a SOM as an abstraction layer of the dataset (Vesanto & Alhoniemi 2000). The prototype vectors are clustered using a partitional clustering technique or a hierarchical agglomerative nesting (AGNES) technique to form the final clusters. The optimal clustering results are then selected based on cluster validity indexes.

When comparing maps $\mathcal{M}(\tau_1)$ and $\mathcal{M}(\tau_2)$, density estimation $\hat{f}_{h,\mathcal{M}(\tau_1)}(\mathbf{v})$ centred at the location of vector $\mathbf{v} \in \mathbb{R}^d$ on map $\mathcal{M}(\tau_1)$ might be different compared to density estimation $\hat{f}_{h,\mathcal{M}(\tau_2)}(\mathbf{v})$ at the same location on map $\mathcal{M}(\tau_2)$. When the density centred at the location of vector $\mathbf{v}$ in dataset $\mathcal{D}(\tau_2)$ is lower than in dataset $\mathcal{D}(\tau_1)$, the density estimation $\hat{f}_{h,\mathcal{M}(\tau_2)}(\mathbf{v})$ on map $\mathcal{M}(\tau_2)$ centred at the location of vector $\mathbf{v}$ is lower compared to the density estimation $\hat{f}_{h,\mathcal{M}(\tau_1)}(\mathbf{v})$ at the same location on map $\mathcal{M}(\tau_1)$, and vice-versa. Therefore, relative density $\mathrm{RD}_{\mathcal{M}(\tau_2)/\mathcal{M}(\tau_1)}(\mathbf{v})$ is defined as the log ratio of the density estimation centred at the location of vector $\mathbf{v}$ on map $\mathcal{M}(\tau_2)$ to the density estimation centred at the same location on the reference map $\mathcal{M}(\tau_1)$ (Denny et al. 2010):

$$\mathrm{RD}_{\mathcal{M}(\tau_2)/\mathcal{M}(\tau_1)}(\mathbf{v}) = log_2\left(\frac{\hat{f}_{h,\mathcal{M}(\tau_2)}(\mathbf{v})}{\hat{f}_{h,\mathcal{M}(\tau_1)}(\mathbf{v})}\right) \quad (1)$$

Let $\mathrm{rd1}_j \leftarrow \mathrm{RD}_{\mathcal{M}(\tau_2)/\mathcal{M}(\tau_1)}(\mathbf{m}_j(\tau_1))$ as a shorthand for the relative density at the location of prototype vector $\mathbf{m}_j(\tau_1)$ on map $\mathcal{M}(\tau_2)$ compared to reference map $\mathcal{M}(\tau_1)$. Similarly, let $\mathrm{rd2}_j \leftarrow \mathrm{RD}_{\mathcal{M}(\tau_2)/\mathcal{M}(\tau_1)}(\mathbf{m}_j(\tau_2))$ be the relative density at the location of prototype vector $\mathbf{m}_j(\tau_2)$ on map $\mathcal{M}(\tau_2)$ compared to reference map $\mathcal{M}(\tau_1)$. Figure 1 shows the relative density calculation.

To visualize the relative density of the locations of all prototype vectors $\mathbf{m}_j(\tau_1)$, the values of $\mathrm{rd1}_j$ are visualized on a map $\mathcal{M}(\tau_1)$ in a gradation of blue for positive values and red for negative values. ReDSOM visualization uses diverging colour scheme as this visualization has a critical mid-point which is zero (no change in density). Values of relative density over $+\delta$ are represented as dark blue, and values less than $-\delta$ are represented as dark red, where δ is a density threshold parameter. Based on experiments, the value of δ is set to 3, which means a region is considered as emerging or disappearing when its density increase by eightfold or one-eightfold, respectively.

Visualization of $\mathrm{rd1}_j$ on map $\mathcal{M}(\tau_1)$ alone cannot be used to detect emerging regions in period τ_2. Regions that emerge in dataset $\mathcal{D}(\tau_2)$ are not represented on $\mathcal{M}(\tau_1)$ because map $\mathcal{M}(\tau_1)$ only represents populated Voronoi regions of dataset $\mathcal{D}(\tau_1)$ due to the VQ property. Therefore, $\mathrm{rd2}_j$ is used to detect emerging regions because the emerging regions at period τ_2 are represented on map $\mathcal{M}(\tau_2)$. The $\mathrm{rd2}_j$ value for the Voronoi region of prototype vector $\mathbf{m}_j(\tau_2)$ would be high because the density $\hat{f}_{h,\mathcal{M}(\tau_2)}(\mathbf{m}_j(\tau_2))$ is high and the density $\hat{f}_{h,\mathcal{M}(\tau_1)}(\mathbf{m}_j(\tau_2))$ is low. In sum, values of $\mathrm{rd2}_j$ should be visualized on map $\mathcal{M}(\tau_2)$ to detect emerging clusters.

For similar reason, visualization $\mathrm{rd2}_j$ on map $\mathcal{M}(\tau_2)$ alone cannot be used to detect disappearing regions on map $\mathcal{M}(\tau_2)$. Disappearing regions are not represented by prototype vectors of map $\mathcal{M}(\tau_2)$. As a result, visualization of $\mathrm{rd1}_j$ on map $\mathcal{M}(\tau_1)$ is used to detect disappearing regions. The disappearing regions exist on map $\mathcal{M}(\tau_1)$, but no longer exist on map $\mathcal{M}(\tau_2)$.

To discover changes in clustering structure, Denny & Squire (2005) proposed cluster colour linking techniques, which is a techniques to link two clustering results from two snapshot datasets $\mathcal{D}(\tau_1)$ and $\mathcal{D}(\tau_2)$. The *cluster colour linking* technique, then, is used to visualize the first clustering result $\mathcal{CM}(\tau_1)$ on the second map $\mathcal{M}(\tau_2)$. The cluster colour of the nodes of map $\mathcal{M}(\tau_2)$ are determined by the cluster colour of their BMU (best matching unit) on map $\mathcal{M}(\tau_1)$. Given the colour of node j on map $\mathcal{M}(\tau_1)$ as $nodeCluster(j, \mathcal{M}(\tau_1))$, the colour of node j on map $\mathcal{M}(\tau_2)$ is calculated as:

$$nodeCluster\big(j, \mathcal{M}(\tau_2)\big) = nodeCluster\big(BMU\big(\mathbf{m}_j(\tau_2), \mathcal{M}(\tau_1)\big),\ \mathcal{M}(\tau_1)\big) \quad (2)$$

Since map $\mathcal{M}(\tau_2)$ follows the distribution of dataset $\mathcal{D}(\tau_2)$, the size of each cluster on map $\mathcal{M}(\tau_2)$ would follow as well.

A cluster $C_i(\tau_2) \in \mathcal{C}(\tau_2)$ is said to have emerged at time period τ_2 when the density of the cluster $C_i(\tau_2)$ occupies a well separated region that has significantly increased density in dataset $\mathcal{D}(\tau_2)$ ($\mathrm{rd2}_j \geq +\delta$) compared the region's density in the previous dataset $\mathcal{D}(\tau_1)$.

$$\frac{\big|\{\mathbf{m}_j(\tau_2) \in C_i(\tau_2) \mid \mathrm{rd2}_j \geq +\delta\}\big|}{\big|C_i(\tau_2)\big|} \geq \theta_{\text{emerging}} \quad (3)$$

A cluster $C_i(\tau_1) \in \mathcal{C}(\tau_1)$ is said to have disappeared at time period τ_2 when the density of region $\mathbf{m}_j(\tau_1) \in C_i(\tau_1)$ is significantly decreased ($\mathrm{rd1}_j \leq -\delta$) in the dataset $\mathcal{D}(\tau_2)$ compared to the previous dataset $\mathcal{D}(\tau_1)$.

$$\frac{\big|\{\mathbf{m}_j(\tau_1) \in C_i(\tau_1) \mid \mathrm{rd1}_j \leq -\delta\}\big|}{\big|C_i(\tau_1)\big|} \geq \theta_{\text{disappearing}} \quad (4)$$

Unlike a new cluster that resides in a previously unoccupied region, split clusters do not occupy a new region. A cluster split can be identified when a cluster in map $\mathcal{M}(\tau_1)$ can be separated in map $\mathcal{M}(\tau_2)$. ReDSOM visualization has to show that both split clusters in period τ_2 do not occupy new region ($0 \leq \mathrm{rd2}_j \leq \delta$). A cluster $C_i(\tau_1)$ is said to have split at time period τ_2 when the Voronoi region of cluster $C_i(\tau_1)$ is occupied by two or more well separated clusters $C_{k1}(\tau_2), \ldots, C_{kn}(\tau_2)$ in the dataset $\mathcal{D}(\tau_2)$.

Cluster merging occurs when two clusters on map $\mathcal{M}(\tau_1)$ are no longer well separated on map $\mathcal{M}(\tau_2)$. Cluster merging can be identified by cluster colour linking when two clusters on map $\mathcal{M}(\tau_1)$ are merged into one cluster outline on map $\mathcal{M}(\tau_2)$. Cluster merging is different to lost cluster where one of the clusters shrinks significantly thus having $\mathrm{rd1}_j < -\delta$. In cluster merging, the density of gap between clusters should increased in a way that can be verified using ReDSOM visualization. Clusters $C_{i1}(\tau_1), \ldots, C_{in}(\tau_1)$ are said to have merged into $C_k(\tau_2)$ at time period τ_2 when the gap between the clusters is disappear in the dataset $\mathcal{D}(\tau_2)$.

Cluster $C_i(\tau_2)$ is said to have enlarged at time period τ_2 when the part of the cluster region has significantly increased density in the dataset $\mathcal{D}(\tau_2)$.

$$\theta_{\text{overlap}} \leq \frac{\left|\left\{\mathbf{m}_j(\tau_2) \in C_i(\tau_2) \mid \mathrm{rd2}_j \geq \delta \right\}\right|}{\left|C_i(\tau_2)\right|} < \theta_{\text{emerging}} \quad (5)$$

Similarly, cluster contraction can be identified as a lost region which does not have a good separation to its neighbours. To put this another way, only a part of a cluster has disappeared. Cluster $C_i(\tau_1)$ is said to have contracted at time period τ_2 when the clusters occupies smaller region in the dataset $\mathcal{D}(\tau_2)$.

$$\theta_{\text{overlap}} \leq \frac{\left|\left\{\mathbf{m}_j(\tau_1) \in C_i(\tau_1) \mid \mathrm{rd1}_j \leq -\delta \right\}\right|}{\left|C_i(\tau_1)\right|} < \theta_{\text{disappearing}} \quad (6)$$

If a cluster does not fall into the above categories, the cluster $C_i(\tau_1)$ is evaluated wheter it is overlapped with another cluster $C_j(\tau_2)$. As above, the overlap is determined based on the Voronoi region of their prototype vectors.

4 Automatic Structural Changes Detection

Based on the ReDSOM reviewed in the previous section, this paper develops a new algorithm that detects the structural cluster changes based on the relative density measurements. The algorithm takes the clustering results, relative density measurements, and hit counts to detect structural changes as shown in Algorithm 1 and Figure 2. In general, the steps are: initializations, detecting disappearing and contracting clusters, detecting emerging and enlarging clusters, detecting merging clusters, detecting splitting clusters, and detecting overlapping clusters.

There are two ways to calculate the number of prototype vector members that disappear ($totalDarkRedRegionT1$), emerge ($totalDarkBlueRegionT2$), or overlap: weighted and unweighted calculations. Unlike the unweighted calculation where each prototype vector counts as one, in the weighted calculation, each prototype vector counts as the number of data vectors mapped to the prototype vector (hit count). Algorithm 2 shows the initialization of the algorithm.

The algorithm starts by discovering disappearing and emerging clusters (Algorithms 3 and 4). The ratio is calculated using $totalDarkRedRegionT1$ and $totalDarkBlueRegionT2$. When the ratio of the 'dark red' region in a cluster $C_i(\tau_1)$ is compared to the whole cluster $C_i(\tau_1)$ above $\theta_{\text{disappearing}}$, the cluster is considered to disappear in period τ_2. Similarly, when the ratio of the 'dark blue' region in a cluster $C_j(\tau_2)$ is compared to the whole cluster $C_j(\tau_2)$ above

Algorithm 1: Automatic structural cluster changes detection algorithm.

Input: array of cluster assignment in τ_1 CT1, array of cluster assignment in τ_2 CT2, array of cluster color linking assignment CT21, array of relative density rd1 and rd2, array of hit count in τ_1 HT1, array of hit count in τ_2 HT2
Output: List of structural changes LC for all cluster in τ_1 and τ_2

1 **initialization** (Algorithm 2)
2 **detect disappearing and contracting clusters** (Algorithm 3)
3 **detect emerging and enlarging clusters** (Algorithm 4)
4 **detect merging clusters** (Algorithm 5)
5 **detect splitting clusters** (Algorithm 6)
6 **detect overlapping clusters** (Algorithm 7)
7 **return** LC

Algorithm 2: Initialization.

```
1  LC ← ∅
2  listUnknownChangeT1
   ← {C_1(τ_1), ..., C_numberClusterT1(τ_1)}
3  listUnknownChangeT2
   ← {C_1(τ_2), ..., C_numberClusterT2(τ_2)}
4  contractingClusterT1 ← ∅
5  enlargingClusterT2 ← ∅
6  for j = 0 to |M| do
7      if weighted then
8          weightT1[j] ← HT1[j]
9          weightT2[j] ← HT2[j]
10     else
11         weightT1[j] ← 1
12         weightT2[j] ← 1
13     totalClusterMemberT1[CT1[j]] +=
       weightT1[j]
14     totalClusterMemberT2[CT2[j]] +=
       weightT2[j]
15     totalClusterMemberT21[CT21[j]] +=
       weightT2[j]
16     if rd1_j ≤ −3 then
17         totalDarkRedRegionT1[CT1[j]] +=
           weightT1[j]
18     if rd2_j ≥ 3 then
19         totalDarkBlueRegionT2[CT2[j]] +=
           weightT2[j]
```

θ_{emerging}, the cluster is considered to emerge in period τ_2. If a cluster has some 'dark red' or 'dark blue' region, the cluster is flagged as contracting and enlarging, respectively. This kind of cluster changes still needs to be checked further if the cluster participates in cluster splitting or cluster merging.

In the second phase, the clusters $C_i(\tau_1)$ and $C_j(\tau_2)$ are checked whether they experience splitting or merging (Algorithms 5 and 6). When two or more clusters $C_{i_1}(\tau_1), C_{i_2}(\tau_1), \ldots, C_{i_n}(\tau_1)$ from period τ_1 overlaps with the region of cluster $C_j(\tau_2)$, the clusters $C_{i_1}(\tau_1), C_{i_2}(\tau_1), \ldots, C_{i_n}(\tau_1)$ are said to merge into cluster $C_j(\tau_2)$. The overlap between cluster $C_i(\tau_1)$ and $C_j(\tau_2)$ is defined as the ratio of the Voronoi region of $C_i(\tau_1)$ which overlaps/intersects with the Voronoi region of $C_j(\tau_2)$ to the whole cluster $C_{i_1}(\tau_1)$. When this overlap is above the θ_{merging} threshold, most of

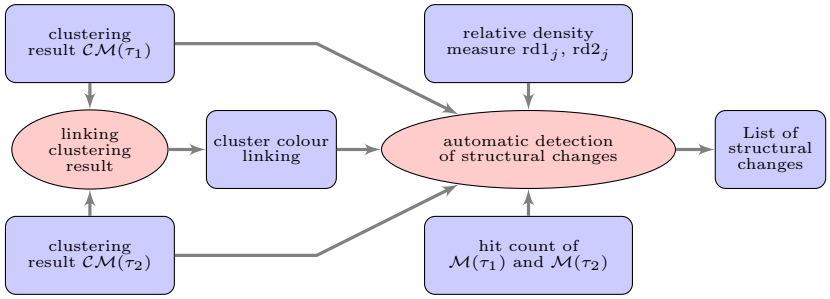


Figure 2: Automatic detection of changes in cluster structure.

Algorithm 3: Detecting disappearing and contracting cluster.

```
1 foreach C_i(τ1) ∈ listUnknownChangeT1 do
2     if totalDarkRedRegionT1[i] > 0 then
3         ratio ← totalDarkRedRegionT1[i] /
          totalClusterMemberT1[i]
4         if ratio ≥ θ_disappearing then
5             LC ← LC
              ∪{(C_i(τ1), ∅, disappearing)}
6             listUnknownChangeT1 ←
              listUnknownChangeT1 − {C_i(τ1)}
7         else
8             contractingClusterT1 ←
              contractingClusterT1 ∪{C_i(τ1)}
```

Algorithm 4: Detecting emerging and enlarging cluster.

```
1 foreach C_j(τ2) ∈ listUnknownChangeT2 do
2     if totalDarkBlueRegionT2[j] > 0 then
3         ratio ← totalDarkBlueRegionT2[j] /
          totalClusterMemberT2[j]
4         if ratio ≥ θ_emerging then
5             LC ← LC ∪{(∅, C_j(τ2), emerging)}
6             listUnknownChangeT2 ←
              listUnknownChangeT2 − {C_j(τ2)}
7         else
8             enlargingClusterT2 ←
              enlargingClusterT2 ∪{C_j(τ2)}
```

Algorithm 5: Detecting merging clusters.

```
  // Detecting overlap between all pair of
     cluster from period τ1 and τ2
1 foreach C_j(τ2) ∈ listUnknownChangeT2 do
2     listOverlapClusterT1 ← ∅
3     foreach C_i(τ1) ∈ listUnknownChangeT1 do
4         overlapCount ← 0
5         for mapUnit = 1 → |M(τ2)| do
6             if CT21[mapUnit] = i ∧
              CT2[mapUnit] = j then
                  // when the map unit of
                     M(τ2) is assigned to
                     both C_i(τ1) and C_j(τ2)
7                 overlapCount +=
                  weightT2[mapUnit]
8         ratio ← overlapCount /
          totalClusterMemberT21[i]
9         if ratio ≥ θ_merging then
10            listOverlapClusterT1 ←
              listOverlapClusterT1 ∪{C_i(τ1)}
11    if |listOverlapClusterT1| ≥ 2 then
12        foreach C_i(τ1) ∈ listOverlapClusterT1
          do
13            LC ← LC
              ∪{(C_i(τ1), C_j(τ2), merging)}
14            listUnknownChangeT1 ←
              listUnknownChangeT1 − {C_i(τ1)}
15        listUnknownChangeT2 ←
          listUnknownChangeT2 − {C_j(τ2)}
```

$C_i(\tau_1)$ is part of $C_j(\tau_2)$. Merging clusters require two or more clusters from period τ_1 that are part of cluster $C_j(\tau_2)$. The overlap is calculated and visualized using the cluster colour linking technique described earlier. Detecting cluster splitting is basically the mirror case of detecting cluster merging.

In the last phase, clusters $C_i(\tau_1)$ and $C_j(\tau_2)$ that are not yet classified are checked if their region partially overlap. If the ratio of overlap above θ_{overlap} and the cluster $C_i(\tau_1)$ was flagged as contracting or enlarging, cluster $C_i(\tau_1)$ is considered to be contracting or enlarging in period τ_2, respectively. Otherwise, the clusters with the ratio of overlap above θ_{overlap} is considered as overlap.

The complexity of the whole algorithm is $\mathcal{O}\left(|\mathcal{C}(\tau_1)| \cdot |\mathcal{C}(\tau_2)| \cdot |\mathcal{M}|\right)$, where $|\mathcal{C}(\tau_i)|$ is the number of cluster in period τ_i and $|\mathcal{M}|$ is the number of prototype vectors in map $\mathcal{M}$. The running time for Algorithm 2 is bounded by $\mathcal{O}\left(|\mathcal{M}|\right)$. The complexities of Algorithms 3 and 4 are $\mathcal{O}\left(|\mathcal{C}(\tau_1)|\right)$ and $\mathcal{O}\left(|\mathcal{C}(\tau_2)|\right)$ respectively. The running time for Algorithms 5–7 are bounded by $\mathcal{O}\left(|\mathcal{C}(\tau_1)| \cdot |\mathcal{C}(\tau_2)| \cdot |\mathcal{M}|\right)$.

5 Experiments on the Threshold Parameters

To determine the threshold parameters for each type of cluster changes, experiments on different values of these threshold parameters are performed on synthetic datasets. In these synthetic datasets, only one structural cluster change is introduced in the dataset $\mathcal{D}(\tau_2)$. In total, there are eight pairs of synthetic datasets. The threshold values used range between 0.3 and 1.0 in increments of 0.1. When the threshold value is too high, the algorithm cannot detect the changes. On the other hand, when the threshold value is too low, the algorithm might detect false positive changes. While this paper provides the guidelines for setting the value of these threshold parameter, these parameters can be tuned to suit the need of analysis.

These experiments showed that the weighted calculation allows the use of higher threshold values. For the datasets where an emerging cluster exists in the dataset $\mathcal{D}(\tau_2)$, the emerging cluster is no longer detected at $\theta_{emerging} = 0.8$ using the unweighted version. On the other hand, the weighted version can

Algorithm 6: Detecting splitting clusters.

```
1  foreach C_i(τ1) ∈ listUnknownChangeT1 do
2      listOverlapClusterT2 ← ∅
3      foreach C_j(τ2) ∈ listUnknownChangeT2 do
4          overlapCount ← 0
5          for mapUnit = 1 → |M(τ2)| do
6              if CT21[mapUnit] = i ∧
               CT2[mapUnit] = j then
7                  overlapCount +=
                   weightT2[mapUnit]
8          ratio ← overlapCount /
           totalClusterMemberT2[i]
9          if ratio ≥ θ_splitting then
10             listOverlapClusterT2 ←
               listOverlapClusterT2 ∪ {C_i(τ1)}
11     if |listOverlapClusterT2| ≥ 2 then
12         foreach C_j(τ2) ∈ listOverlapClusterT2
           do
13             LC ← LC
               ∪{(C_i(τ1), C_j(τ2), splitting)}
14             listUnknownChangeT2 ←
               listUnknownChangeT2 − {C_j(τ2)}
15         listUnknownChangeT1 ←
           listUnknownChangeT1 − {C_i(τ1)}
```

Algorithm 7: Detecting overlapping clusters.

```
1  foreach C_i(τ1) ∈ listUnknownChangeT1 do
2      foreach C_j(τ2) ∈ listUnknownChangeT2 do
3          overlapCount ← 0
4          for mapUnit = 1 → |M(τ2)| do
5              if CT21[mapUnit] = i ∧
               CT2[mapUnit] = j then
6                  overlapCount +=
                   weightT2[mapUnit]
7          ratio ← overlapCount /
           totalClusterMemberT2[j]
8          if ratio ≥ θ_overlapping then
9              if C_i(τ1) ∈ contractingClusterT1
               then
10                 LC ←
                   LC∪{C_i(τ1), C_j(τ2), contracting}
11             else if
               C_j(τ2) ∈ enlargingClusterT2 then
12                 LC ←
                   LC ∪ {C_i(τ1), C_j(τ2), enlarging}
13             else
14                 LC ← LC ∪
                   {C_i(τ1), C_j(τ2), overlapping}
15             listUnknownChangeT1 ←
               listUnknownChangeT1 − {C_i(τ1)}
16             listUnknownChangeT2 ←
               listUnknownChangeT2 − {C_j(τ2)}
17 foreach C_i(τ1) ∈ listUnknownChangeT1 do
18     LC ← LC ∪ {C_i(τ1), −, −}
19 foreach C_j(τ2) ∈ listUnknownChangeT2 do
20     LC ← LC ∪ {−, C_j(τ2), −}
```

detect the emerging cluster up to $\theta_{emerging} = 0.9$. The higher threshold value means that the weighted version is more sensitive to detect structural cluster changes. Therefore, the subsequent experiments use the weighted version with the threshold parameter 0.5.

Table 1: Threshold values used in the subsequent experiments.

Threshold name	*Threshold value*
Emerging threshold	0.5
Disappearing threshold	0.5
Merging threshold	0.5
Splitting threshold	0.5
Overlapping threshold	0.6

The experiment on the datasets where a cluster disappears in dataset $\mathcal{D}(\tau_2)$ shows that the weighted calculation can detect the changes up to $\theta_{disappearing} = 0.6$, while the unweighted version can detect up to $\theta_{disappearing} = 0.5$. On detecting merging clusters, the weighted calculation can detect the changes up to $\theta_{merging} = 0.8$, while the unweighted version can detect up to $\theta_{merging} = 0.7$. The experiment on the datasets where a cluster splits into two clusters in dataset $\mathcal{D}(\tau_2)$ shows that the weighted calculation can detect the changes up to $\theta_{splitting} = 1.0$, while the unweighted version can detect up to $\theta_{splitting} = 0.9$. Lastly, when the algorithm is evaluated on the datasets where a cluster $C_i(\tau_1)$ overlaps with a cluster $C_j(\tau_2)$, the weighted calculation can detect the changes up to $\theta_{overlapping} = 0.6$, while the unweighted version can detect up to $\theta_{splitting} = 0.7$. Therefore, the subsequent experiments use the threshold parameter shown in Table 1.

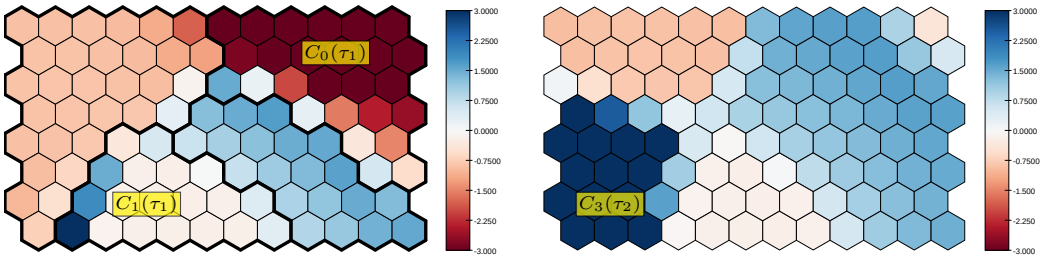
6 Evaluations on Synthetic and Real-Life Datasets

To evaluate the algorithm and the threshold values, this research uses two pairs of synthetic datasets with multiple structural changes and two pairs of real-life datasets. The first synthetic datasets 'lost-new' contains an emerging cluster and a disappearing cluster as shown in Figure 3. The second synthetic datasets contains a splitting cluster, merging clusters, expanding clusters, and contracting clusters.

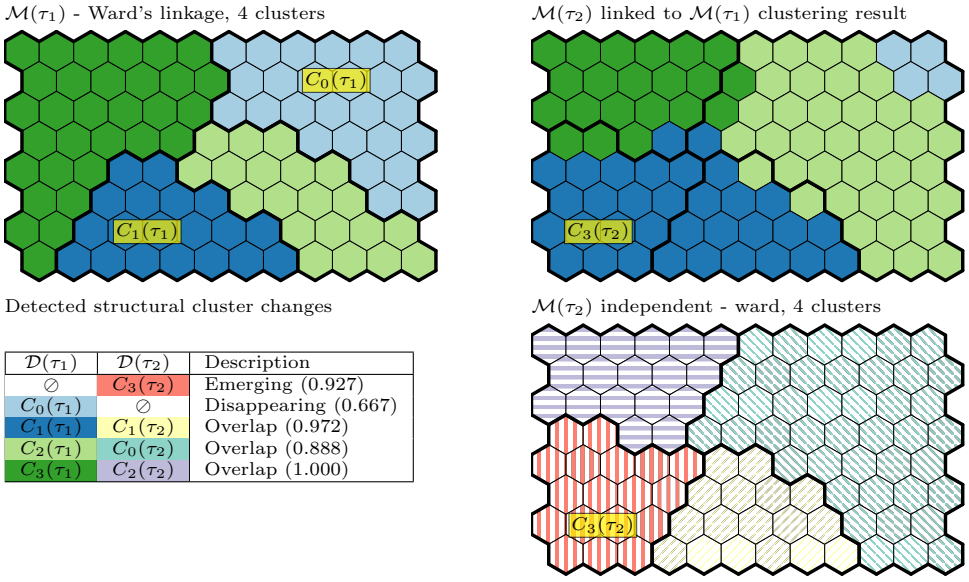
Analysis using ReDSOM and cluster colour linkage on the 'lost-new' dataset shows that there are a disappearing cluster and an emerging cluster. The scatter plot of both dataset $\mathcal{D}(\tau_1)$ (*red dots*) and $\mathcal{D}(\tau_2)$ (*blue pluses*) shown in Figure 3 indicates a lost cluster and an emerging cluster. The ReDSOM visualization shown Figure 4(a) indicates that cluster $C_0(\tau_1)$ is disappearing in the map $\mathcal{M}(\tau_2)$. Furthermore, the ReDSOM visualization of the map $\mathcal{M}(\tau_2)$ (*right*) identifies an emerging cluster $C_3(\tau_2)$. Similarly, based on analysis using cluster colour linkage on Figure 4(b), the cluster $C_0(\tau_1)$ on the map $\mathcal{M}(\tau_1)$ no longer exists in the map $\mathcal{M}(\tau_2)$. An emerging cluster $C_3(\tau_2)$ emerged on the map $\mathcal{M}(\tau_2)$.

Evaluation on the 'lost-new' dataset shows that the proposed algorithm able to identify structural cluster changes correctly. The algorithm proposed in this paper produced the table shown in Figure 4(b) (*bottom left*). The algorithm correctly identifies cluster $C_3(\tau_2)$ as an emerging cluster. Furthermore, the algorithm also identifies cluster $C_0(\tau_1)$ as a disappearing cluster. No significant structural changes detected in the other clusters in $\mathcal{M}(\tau_1)$: $C_1(\tau_1)$, $C_2(\tau_1)$, and $C_3(\tau_1)$.

The World Development Indicator (WDI) dataset (World Bank 2003) is a multi-variate temporal dataset covering 205 countries. The experiments compare the clustering structure based on 25 selected indicators that reflect different aspects of welfare,



(a) The ReDSOM visualization of the $\mathcal{M}(\tau_1)$ (*left*) identifies that cluster $C_0(\tau_1)$ is disappearing in the map $\mathcal{M}(\tau_2)$. The ReDSOM visualization of the map $\mathcal{M}(\tau_2)$ (*right*) identifies an emerging cluster $C_3(\tau_2)$.



$\mathcal{D}(\tau_1)$	$\mathcal{D}(\tau_2)$	Description
⊘	$C_3(\tau_2)$	Emerging (0.927)
$C_0(\tau_1)$	⊘	Disappearing (0.667)
$C_1(\tau_1)$	$C_1(\tau_2)$	Overlap (0.972)
$C_2(\tau_1)$	$C_0(\tau_2)$	Overlap (0.888)
$C_3(\tau_1)$	$C_2(\tau_2)$	Overlap (1.000)

(b) The clustering result of the map $\mathcal{M}(\tau_1)$ is shown using colour on both the map $\mathcal{M}(\tau_1)$ (*top left*) and the map $\mathcal{M}(\tau_2)$ (*top right*). The independent clustering result of the map $\mathcal{M}(\tau_2)$ is shown using thick outlines and patterns on the map $\mathcal{M}(\tau_2)$ (*bottom right*). The cluster $C_0(\tau_1)$ on the map $\mathcal{M}(\tau_1)$ no longer exists in the map $\mathcal{M}(\tau_2)$. An emerging cluster $C_3(\tau_2)$ emerged on the map $\mathcal{M}(\tau_2)$.

Figure 4: ReDSOM, distance matrix, and clustering result visualizations of the synthetic 'lost-new' datasets. Map $\mathcal{M}(\tau_1)$ shown on the *left hand side* and $\mathcal{M}(\tau_2)$ on the *right hand side*.

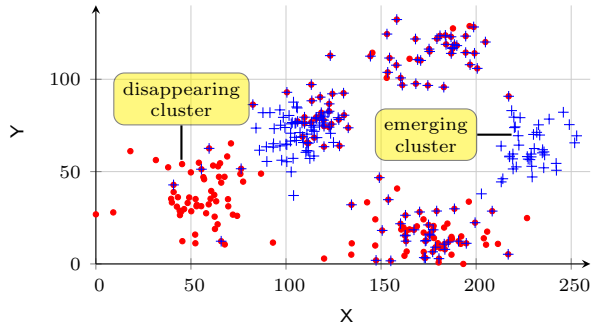


Figure 3: The scatter plot of the synthetic datasets $\mathcal{D}(\tau_1)$ (*red dots*) and $\mathcal{D}(\tau_2)$ (*blue pluses*) that contains emerging cluster and disappearing cluster.

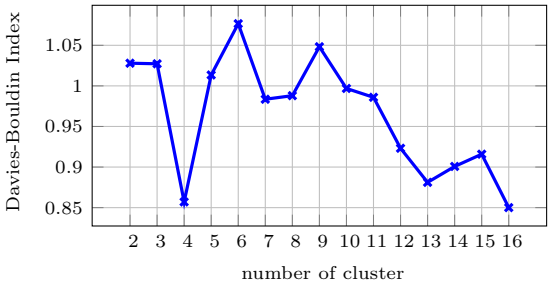
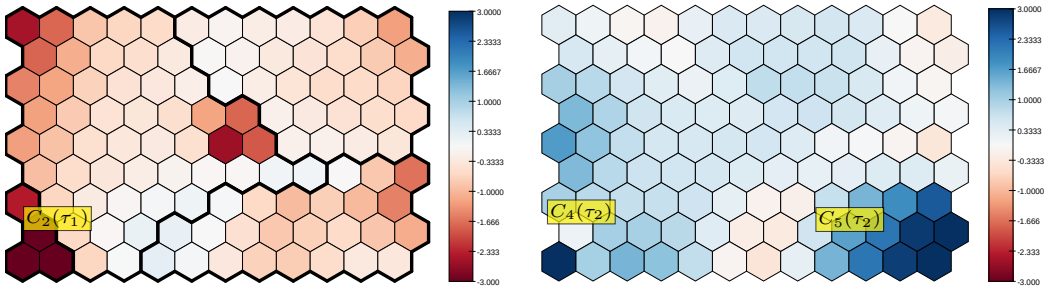


Figure 6: The plot of the Davies-Bouldin Index for k-means clustering result of the 1980s dataset.

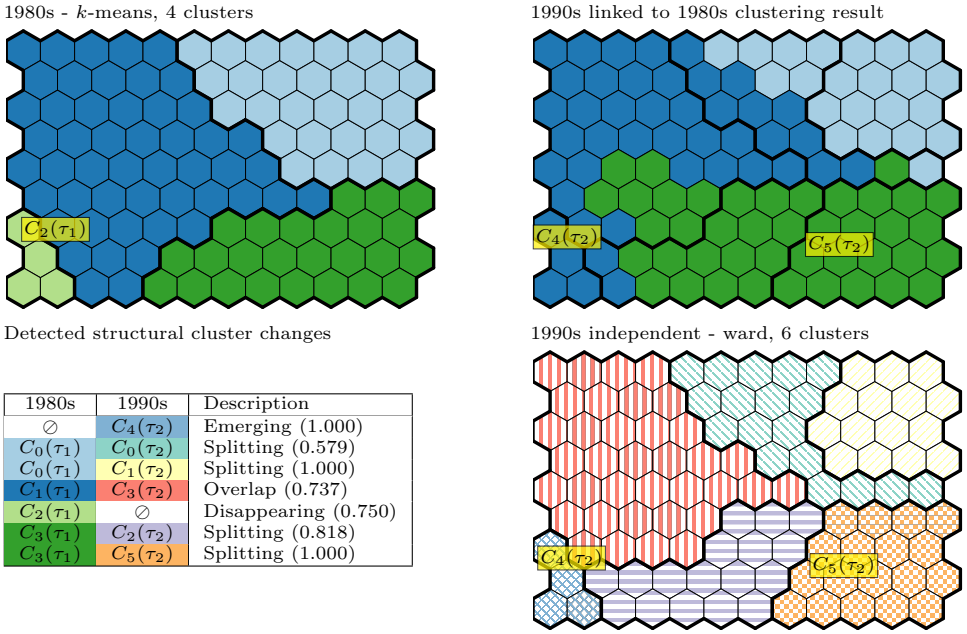
such as *population*, *life expectancy*, *mortality rate*, *immunization*, *illiteracy rate*, *education*, *television ownership*, and *inflation* (Denny & Squire 2005). The annual values are grouped into 10-years value by taking the latest value available in the period. The 1980s data is used as period τ_1 and the 1990s as period τ_2.

The analysis of cluster changes starts by selecting the clustering result of the datasets $\mathcal{D}(\tau_1)$ and $\mathcal{D}(\tau_2)$. Cluster validity indexes, such as Davies-Bouldin Index and Silhouette Index, or dendrogram tree were used to guide this selection. The Davies-Bouldin index of the k-means clustering results (Figure 6) shows that the optimal clustering result is four clusters for the 1980s. Based on the dendrogram of Ward linkage, the optimal clustering result is six clusters for the 1990s.

The table (*bottom left* in Figure 5(b)) shows the detected structural cluster changes. The cells' back-



(a) The ReDSOM visualization of the 1980s map (*left*) identifies cluster $C_2(\tau_1)$ is disappearing in the 1990s map. The ReDSOM visualization of the 1990s map (*right*) identifies a new cluster $C_4(\tau_2)$ and a new region $C_5(\tau_2)$ compared to the 1980s map.



1980s	1990s	Description
⊘	$C_4(\tau_2)$	Emerging (1.000)
$C_0(\tau_1)$	$C_0(\tau_2)$	Splitting (0.579)
$C_0(\tau_1)$	$C_1(\tau_2)$	Splitting (1.000)
$C_1(\tau_1)$	$C_3(\tau_2)$	Overlap (0.737)
$C_2(\tau_1)$	⊘	Disappearing (0.750)
$C_3(\tau_1)$	$C_2(\tau_2)$	Splitting (0.818)
$C_3(\tau_1)$	$C_5(\tau_2)$	Splitting (1.000)

(b) The clustering result of the 1980s map is shown using colour on both the 1980s map (*left*) and the 1990s map (*right*). The independent clustering result of the 1990s map is shown using thick outlines and patterns on the 1990s map (*right*). The cluster $C_2(\tau_1)$ on the 1980s map no longer exists in the 1990s map. A new cluster $C_4(\tau_2)$ emerged on the 1990s map.

Figure 5: ReDSOM, clustering result visualizations, and detected structural cluster changes of the world's welfare and poverty maps of the 1980s (*left*) and the 1990s (*right*).

ground colour refers to the colour of clusters used in the figure. The rd1_j visualization (*left* of Figure 5(a)) shows that cluster $C_2(\tau_1)$ disappears in the 1990s. This change is confirmed by the cluster colour linking *top right* visualization shown in Figure 5(b). This disappearing cluster is detected in the table. This cluster consists of four Latin American countries: Brazil, Argentina, Nicaragua, and Peru who experienced a debt crisis in the 1980s, which is known as the 'lost decade'. However, many Latin American countries undertook rapid reforms in the late 1980s and early 1990s. The algorithm also detected two clusters, $C_0(\tau_1)$ and $C_3(\tau_1)$, in the 1980s that were split in the 1990s. The cluster $C_4(\tau_2)$ emerged in the 1990s, consisting of China and India, which were characterized by high total labour forces.

7 Conclusion and Future Work

We have presented an algorithm to automatically detect structural changes between two clustering results obtained from two snapshot datasets. Based the evaluations, the algorithm can detect various structural changes. As a SOM produces a considerably smaller-sized set of prototype vectors, it allows an efficient use of two-level clustering and this automatic detection. The method presented here can be extended further to detect cluster changes from multiple snapshots.

References

Adomavicius, G. & Bockstedt, J. (2008), 'C-TREND: Temporal cluster graphs for identifying and visualizing trends in multiattribute transactional data', *IEEE Transaction on Knowledge and Data Engineering* **20**(6), 721–735.

Aggarwal, C. C. (2005), 'On change diagnosis in evolving data streams', *IEEE Transactions on Knowledge and Data Engineering* **17**, 587–600.

Antunes, C. M. & Oliveira, A. L. (2001), Temporal data mining: An overview, *in* 'KDD 2001 Workshop on Temporal Data Mining', pp. 1–13.

Bartolini, I., Ciaccia, P., Ntoutsi, I., Patella, M. & Theodoridis, Y. (2009), 'The panda framework for

comparing patterns', *Data and Knowledge Engineering* **68**(2), 244–260.

Boettcher, M. (2011), 'Contrast and change mining', *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **1**(3), 215–230.

Denny & Squire, D. M. (2005), Visualization of cluster changes by comparing self-organizing maps, *in* 'Advances in Knowledge Discovery and Data Mining, PAKDD 2005, Proceedings', Vol. 3518 of *LNCS*, Springer, pp. 410–419.

Denny, Williams, G. J. & Christen, P. (2010), 'Visualizing Temporal Cluster Changes using Relative Density Self-Organizing Maps', *KAIS* **25**(2), 281–302.

Diggle, P. J., Liang, K.-Y. & Zeger, S. L. (1994), *Analysis of Longitudinal Data*, Oxford University Press, New York.

Fukui, K.-I., Saito, K., Kimura, M. & Numao, M. (2008), Sequence-based som: Visualizing transition of dynamic clusters, *in* 'Computer and Information Technology (CIT) 2008. 8th IEEE International Conference on', pp. 47–52.

Ganti, V., Gehrke, J., Ramakrishnan, R. & Loh, W.-Y. (2002), 'A framework for measuring differences in data characteristics', *Journal of Computer and System Sciences* **64**(3), 542–578.

Han, J., Kamber, M. & Pei, J. (2011), *Data Mining: Concepts and Techniques (third edition)*, Morgan Kaufmann.

Held, P. & Kruse, R. (2013), Analysis and visualization of dynamic clusterings, *in* 'HICSS', IEEE, pp. 1385–1393.

Hinneburg, A. & Keim, D. A. (2003), 'A general approach to clustering in large databases with noise', *KAIS* **5**, 387–415.

Jain, A. K., Murty, M. N. & Flynn, P. J. (1999), 'Data clustering: a review', *ACM Computing Survey* **31**(3), 264–323.

Kalnis, P., Mamoulis, N. & Bakiras, S. (2005), On discovering moving clusters in spatio-temporal data., *in* 'SSTD 2005, Proceedings', Vol. 3633 of *LNCS*, Springer, pp. 364–381.

Macqueen, J. B. (1967), Some methods of classification and analysis of multivariate observations, *in* 'Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability', University of California Press, pp. 281–297.

Ntoutsi, I., Spiliopoulou, M. & Theodoridis, Y. (2009), 'Tracing cluster transitions for different cluster types', *Control and Cybernetics* **38**(1), 239–259.

Oliveira, M. & Gama, J. a. (2010), Bipartite graphs for monitoring clusters transitions, *in* 'Advances in Intelligent Data Analysis IX', Vol. 6065 of *LNCS*, Springer Berlin / Heidelberg, pp. 114–124.

Parzen, E. (1962), 'On estimation of a probability density function and mode', *The Annals of Mathematical Statistics* **33**(3), 1065–1076.

Roddick, J. F., Hornsby, K. & Spiliopoulou, M. (2001), An updated bibliography of temporal, spatial, and spatio-temporal data mining research, *in* 'Temporal, Spatial, and Spatio-Temporal Data Mining', Vol. 2007 of *LNCS*, pp. 147–163.

Roddick, J. F. & Spiliopoulou, M. (2002), 'A survey of temporal knowledge discovery paradigms and methods', *IEEE TKDE* **14**(4), 750–767.

Rosenblatt, M. (1956), 'Remarks on some nonparametric estimates of a density function', *Annals of Mathematical Statistics* **27**(3), 832–837.

Silverman, B. (1986), *Density estimation for statistics and data analysis*, Chapman and Hall Ltd, London, New York.

Skupin, A. & Hagelman, R. (2005), 'Visualizing demographic trajectories with Self-Organizing Maps', *GeoInformatica* **9**, 159–179.

Spiliopoulou, M., Ntoutsi, I., Theodoridis, Y. & Schult, R. (2006), Monic: modeling and monitoring cluster transitions, *in* 'KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining', ACM Press, New York, NY, USA, pp. 706–711.

Vesanto, J. & Alhoniemi, E. (2000), 'Clustering of the Self-Organizing Map', *IEEE Transactions on Neural Networks* **11**(3), 586–600.

World Bank (2003), *World Development Indicators 2003*, The World Bank, Washington DC.

Market Segmentation of EFTPOS Retailers

Ashishkumar Singh **Grace Rumantir** **Annie South**
Faculty of Information Technology Caulfield
Princes Highway, Caulfield East
Victoria 3145
{ashish.singh, grace.rumantir}@monash.edu

Australia New Zealand Bank
Level 9, 833 Collins Street,
Docklands, Victoria 3008
annie.south@anz.com

Abstract

Almost all the papers on market segmentation modeling using retail transaction data reported in the literatures deal with finding groupings of customers. This paper proposes the application of clustering techniques on finding groupings of retailers who use the Electronic Funds Transfer at Point Of Sale (EFTPOS) facilities of a major bank in Australia in their businesses. The RFM (Recency, Frequency, Monetary) analysis on each retailer is used to reduce the large data set of customer purchases through the EFTPOS network into attributes that may explain the business activities of the retailers. Preliminary results show that groupings of retailers with distinct combinations of RFM values can be established which encourage further modeling using other business and demographic attributes on bigger data set over a longer period of time.

Keywords: Market Segmentation, Clustering, RFM Analysis, EFTPOS.

1 Introduction

Electronic Funds Transfer at Point Of Sale (EFTPOS) is one of the leading methods of payment at checkouts or Point Of Sale (POS). Payments on EFTPOS terminals are done using debit and credit cards, the most common non-cash payment methods. The banking sector gains profits by the renting out EFTPOS machines to retailers through set up fee, periodic service fee and transaction fee on each purchase put through the EFTPOS machine. Banks also profit through the availability of interest free fund en-route to the designated retailer's bank account after being debited from the payee's account and the availability of the deposited fund into the retailer's account itself. In order for the banking sector to develop this part of the business further, it can make use of market segmentation modeling to gain better understanding on the business behaviours of the retailers using the EFTPOS facilities.

The concept of market segmentation was first introduced in Smith (1956) and defined as the "process of subdividing a market into distinct subsets of customers that behave in the same way or have similar needs. Each subset may conceivably be chosen as a market target to be reached with a distinctive marketing strategy" (Doyle 2011). The heart of any good market segmentation tool is its ability to analyse, understand and draw good market segments based on customer's purchasing behaviours. Whilst market segmentation has been extensively applied on transaction data to find insights into customer purchasing behaviours on the market, very limited work has been done to find in-sights into retailer business activities. This paper reports on the use of clustering techniques on EFTPOS transaction data to identify segments of retailers who have certain common characteristics.

The selection of attributes plays an important role in good clustering analysis. This paper uses the RFM (Recency, Frequency, Monetary) analysis, popular in marketing, to reduce the data set for the clustering experiments.

The paper is organized as follows: Section 2 gives a summary of the review of the literatures on market segmentation papers for the past 10 years; Section 3 explains the data reduction/transformation using the RFM Analysis; Section 4 briefly explains the clustering techniques used; Section 5 outlines and discusses the results of the experiments; Section 6 concludes the paper and explains our plans for the future.

2 Related Work

Table 1 summaries our review of the literatures on related work in market segmentation with respect to the attribute selection techniques and clustering techniques employed. All of the papers reviewed report the use of transaction data as input and segmentation experiments on customers. Only one paper i.e. (Bizhani & Tarokh 2011), reports segmentation experiments on retailers using EFTPOS data. This is the only work on EFTPOS data we have found in the literatures. The work reported in Bizhani & Tarokh (2011) is on a much smaller data set and considers individual EFTPOS machines as "retailers". The data set we have been acquiring for our work falls in the category of Big Data with each merchant/retailer having multiple EFTPOS machines. Our experience in acquiring, secured-storing and processing the commercial in confidence EFTPOS data has been reported in Singh, A., et al. (2014).

As shown in **Table 1** socio-demographic analysis and RFM analysis are the two most popular attribute selection techniques for market segmentation. For example, in Y. Kim et al. (2005) and J. H. Lee & Park (2005), attributes derived from socio-demographic characteristics (e.g.

average age of residents, proportion of residents in high status and others) of customer have been reported as effective in customer segmentation. Whereas in Olson et al. (2009), attributes based on RFM comparatively yield

Table 1. Summary of review of the literatures on market segmentation in the past 10 year

Related Work	Attribute Selection Techniques	Clustering Techniques			
	Socio-demographic Analysis	RFM Analysis	K-means	Hierarchical clustering	Other
(Y.-S. Chen et al. 2012) (Lefait & Kechadi 2010)		X	X		
(Hsieh 2004)		X			X Neural Network
(Bizhani & Tarokh 2011)		X	X		X Unsupervised Learning Vector Quantization
(D. Chen et al. 2012)		X	X		X Decision Tree
(Olson et al. 2009)		X			X
(Namvar et al. 2010)	X	X	X		
(Y. Kim et al. 2005) (J. H. Lee & Park 2005)	X				X Neural Networks
(Dennis et al. 2003)	X				X
(Ho et al. 2012)			X Genetic Algorithms		
(Salvador & Chan 2004)			X	X	
(D. Gaur & S. Gaur 2013)			X	X	X
(Zakrzewska & Murlewski 2005)			X	X	X Density based Clustering
(Alam et al. 2010) (Yoon et al. 2013)				X	
(Li et al. 2009)				X Chameleon	
(Suib & Deris 2008)				X Hierarchical Pattern Based Clustering	

better clustering results and can effectively handle a large degree of multi-dimensional data. With respect to clustering algorithms used, K-means and hierarchical clustering are the most popular techniques for market segmentation.

3 RFM Analysis

EFTPOS transaction data are being collected from one of the four major banks in Australia. This paper reports on the preliminary stage of our project where we use data for a total period of 18 days, starting from 19-September-2013 to 07-October-2013. Each transaction record has 55 attributes.

The 18 daily data files contain approximately 77.5 million transaction records from over 1 million unique retailers. This high volume of data makes even the basis operations, such as finding the total monetary amount of each retailer in the data set, very time consuming and resource intensive. To overcome this Big Data problem, a hash table for the retailer ids is used.

Normalisation is very important in this clustering project as clustering algorithms use various distance measures between attribute values. An attribute with values covering a large range, like retailer's Monetary values, may dominate over another attribute with smaller range of values, like retailer's Recency and Frequency values. To alleviate this problem, in this project, the min-max normalization technique is used where the values of all three attributes are normalised into a similar range between 0 and 100.

The RFM (Recency, Frequency, Monetary) analysis was proposed in Hughes (2006). This paper proposes the use of these three attributes to group retailers exhibiting similar business activities:

Recency - the Recency value for a retailer is the time interval between a global datum and his/her latest transaction. A retailer with a smaller Recency value is seen as more current in his/her business activities than a retailer with a bigger Recency value. The global datum is chosen as midnight after the last transaction day in the data set (i.e. the midnight of 8th October 2013 (00:00:000000 in HH:MM:SSSSSS format). Hence, all the Recency values are calculated from midnight of 8th October 2013 backwards.

Frequency - the total number of transactions put through all of the EFTPOS terminals belonging to a retailer forms the Frequency value of the retailer.

Monetary value - the total amount of transactions put through all of the EFTPOS terminals belonging a retailer forms the Monetary value of the retailer.

Figure 1 shows the histogram of the distribution of the Recency, Frequency and Monetary values of all of the retailers in the data set. The histogram shows that the Frequency and Monetary values are skewed to the lower end of the values.

The histogram in **Figure 1** is open ended at the top end because there are small numbers of very large values of the three attributes in the data set. This is shown more clearly in the histograms in **Figure 2** where the horizontal axis of each histogram is intentionally "squeezed" in the middle to show these extreme values and also because there are very few data in this range (no data for Frequency and Monetary values).

Figure 2 also shows the correlations amongst the three attributes. There is positive correlation between Frequency and Monetary values and no correlation between Recency with the other two attributes. This implies that retailers that generate a lot of transactions tend to accumulate large Monetary values within the observation time period.

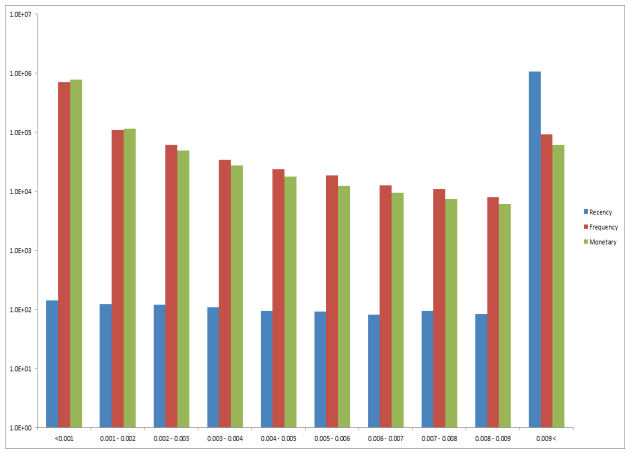


Figure 1. The distribution of Recency, Frequency and Monetary values of all retailers

4 Cluster Analysis

Being the most popular methods used in the literatures for market segmentation, we use K-means and Agglomerative Hierarchical Clustering (AHC) in this preliminary work. Clustering analysis on a large data set is time consuming and processor intensive. with 16 cores and 32 GB RAM is employed. For this work, parallelisation in the form of multi-threading using Intel Xeon and AMD Opteron CPUs of various clock speeds. Our experience in acquiring, secured-storing and processing the commercial in confindence EFTPOS data is reported in Ashish, et al. (2014).

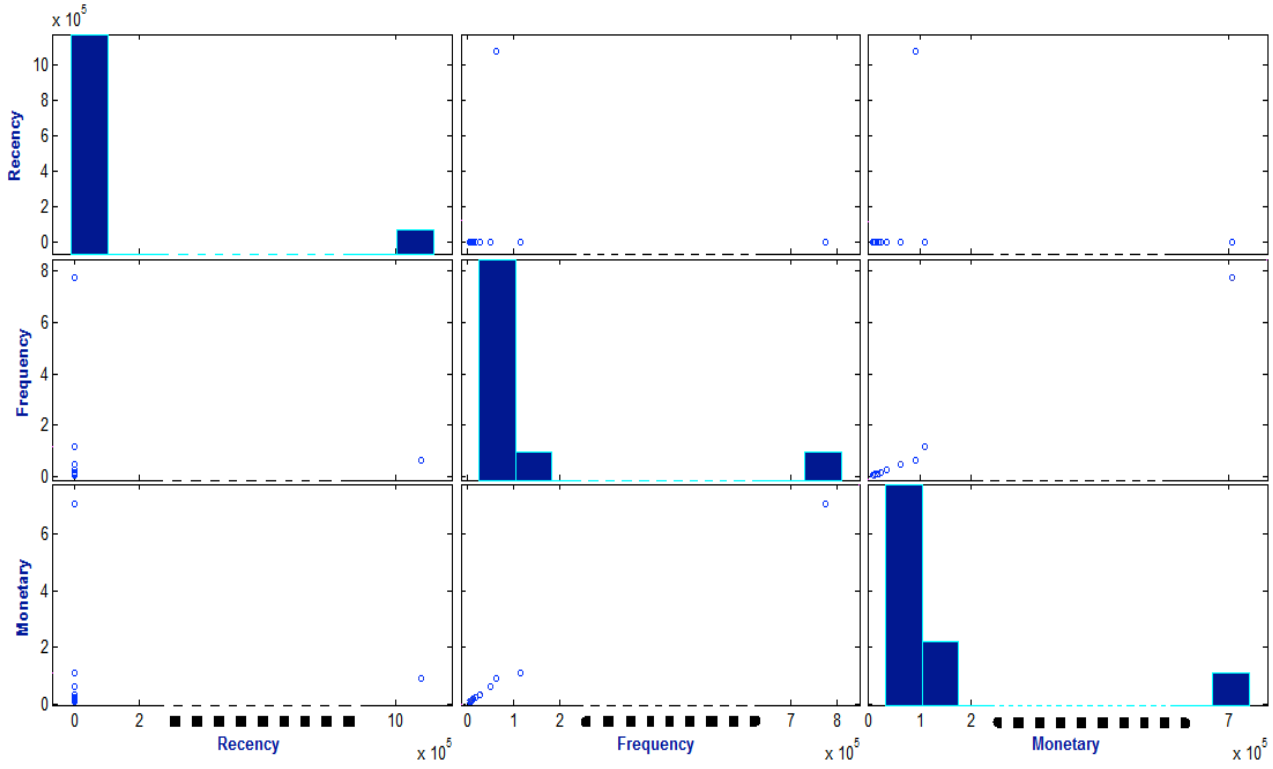


Figure 2. The correlations amongst the three attributes. The middle of each of the x-axis are deliberately truncated to show the small number of extreme values at the upper end, made possible as there are no data Frequency and Monetary values and very few Recency values in the middle range

For the K-means clustering, the calculation of the Euclidean Distance between each data point to the centroid of a cluster is done in parallel over subsets of the data set. For the AHC, the calculation of the Ward Minimum Variance to evaluate the inter-cluster distance measure (starting with clusters each with 1 member data point) and the creation of the clusters through agglomerative process are done on 60% of the data set with the remaining 40% are allocated to the clusters based on Euclidean distance.

In this project, we create clustering models with 2 to 25 clusters using both K-means and AHC.

Both sets of clustering models are then tested based on Dunn's Index (Dunn† 1974) and Davis-Bouldin Index (Davies & Bouldin 1979) to find the number of clusters which result in high intra-cluster similarity and high inter-cluster dissimilarity.

Dunns Index is given as:

$$D = \min_{1 \le i \le c} \left\{ \min_{1 \le j \le c, j \ne i} \left\{ \frac{\delta(C_i, C_j)}{\max_{1 \le k \le c} \{\Delta(C_k)\}} \right\} \right\}$$

where: $\delta(C_i, C_j)$ is the inter-cluster distance between clusters C_i and C_j; $\Delta(C_k)$ is the intra-cluster distance of cluster C_k; c is the number of clusters.

Also Davis-Bouldin Index is given as:

$$DB = \frac{1}{c} \sum_{i=1}^{c} \max_{i \ne j} \left\{ \frac{\Delta(C_i) + \Delta(C_j)}{\delta(C_i, C_j)} \right\}$$

where: $\delta(C_i, C_j)$ is the inter-cluster distance between two clusters C_i and C_j; $\Delta(C_i)$ and $\Delta(C_j)$ are the intra-cluster distances of clusters C_i and C_j respectively; c is the number of clusters.

Two inter-cluster distance measures and two intra-cluster distance measures are used to evaluate these two indices as outlined in **Table 2**.

Table 2. Two pairs of inter-cluster and intra-cluster distance measures to be used in Dunn's and Davies Bouldin Indices.

	Average linkage distance measure	Centroid linkage distance measure
Intercuster distance measure	$\Delta(C) = \frac{1}{\lvert C\rvert \times (\lvert C\rvert - 1)} \sum_{x \in C,\, y \in C,\, x \ne y} d(x,y)$	$\Delta(C) = 2\left(\frac{\sum_{x \in C} d(x, \bar{C})}{\lvert C\rvert}\right)$ where $\bar{C} = \frac{1}{\lvert C\rvert} \sum_{x \in C} x$
	Average diameter distance measure	**Centroid diameter distance measure**
Intracluster distance measure	$\delta(C_X, C_Y) = \frac{1}{\lvert C_X\rvert \lvert C_Y\rvert} \sum_{x \in C_X,\, y \in C_Y} d(x,y)$	$\delta(C_X, C_Y) = d\left(\frac{1}{\lvert C_X\rvert} \sum_{x \in C_X} x, \frac{1}{\lvert C_Y\rvert} \sum_{x \in C_Y} y\right)$

5 Results and Discussion

The Dunns and Davies–Bouldin indices for each of the 24 clustering models (with 2 to 25 clusters respectively) created using each of the two clustering techniques are plotted in **Figure 3** and **Figure 4** respectively. Comparing the results between the K-means and AHC models, we can see that the AHC models with 16 and 17 clusters have higher Dunn's indices than any of the K-means models. The K-means models with 13 clusters or more tend have higher Davies–Bouldin indices than the AHC models with the same number of clusters. It is then decided to analyse the AHC model with 19 clusters with a view that manual merging of clusters might be warranted post-analysis.

Table 3 shows the AHC clusters, each with the centroid represented by the average RFM values. Each of the R, F, M values in the data set is divided into 5 quantiles, with labels 1 to 5. Based on the centroid values, each cluster is assigned a 3 digit label which values depend on which range each of its R, F, M values falls in.

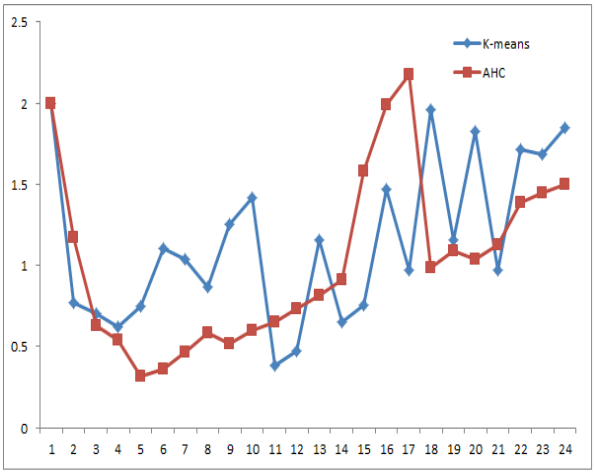


Figure 3. A plot of Dunns indices for each of the number of clusters (x-axis) in the models created using K-means and Agglomerative Hierarchical Clustering techniques

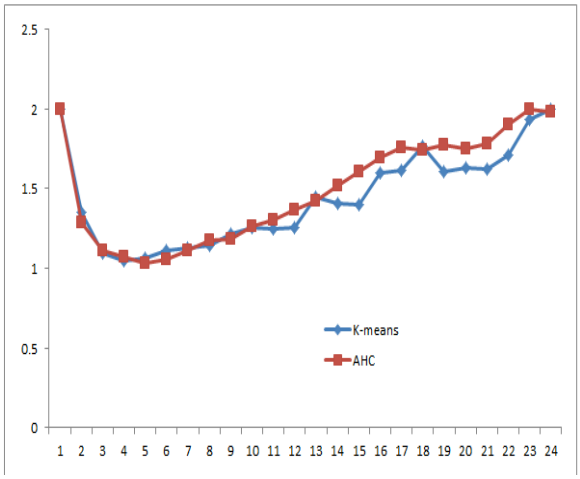


Figure 4. A plot of Davies–Bouldin indices for each of the number of clusters (x-axis) in the models created using K-means and Agglomerative Hierarchical Clustering techniques

Low value in Recency suggests the retailer has recent transactions. Hence, low value in Recency and very high values in both Frequency and Monetary suggest the feature of a retailer with active, successful business. In **Table 3**, retailers in Clusters 1, 2, 18 (constitute 34.23%

of the total retailers) exhibit such characteristics. The bank may wish to provide incentives for these retailers to maintain their excellent business performances.

Retailers characterized by moderately high Recency value with average levels of Frequency and Monetary values are seen in Clusters 5, 6, 13, 14, 15. These retailers have businesses with moderate level of activities but have recently been inactive for some time. The bank may want to engage these retailers into more active participation in generating EFTPOS transactions as this will consequently boost their Frequency and Monetary values.

Table 3. Clustering Results

Cluster #	%	Average Recency	Average Frequency	Average Monetary Value	R,F,M labels (1 to 5 quantiles)
1	**10.30**	**2.368704**	**0.004604**	**0.003177**	**255**
2	**15.27**	**1.018574**	**0.038917**	**0.017611**	**155**
3	2.41	67.028300	0.000117	0.000333	523
4	2.30	71.897224	0.000105	0.000313	523
5	1.72	47.020206	0.000343	0.000331	433
6	3.20	42.105650	0.000293	0.000681	434
7	1.31	82.475670	0.000085	0.000163	522
8	2.14	76.985930	0.000071	0.000287	523
9	2.11	97.306885	0.000025	0.000220	523
10	2.32	92.311350	0.000067	0.000303	523
11	3.25	57.261173	0.000191	0.000473	523
12	2.79	62.190346	0.000169	0.000406	523
13	5.12	27.091051	0.000393	0.000818	434
14	4.46	32.134422	0.000318	0.000703	434
15	7.36	21.853570	0.000626	0.001149	434
16	11.78	17.150494	0.001816	0.002548	345
17	2.39	52.247010	0.000147	0.000263	523
18	**8.66**	**6.925114**	**0.011165**	**0.004900**	**255**
19	11.12	12.214928	0.002484	0.002045	245

Retailers characterized by very high Recency value with average levels of Frequency and Monetary values are seen in Clusters 3, 4, 7 to 12 and 17. These retailers have been inactive in the EFTPOS network for a long period of time and/or have low level of business activities overall. They constitute 21% of the EFTPOS market for the bank. The bank can see this type of retailers as growth area in the EFTPOS business sector. It however needs to determine if it is cost effective to launch aggressive marketing strategy to help improve the performance of these retailers. If it is decided that no aggressive campaigning is to be done for these retailers, due to the size of this market, some kind of marketing campaign will nevertheless still be required to maintain them.

6 Conclusion and Future Work

This paper proposes the use of clustering techniques to group retailers on an EFTPOS network based on the similarities in their business activities as characterized by how recent their business activities are, how frequent they conduct their business on the EFTPOS network and how much money their business activities have generated over a period of time. The preliminary results show that there are distinct combinations of RFM values of retailers in the clusters that may give the bank indications of the different marketing strategies that can be applied to each of the retailer types.

Further analysis into each cluster to find out more on the characteristics of the business and background of the retailers will help the bank in fine tuning their target marketing strategy for each retailer type. The next step of this project will be broken down into 3 categories. First, in observing if there are latent variables in the data set that may influence the variations in the volume of transactions in different days and possibly different periods in a day. Second, in trying out other clustering techniques to see of better quality clusters can be formed. Third, in building classification or causal models to find explanatory rules on the characteristics of each cluster using other attributes in the data set (e.g. the line of business a retailer is in, the business premise, etc.) and in exogenous variables like socio-demographic, advertising, social media data.

The data set used in this preliminary work is just from the first 2 weeks of our EFTPOS data extraction. We have since collected a few months of EFTPOS transaction data which will allow us to conduct more extensive Big Data analysis with more convincing results as a consequence. This will also open up new research avenues into the kinds of suitable big data computing techniques for market segmentation projects involving MapReduce/Hadoop system that we have put in place for this project.

7 Acknowledgments

The authors wish to thank Jon Scheele and Jo Spencer for their help in data acquisition and Blair Bethwaite for his help in setting up the secure computing environment to enable the execution of this commercial in confidence Big Data project.

8 References

Alam, S. et al., 2010. Particle swarm optimization based hierarchical agglomerative clustering. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*. pp. 64–68.

Bizhani, M. & Tarokh, M.J., 2011. Behavioral rules of bank's point-of-sale for segments description and scoring prediction. *Int. J. Industrial Eng. Comput*, 2, pp.337–350.

Chen, D., Sain, S.L. & Guo, K., 2012. Data mining for the online retail industry: A case study of RFM model-based customer segmentation using data mining. *Journal of Database Marketing \& Customer Strategy Management*, 19(3), pp.197–208.

Chen, Y.-S. et al., 2012. Identifying patients in target customer segments using a two-stage clustering-classification approach: A hospital-based assessment. *Computers in Biology and Medicine*, 42(2), pp.213–221.

Davies, D.L. & Bouldin, D.W., 1979. A cluster separation measure. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (2), pp.224–227.

Dennis, C. et al., 2003. Market segmentation and customer knowledge for shopping centers. In *Information Technology Interfaces, 2003. ITI 2003. Proceedings of the 25th International Conference on*. pp. 417–424.

Doyle, C., 2011. *A dictionary of marketing*, Oxford University Press.

Dunn†, J.C., 1974. Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1), pp.95–104.

Gaur, D. & Gaur, S., 2013. Comprehensive Analysis of Data Clustering Algorithms. In *Future Information Communication Technology and Applications*. Springer, pp. 753–762.

Ho, G.T. et al., 2012. Customer grouping for better resources allocation using GA based clustering technique. *Expert Systems with Applications*, 39(2), pp.1979–1987.

Hsieh, N.-C., 2004. An integrated data mining and behavioral scoring model for analyzing bank customers. *Expert Systems with Applications*, 27(4), pp.623–633.

Hughes, A.M., 2006. *Strategic database marketing*, McGraw-Hill.

Kim, Y. et al., 2005. Customer targeting: A neural network approach guided by genetic algorithms. *Management Science*, 51(2), pp.264–276.

Lee, J.H. & Park, S.C., 2005. Intelligent profitable customers segmentation system based on business intelligence tools. *Expert Systems with Applications*, 29(1), pp.145–152.

Lefait, G. & Kechadi, T., 2010. Customer Segmentation Architecture Based on Clustering Techniques. In *Digital Society, 2010. ICDS'10. Fourth International Conference on*. pp. 243–248.

Li, J., Wang, K. & Xu, L., 2009. Chameleon based on clustering feature tree and its application in customer segmentation. *Annals of Operations Research*, 168(1), pp.225–245.

Namvar, M., Gholamian, M.R. & KhakAbi, S., 2010. A two phase clustering method for intelligent customer segmentation. In *Intelligent Systems, Modelling and Simulation (ISMS), 2010 International Conference on*. pp. 215–219.

Olson, D.L. et al., 2009. Comparison of customer response models. *Service Business*, 3(2), pp.117–130.

Salvador, S. & Chan, P., 2004. Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In *Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on*. pp. 576–584.

Singh, A., et al., Clustering Experiments on Big Transaction Data for Market Segmentation, in the Proceedings of the BigDataScience '14 Conference, August 04 - 07 2014, Beijing, China, ACM 978-1-4503-2891-3/14/08, http://dx.doi.org/10.1145/2640087.2644161 (in press)

Smith, W.R., 1956. Product differentiation and market segmentation as alternative marketing strategies. *The Journal of Marketing*, 21(1), pp.3–8.

Suib, D.S. & Deris, M.M., 2008. An efficient hierarchical clustering model for grouping web transactions. *International Journal of Business Intelligence and Data Mining*, 3(2), pp.147–157.

Yoon, S.-H. et al., 2013. A data partitioning approach for hierarchical clustering. In *Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication*. p. 72.

Zakrzewska, D. & Murlewski, J., 2005. Clustering algorithms for bank customer segmentation. In *Intelligent Systems Design and Applications, 2005. ISDA'05. Proceedings. 5th International Conference on*. pp. 197–202.

Hartigan's Method for K-modes Clustering and Its Advantages

Zhengrong Xiang[1] **Md Zahidul Islam**[2]

[1] College of Computer Science, Zhejiang University, China
Email: zolaxiang@gmail.com

[2] School of Computing and Mathematics, Charles Sturt University, Australia
Email: zislam@csu.edu.au

Abstract

Recently it has been shown that for k-means, Hartigan's method has better optimization performance than the prevalent Lloyd's method. Hartigan's method is a general idea of optimization heuristic. When considering moving a point to another cluster, it measures the exact change to the objective function. In this paper we develop a Hartigan's method for another important clustering objective: k-modes, which is popularly used for categorical data. The proposed algorithm is as efficient as the Lloyd's method on k-modes. Moreover, we rigorously prove that Hartigan's method can further improve some local optima achieved by Lloyd's method. Empirical evaluation verifies this conclusion. Furthermore, when these two methods are used independently, Hartigan's method also achieves better optimization performance.

Keywords: Cluster Analysis, K-modes, Lloyd's method, Hartigan's method

1 Introduction

Partitioning relocation clustering (Kaufman 2009, Everitt 2011) is an important and popular scheme for cluster analysis, typical example being k-means (Steinley 2006). K-modes (Huang 1998, Chaturvedi 2001) is a similar method as k-means, which is used specifically for categorical data. It defines the cluster center to be the "mode" of the attribute values of all records/objects in the cluster. The goal is also to minimize the sum of distances from objects to the respective modes. Like k-means, k-modes has a clear clustering objective, and it's efficient to achieve local optima. Since it was proposed, there has been a lot of related research, and has appeared in commercial products (Daylight 2010).

The optimization methods for k-means and k-modes are the same: iteratively assign objects to the nearest center (mean/mode), and update the centers. This method is called Lloyd's method (Lloyd 1982, MacQueen 1967). For k-means, there is actually a less well known method called Hartigan's method (Hartigan 1975, 1979). It proceeds one point at a time. For each point, move it to another cluster if the k-means objective is improved. Iteratively scan the data set until no more points can be moved. For Euclidean distance, there is a closed-form expression for deciding whether or not a move is profitable, so that Hartigan's method is as efficient as Lloyd's method.

Recently it has been shown that for k-means, Hartigan's method achieves better optimization performance than Lloyd's method (Telgarsky 2010). Specifically, the data partitions reached by Hartigan's method are tighter than the Voronoi partitions reached by Lloyd's method. They also discussed the characteristics of situations where Hartigan's method can improve on the final partitions of Lloyd's method. Empirical results support the above statements.

Lately another piece of research (Slonim 2013) further advocates the advantages of Hartigan's method over Lloyd's method for k-means. It developed a closed-form expression for any Bregman divergence, which Euclidean distance is an example of. Moreover, some types of problems are provided where Lloyd's method can not make any improvements while Hartigan's method can correctly converge.

In this paper, we focus on a similar issue concerning the quality of optimization algorithms, while we change the subject to k-modes. First note that Hartigan's method is a general idea of optimization heuristic. Unlike k-means, there is no previous research about how to use this idea on k-modes. The first part of our work develops an instantiation of Hartigan's method on k-modes, which has never been done to the best of our knowledge. The computational complexity of the proposed application of Hartigan's method on k-modes is as good as Lloyd's method on k-modes.

Secondly, we rigorously prove that when Lloyd's method gets stuck at a local optimum, sometimes it can be further optimized by Hartigan's method. Specifically, we characterize the kind of situations when the improvement is possible, and when it appears more often. We also prove that it's strictly not possible for Lloyd's method to improve upon Hartigan's method.

In the empirical evaluations, both synthetic and real-world data verify the above conclusion. Moreover, an interesting discovery is that Hartigan's method also outperforms Lloyd's method when the two methods are used independently. Finally, experiments also show that Hartigan's method requires less computations than Lloyd's method when the number of clusters and the number of features are large.

The second author would like to thank the Faculty of Business COMPACT Fund R4 P55 in Charles Sturt University, Australia.

2 Hartigan's Method For K-modes

2.1 K-modes and Lloyd's Method

First we briefly present the original k-modes. Suppose we have a data set including N objects X_i, $i = [1, N]$. The number of features/attributes is D, and we want K clusters: C_k, $k = [1, K]$. The objective function of k-modes is:

$$F = \sum_{k=1}^{K} \sum_{X_i \in C_k} d(X_i, M_k) \tag{1}$$

In the objective, M_k is the mode of cluster C_k, $d(X, Y)$ refers to the distance measure between two categorical objects. In k-modes, it's the simple matching distance (let x_j be the jth feature of X):

$$d(X, Y) = \sum_{j=1}^{D} d(x_j, y_j), \;\; d(x_j, y_j) = \begin{cases} 0; & if \; x_j = y_j \\ 1; & otherwise \end{cases} \tag{2}$$

The mode takes the most frequent value for each attribute, in order to reach minima. For example, if *red* is the most dominant value for the Color attribute, the mode takes *red* for this attribute.

Lloyd's method for k-modes is just like that of k-means: iteratively assign objects to the nearest mode, and update the modes. When no objects are reassigned, the algorithm converges to a local optimum.

2.2 Hartigan's Method

In this section, we propose Hartigan's method for k-modes, and show its efficiency. While Hartigan's method was used for k-means (Hartigan 1975, Telgarsky 2010), it has never been used in the past for k-modes.

Hartigan's method is a greedy heuristic for optimizing clustering objectives. Single data point is relocated to another cluster, if this move improves the objective (in our scenario, the value of the objective function decreases). Proceed until no points can be relocated, and the algorithm reaches to a local optimum.

To formalize: suppose an object t is under consideration. t is originally in cluster P. Moving t to another cluster Q has the following consequences: cluster P lost an object, the value of objective function for P also changes into F_{P-t}; cluster Q added an object, the objective becomes F_{Q+t}. If this relocation improves the total objective, then:

$$F_{P-t} + F_{Q+t} < F_P + F_Q \tag{3}$$

Rewritten as

$$\Delta F = (F_{P-t} - F_P) + (F_{Q+t} - F_Q) < 0 \tag{4}$$

Like in the work(Xiang 2013), we call this *transfer test*. For one object, there can be multiple destination clusters that satisfy the transfer test. Usually, picking the cluster that maximizes ΔF gives better overall performance (Tarsitano 2003).

Now we propose a technique to efficiently optimize the objective of k-modes with the basic idea of Hartigan's method. At the first sight, computing the transfer test requires computing values of the objective function, which is computationally expensive ($O(N^2)$). However, it can be carried out quite efficiently by computing only the changes: $F_{P-t} - F_P$, $F_{Q+t} - F_Q$.

Table 1: A Particular Frequency Table

Attributes	f_1		f_2		
Attribute Values	a	b	x	y	z
Frequency	7	3	4	4	2

Table 2: Computation of the Transfer Test

Departure Value	$F_{P-t} - F_P$
Not-Mode	-1
Is-Mode:	0
Is-Mode and Exists-Duplicated-Mode	-1

Arrival Value	$F_{Q+t} - F_Q$
Is-Mode	0
Not-Mode	1
Not-Mode and Is-Duplicated-Mode	0

For every cluster we need to keep a *frequency table* to record the frequencies of all attribute values. The table is stored in memory , and every entry should be accessed in $O(1)$. For example, say a cluster has 10 objects, with 2 features f_1 and f_2, then the table might be like the one shown in Table 1. Note that the mode is (a, x) or (a, y).

Because the objective is summed over different attributes independently, we only need to know how to compute the transfer test at a single attribute. First look at the change at the original cluster: $F_{P-t} - F_P$.

One possibility is that the value of object t is different from the value of mode. Say it's the value b in attribute f_1. The departure of t wouldn't change the mode, because the mode value is still the most frequent one. F_{P-t}, which is the sum of distances from the remaining objects to the mode, decreases by 1, because we lost *one* object.

The value of t can also be the same as the mode value. For example, the value a in attribute f_1, and x in attribute f_2. This can be further divided into two cases. One case is: if a departs, the mode doesn't change. Meanwhile, the objective value stays the same, because the departed object had a distance of 0.

The other case is when duplicated mode exists. In f_2, y is a duplicated mode for x. If x departs, the mode becomes y. The change is computed as follows: before the departure, the mode is x, and $F = \sum d(y, x) = 4$; after the departure, the mode is y, $F = \sum d(x, y) = 3$. So the result is objective value decreases by 1. Note that we don't need to check other objects who are neither x nor y, because their distances to the mode remain to be 1.

A similar analysis can be made for the arriving cluster Q, which we do not provide in detail. The results are summarized in Table 2.

In the above analysis, the changes of modes (or sometimes no change) have also been discussed. After each transfer, the modes need to be updated accordingly. This is a trivial task that can be done in O(D).

By referring to Table 2, the transfer test can be computed in O(D). Thus the algorithm has a complexity that is linear to the number of objects, which is a desired quality for partitioning clustering methods. The whole algorithm is as follows:

1. Initialize a partition (for example, a random one) of the data set.

2. Scan every object of the data set. If there is an object that satisfies the transfer test (computed by referring to Table 2), relocate it to the cluster that has the most decrease of the objective function, i.e. the one with the biggest ΔF. Update the modes and the frequency tables of the two involving clusters.

3. Repeat Step 2 until no objects are relocated in a full cycle scan of the whole data set.

3 Advantages of Hartigan's Method

In k-means, the prevalent optimization method is called Lloyd's method, which is also used by k-modes. Recently it has been shown that Hartigan's method is superior than Lloyd's method on k-means, in that better optima are discovered (Telgarsky 2010). In this section, we rigorously prove a similar result on k-modes.

Lloyd's method and Hartigan's method are two different heuristics that would end up with different optima. One question is, can the result of Lloyd's method be further improved by Hartigan's method? That is to say, if we use the result of Lloyd's method as the initial partition for Hartigan's method, are there any relocations that can be made? Or in the opposite direction, can the result of Hartigan's method be further improved by Lloyd's method?

Theorem 1. Any optimum achieved by Hartigan's method, can not be further improved by Lloyd's method.

Proof. Suppose there is an object t that should be relocated according to Lloyd's method, from cluster P to cluster Q. Without losing generality, we assume there is only one attribute. Let the attribute value of t also be t, the modes of cluster P and Q be m_P, m_Q.

According to Lloyd's method, $d(t, m_P) > d(t, m_Q)$. To satisfy this, there is only one possible combination: $d(t, m_P) = 1$, and $d(t, m_Q) = 0$.

Refer to Table 2. $d(t, m_P) = 1$ corresponds to a Not-Mode departure, with a change of -1. $d(t, m_Q) = 0$ corresponds to a Is-Mode arrival, with a change of 0. So the transfer test is satisfied, which means it's not a final result of Hartigan's method. The contradiction proves the theorem.

Theorem 2. For some optima achieved by Lloyd's method, they can be further improved by Hartigan's method. In other words, Hartigan's method can escape certain local optima which trap Lloyd's method.

Proof. To simplify, we also look at the case when there is only one attribute. In order to satisfy the transfer test, the change of departure has to be -1, and the change of arrival has to be 0. From Table 2, both departure and arrival have two cases being -1 and 0 respectively, thus four combinations:

1. Not-mode -> Is-mode: $(d(t, m_P) = 1) > (d(t, m_Q) = 0)$.

2. Not-mode -> Not-mode and Is-Duplicated-Mode: $(d(t, m_P) = 1) = (d(t, m_Q) = 1)$.

3. Is-mode and Exists-Duplicated-Mode -> Is-mode: $(d(t, m_P) = 0) = (d(t, m_Q) = 0)$.

4. Is-mode and Exists-Duplicated-Mode -> Not-mode and Is-Duplicated-Mode: $(d(t, m_P) = 0) < (d(t, m_Q) = 1)$.

We can see that for the first combination, the Lloyd's method will do the relocation. For the other three combinations, the Lloyd's method will do nothing and converges, while Hartigan's method will make the transfer. So if Lloyd's method ends up with a partition which satisfies any one of the three combinations, Hartigan's method can proceed, further improving the objective function.

Now let's see the implications from the two theorems. Firstly, how often is Hartigan's method helpful when it's used after Lloyd's method converges? Sometimes it's more likely for Lloyd's method to end up in either one of three cases listed above, thus it's more often that local optima found by Lloyd's method are improvable by Hartigan's method. What are the characteristics of such situations? One common characteristic of the three cases in the proof is that there are duplicated modes in clusters. Duplicated modes often appears in smaller clusters: say we have two attribute values a and b, it's a bigger probability for a cluster of 10 objects ending up splitting a and b (5 objects each) , than for a cluster of 1000 objects to make the split. So for a fixed data set, the more clusters we want to find, the more helpful is Hartigan's method. Also as the number of features rises, it's more likely there are duplicated modes on at least one of the features, thus Hartigan's method is more helpful for high dimensional data.

Secondly, after a single run of each of the two algorithms, Hartigan's method is not always better than Lloyd's method, even if they start with the same initial partition. This is because they take different greedy choices along the way, and usually end up in different parts of the optimization spaces.

Thirdly, although there seems to have no rigorous way to compare the performances of the two methods used independently, Hartigan's method should perform better on average under certain assumptions. Suppose we first run Lloyd's method and have a resulting partition Π_1. Then we run Hartigan's method, and somewhere in the process, it reaches the same partition Π_1. Then the final result, say Π_2, is definitely better than Π_1. This is the case when Hartigan's method outperforms Lloyd's method in a single run. If we make the assumption that in other cases, the two methods are head to head at the quality of local optima, then altogether Hartigan's method is better. This is validated in empirical evaluations.

4 Empirical Results

In this section, our goal is to show that Hartigan's method does outperform Lloyd's method. Specifically, applying Hartigan's method after Lloyd's method can show improvement of the objective; using two methods independently, Hartigan's method also achieves better averaged performance. With respect to computational cost, empirical results show that when the numbers of attributes and clusters grow, Hartigan's method requires less iterations than Lloyd's method.

When initializing the two methods, generally we can use either *random partition* or *random seeds*. Random partition is to assign all objects to random clusters; random seeds is to select K objects to be the initial modes, and assign the remaining objects to nearest modes. In our experiments, we found that the two initialization methods lead to similar performance comparisons of Hartigan's method and Lloyd's method. Thus we report only part of the results: for synthetic data, we report results from random parti-

Table 3: Improvement of using Hartigan's method after Lloyd's method: synthetic data sets

	Times	Max	Mean	Std
K=2	52	41	23.87	23.49
K=4	124	67	23.96	30.34
K=6	208	79	21.62	32.2
K=8	362	102	20.57	33.69
K=10	514	77	21.03	31.97
	Times	Max	Mean	Std
D=10	42	72	25.57	29.07
D=15	124	67	23.96	30.34
D=20	196	108	26.17	34.88
D=25	254	130	27.15	39.95
D=30	364	132	30.88	44.55

tion; for real-world data, we report results from random seeds. Also note that all the data are recorded by running algorithms for 1000 times.

4.1 Using Hartigan's method After Lloyd's Method

Here, we run Lloyd's method first, then on the resulting partition, we run Hartigan's method. Sometimes Hartigan's method can not proceed, but when it does proceed, the optima can be further improved.

First look at the results on synthetic data (categorical values are uniformly distributed) in Table 3. We vary the number of clusters K, and the number of features D, set the invariant parameters to be 1000 objects, 15 features and 4 clusters. On the columns, "Times" refers to the number of times (in 1000 runs) when Hartigan's method can proceed on the final results of Lloyd's method; "Max" and "Mean" refers to the maximum and average improvement of the objective function; "Std" refers to the standard deviation of the local optima achieved by Lloyd's method.

As we can see, the number of times when Hartigan's method is helpful increases when K and D increases, until it's quite significant (almost half of the times when K=10). This is consistent with our analysis after Theorem 2. Furthermore, from the comparison of maximum improvement, average improvement and the standard deviation, we can see that the improvements are quite significant, as they are comparable with the standard deviation.

For the real-world data, we picked three commonly used categorical data sets from UCI machine learning repository (Bache 2013): mushroom, soybean and congressional voting records. Except for the soybean data, which has only 35 records, we varied the number of clusters. The results are in Table 4. A similar trend from the synthetic data is found: as K increases, more often Hartigan's method further improves upon Lloyd's method. Although the average improvement is less significant than synthetic data (possibly due to attribute value distributions far from uniform), the maximum improvement is still quite large when comparing to the standard deviation.

4.2 Hartigan's Method and Lloyd's Method Independently

Here we run Hartigan's method and Lloyd's method independently and compare their performances. From the analysis after Theorem 2, we argue that generally Hartigan's method should outperform Lloyd's

Table 4: Improvement of using Hartigan's method after Lloyd's method: real-world data sets

data	K	Times	Max	Mean	Std
Mushroom	2	12	1270	137.25	1745.71
	4	315	2827	40.79	2657.93
	6	541	1415	78.11	1894.09
	8	628	6048	124.61	1625.3
Soybean	4	459	69	10.07	21.78
Congress	2	32	3	3.00	92.01
	4	119	110	13.45	37.75
	6	168	136	16.33	36.68
	8	321	104	14.49	36.33

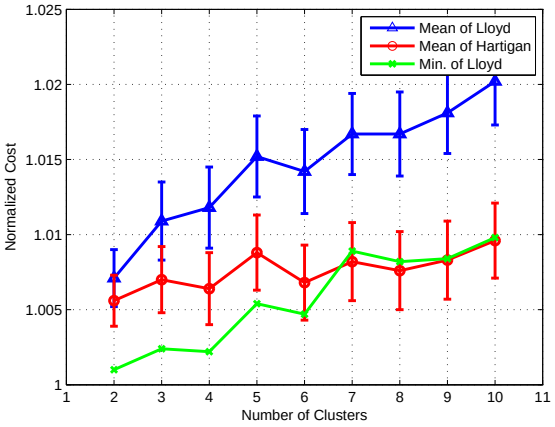


Figure 1: Normalized cost of two methods varying the number of clusters

method, because of the part played by Theorem 2. This is verified by our results here.

Figure 1 and 2 are results from synthetic data sets. Again the numbers of clusters and features are varied. The costs (value of objective function) are normalized by the best optima found by Hartigan's method. So only the minima of Lloyd's method are pictured. We can see that in all cases, Hartigan's method found better minimum optima, and better average optima. From the range of the error bars (standard deviation), we can conclude that the advantage is quite significant. Also the advantage grows as K or D increases. In some extreme cases, e.g, 7 clusters in Figure 1, even the minimum optimum by Lloyd's method is worse than the averaged optima by Hartigan's method.

In Table 5 for real-world data, we also listed the normalized costs of the two methods, their respective normalized standard deviations (Std), and the advantage of Hartigan's method over Lloyd's method measured by percentage (Per.). Like the results in Section 4.1, the advantage of Hartigan's method is less significant comparing to synthetic data, but it's there and it's more significant when K increases (see the decrease of advantage percentage).

4.3 Computational Cost

As we have shown, the Hartigan's method we developed for k-modes has the same time complexity with Lloyd's method for k-modes. However, the number of iterations required until converge is still a factor we are not sure about. Note that the equivalent notion of one iteration for Hartigan's method is one scan of the data set. Here we test the number of iterations

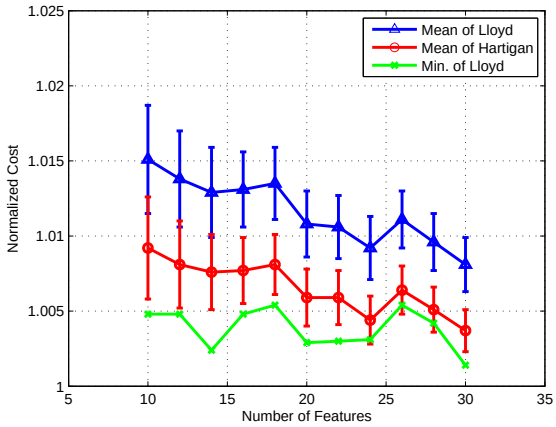


Figure 2: Normalized cost of two methods varying the number of features

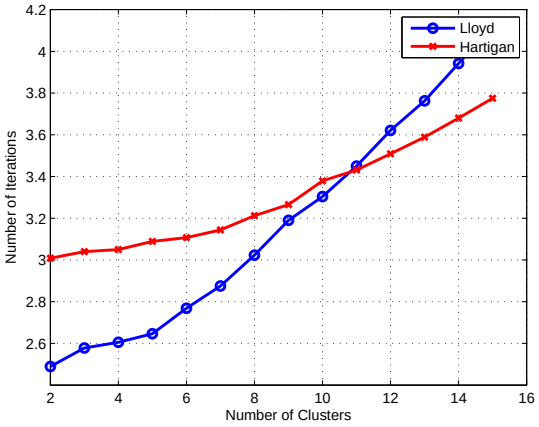


Figure 3: Iterations required by two methods varying the number of clusters

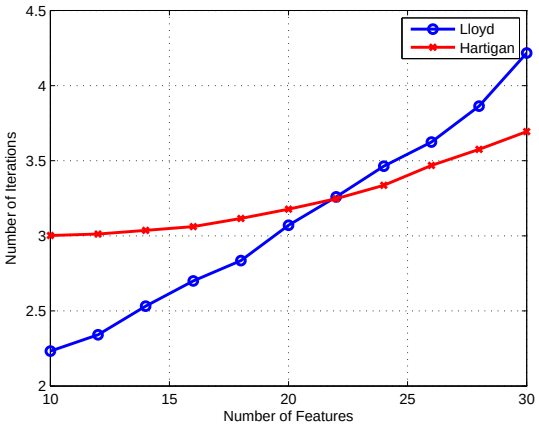


Figure 4: Iterations required by two methods varying the number of features

Table 5: Comparison of two methods independently used: real-world data sets

Data	k	Lloyd	Std	Hartigan	Std	Per.
Mush.	2	1.023	0.028	1.021	0.026	0.998
	4	1.054	0.055	1.050	0.053	0.996
	6	1.058	0.049	1.055	0.047	0.997
	8	1.072	0.044	1.067	0.038	0.995
Soy.	4	1.123	0.117	1.074	0.082	0.956
Con.	2	1.004	0.039	1.001	0.001	0.998
	4	1.060	0.026	1.058	0.024	0.997
	6	1.077	0.028	1.070	0.029	0.993
	8	1.088	0.030	1.079	0.030	0.992

for the two methods on the same synthetic data sets from above, by varying the number of clusters and the number of features, respectively. The unvaried parameters are 4 clusters and 15 features.

From Figure 3 and Figure 4, we can see that Hartigan's method begins to run less iterations when the number of clusters and features become larger. The crossovers are at 11 clusters and 22 features. The two numbers are relatively small, so we can say that generally Hartigan's method requires less computational cost.

As of why the number of iterations have this trend, a reasonable explanation is: as the number of clusters and features grow, Hartigan's method allows more relocations to happen in the early iterations. Since more work is done early on, less overall iterations are needed for Hartigan's method.

5 Final Remarks

To summarize, we developed an efficient Hartigan's method for k-modes, and proved that Hartigan's method can further improve the objective after Lloyd's method converges. Empirical results supported the conclusions. Independently used, Hartigan's method also achieves better optimization performance. Overall Hartigan's method is a better optimization method for k-modes.

Our result is consistent with the k-means literature (Telgarsky 2010, Slonim 2013), that Hartigan's method has better overall performance over Lloyd's method. The reason of this consistency is, although k-means and k-modes are two different objective functions, their Hartigan's methods have the same basic idea. The idea is to relocate objects according to the exact change of objective function. This works better for optimization than the more heuristic Lloyd's method.

For k-means (Telgarsky 2010, Slonim 2013), online k-means was also discussed as another optimization method, which has indistinguishable performance comparing with Lloyd's method. The Online method is slightly different in that it consider one point a time and update centers immediately after a relocation. For k-modes, our experiments also show close performances of the two methods, making it a trivial point that we didn't include.

References

Kaufman, L., Rousseeuw, P. J. (2009). Finding groups in data: an introduction to cluster analysis (Vol. 344). John Wiley & Sons.

Everitt, B. S., Landau, S., Leese, M. and Stahl, D. (2011). Cluster Analysis, 5th Edition. John Wiley & Sons.

Steinley, D. (2006). K-means clustering: a half-century synthesis. British Journal of Mathematical and Statistical Psychology, 59(1), 1-34.

Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. Data mining and knowledge discovery, 2(3), 283-304.

Chaturvedi, A., Green, P. E., Caroll, J. D. (2001). K-modes clustering. Journal of Classification, 18(1), 35-55.

Daylight, Chemical Information Systems, Inc. http://www.daylight.com/

Lloyd, S. (1982). Least squares quantization in PCM. Information Theory, IEEE Transactions on, 28(2), 129-137.

MacQueen, J. (1967, June). Some methods for classification and analysis of multivariate observations. In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability (Vol. 1, No. 14, pp. 281-297).

Hartigan, J. A. (1975). Clustering algorithms.

Hartigan, J. A., Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. Applied statistics, 100-108.

Telgarsky, M., Vattani, A. (2010). Hartigan's Method: k-means Clustering without Voronoi. In International Conference on Artificial Intelligence and Statistics (pp. 820-827).

Slonim, N., Aharoni, E., Crammer, K. (2013, August). Hartigan's K-means versus Lloyd's K-means: is it time for a change?. In Proceedings of the Twenty-Third international joint conference on Artificial Intelligence (pp. 1677-1684). AAAI Press.

Xiang, Z., Ji, L. (2013). The Use of Transfer Algorithm for Clustering Categorical Data. In Advanced Data Mining and Applications (pp. 59-70). Springer Berlin Heidelberg.

Tarsitano, A. (2003). A computational study of several relocation methods for k-means algorithms. Pattern recognition, 36(12), 2955-2966.

Bache, K., Lichman, M.: UCI Machine Learning Repository. (2013). University of California, School of Information and Computer Science, Irvine, http://archive.ics.uci.edu/ml

Tree Based Scalable Indexing for Multi-Party Privacy-Preserving Record Linkage

Thilina Ranbaduge, Peter Christen, and Dinusha Vatsalan

Research School of Computer Science, College of Engineering and Computer Science,
The Australian National University,
Canberra ACT 0200, Australia,
Email: {thilina.ranbaduge, peter.christen, dinusha.vatsalan}@anu.edu.au

Abstract

Recently, the linking of multiple databases to identify common sets of records has gained increasing recognition in application areas such as banking, health, insurance, etc. Often the databases to be linked contain sensitive information, where the owners of the databases do not want to share any details with any other party due to privacy concerns. The linkage of records in different databases without revealing their actual values is an emerging research discipline known as privacy-preserving record linkage. Comparison of records in multiple databases requires significant time and computational resources to produce the resulting matching sets of records. At the same time, preserving the privacy of the data is becoming more problematic with the increase of database sizes.

We propose a novel indexing (blocking) approach for privacy-preserving record linkage between multiple (more than two) parties. Our approach is based on Bloom filters to encode attribute values into bit vectors. The Bloom filters are used to construct a single-bit tree, where the encoded records are arranged into different blocks. The approach requires the parties to only participate in a secure summation protocol to find the best bits to construct the trees in a balanced manner. Leaf nodes of the trees will contain the blocks with encoded records. These blocks can finally be compared using private comparison and classification techniques to determine the similar record sets in different databases. Experiments conducted with datasets of sizes up-to one million records show that our protocol is scalable with both the size of the datasets and the number of parties, while providing better blocking quality and privacy than a phonetic based indexing approach.

Keywords: Multi-party protocol, privacy technologies, scalability, Bloom filter, single-bit tree, secure summation protocol.

Funded by the Australian Research Council under Discovery Project DP130101801.

1 Introduction

Many organizations, including businesses, government agencies and research organizations, are collecting vast amounts of data that are stored, processed and analyzed for the improvement of their works. These data often contain millions of records. As the size of data is continuously increasing, developing techniques for efficient processing, analyzing and mining has gained much recognition in both academia and industry. Many application domains require information from multiple sources to be integrated and combined in order to improve data quality and to facilitate further analysis of the data. The process of matching records that relate to the same entities from different data sources is known as 'record linkage' (Fellegi & Sunter 1969).

In most occasions the linkage of records from multiple sources requires much computational resources for processing. The process becomes even more challenging when the records contain personal information. Organizations commonly do not want their sensitive information to be linked with other data sources due to growing privacy and confidentiality concerns. The research paradigm of finding records in multiple data sources that relate to the same entity without revealing personal information is known as 'privacy-preserving record linkage' (PPRL), 'blind data linkage' or 'private record linkage' (Al-Lawati et al. 2005, Churches & Christen 2004, Karakasidis & Verykios 2011, Yakout et al. 2012).

On occasions where unique identifiers for entities are available across all the databases to be linked, a simple database join would be trivial for the purpose of identifying the matching pairs of records. However, in most cases finding such a common identifier in all databases would not be possible. A possibility to overcome this issue is to use quasi-identifiers (QID) such as first name, last name, address details, age, etc. (Hawashin et al. 2011). This will allow to accurately link records, but it will reveal personal information to other parties involved in the linking process. In order to cope with privacy issues the values in those identifiers need to be somehow encoded.

Bloom filters, which were proposed by Bloom (1970), have widely been used for encoding of records in record linkage approaches. A Bloom filter is a bit array which can hold 1's and 0's according to a bit pattern where initially all bit positions are set to 0's. Records can be hashed by using hash encoding algorithms to generate bit patterns for the records. These bit pattens can then be included in a Bloom filter by setting the relevant bit positions to 1's. Several approaches (Lai et al. 2006, Schnell et al. 2009, de Vries et al. 2011, Vatsalan & Christen 2012) have used Bloom filters for matching record sets.

While preserving the privacy of the QID values, which are used for the linking process, one major challenge is to cope with scalability. Many different indexing or blocking approaches have been introduced to compare databases (Christen 2012*b*), because the naive approach of comparing all pairs of records is not feasible when the databases are large. An indexing mechanism reduces the large number of potential comparisons by removing as many record pairs as possible that correspond to non-matches. This decreases the amount of computational efforts required for the comparison of larger databases.

The aim of this paper is to propose a new indexing mechanism for multi-party PPRL that can provide better scalability, blocking quality, and privacy, which are important factors for any practical PPRL application. We introduce a tree based approach which uses a secure summation protocol to generate blocks. The paper also presents an empirical evaluation of the proposed approach with regard to scalability, blocking quality, and privacy.

The remainder of the paper is structured as follows. In the following section, we provide an overview of related work in PPRL. In Section 3 we describe the current problem of indexing with multiple parties. In Section 4 we provide a detailed description of our protocol, and in Section 5 we analyze our protocol with regard to complexity, quality, and privacy. In Section 6 we validate these analyses through an experimental study. Finally we summarize our findings and discuss future research directions in Section 7.

2 Related Work

Recently, a variety of indexing or blocking approaches have been developed for reducing candidate record pairs that need to be compared in record linkage. In the survey by Christen (2012*b*) a comprehensive review about existing indexing mechanisms is provided. Some of the developed approaches have been adapted for PPRL based on existing indexing techniques, such as standard blocking, mapping based blocking, clustering, sampling, and locality sensitive hash functions. In our research we mainly focus on developing a blocking mechanism that is scalable to large databases as well as with the number of parties while providing blocking quality and privacy, which are the trade-offs of any indexing step in PPRL.

A protocol proposed by Song et al. (2000) tried to address the problem of approximate matching by calculating enciphered permutations for private approximate record matching. Their suggested approach becomes impractical since predicting all possible permutations is impossible in real-world applications. A two-party protocol, which does not require a trusted third party to perform the linking as in three-party protocols, is suggested by Atallah et al. (2003). This approach allows the parties to compute the edit distance between strings without exchanging these strings, but this protocol is impractical due to the large amount of necessary communication required to compute the distances. Ravikumar et al. (2004) used a secure set intersection protocol for PPRL that requires extensive computations, which makes the protocol impractical for large datasets.

A blindfolded multi-party approach was suggested by Churches & Christen (2004), which uses q-gram hash digests to achieve approximate private linkage. Their approach is computationally costly, because of the generation of power sets of the q-grams of record values. Al-Lawati et al. (2005) introduced three blocking mechanisms for three-party protocols which require a trusted third party to perform the linkage. Their work used hash signatures for comparison of records. Their methods provide a trade-off between privacy and computational and communication cost.

Another three-party protocol to provide privacy for both data and schema matching without revealing any information was presented by Scannapieco et al. (2007). It uses a greedy heuristic re-sampling method for arranging records into blocks. However, their experimental results indicate that the linkage quality is affected by this greedy heuristic re-sampling method. Inan et al. (2010) suggested an approach based on anonymization using a cryptography technique to solve the PPRL problem. Their blocking step used value generalization hierarchies and secure multi-party computation (SMC) based matching (Yao 1986) by using a cryptographic technique, which is computationally expensive to perform.

Bloom filters are used commonly in the PPRL context, due to their capability for computing similarities. Various approaches have been suggested for similarity calculation in PPRL by using Bloom filters (Lai et al. 2006, Schnell et al. 2009, Durham 2012, Vatsalan & Christen 2012, Bachteler et al. 2013).

A multi-party approach was proposed by Lai et al. (2006) that uses Bloom filters to securely transfer data between multiple parties for private set intersection. All the records are encoded into Bloom filters and segmented according to the number of parties involved in the protocol. These segments are shared among the parties, where each party will perform a logical conjunction (and) on segments. Each party compares its own full Bloom filter with the segments of the other parties for matches. Their evaluated results showed that the false positive rate increases with the number of parties involved in the communication, where none of the parties will be able to guess the existence of a given record correctly.

A three-party approach was introduced by Schnell et al. (2009), which performs approximate matching of records by using Bloom filters. The string values in the attributes are converted into sets of q-grams which are then mapped into a Bloom filter using a double hashing mechanism. Durham (2012) proposed a three-party framework for PPRL using Bloom filters. She suggested record-level Bloom filters for encoding all attribute values of a record into a single Bloom filter. Locality sensitive hashing (LSH) functions are used to reduce the computational complexity of the private blocking.

An iterative two-party protocol was proposed by Vatsalan & Christen (2012), which reveals selected bits in the Bloom filters between two database owners. The approach classifies record pairs into matches and non-matches in an iterative way to reduce the number of pairs with unknown match status at each iteration without compromising privacy.

Kristensen et al. (2010) used Bloom filters to efficiently find similar chemical fingerprints in a database based on a user defined similarity threshold value. Each chemical fingerprint was represented by a Bloom filter. For rapid screening of fingerprints among the set of Bloom filters, they introduced a novel tree data structure known as multi-bit tree. All fingerprints are arranged in a binary tree data structure according to the value in selected bit positions. These bit positions are selected to keep the tree structure as balanced as possible. According to the experiments conducted by the authors, it was noted that the performance of the queries increased with the use of this tree data structure. The tree reduced the amount of comparison calculations and computationally scaled linearly when the size of the datasets was increasing.

Table 1: Number of candidate record sets generated with multiple parties for different sizes of datasets and blocks.

	Number of parties			
Data set / Block size	3	5	7	10
10,000 / 10	10^6	10^8	10^{10}	10^{13}
10,000 / 100	10^8	10^{12}	10^{16}	10^{22}
10,000 / 1,000	10^{10}	10^{16}	10^{22}	10^{31}
100,000 / 10	10^7	10^9	10^{11}	10^{14}
100,000 / 100	10^9	10^{13}	10^{17}	10^{23}
100,000 / 1,000	10^{11}	10^{17}	10^{23}	10^{32}
1,000,000 / 10	10^8	10^{10}	10^{12}	10^{15}
1,000,000 / 100	10^{10}	10^{14}	10^{18}	10^{24}
1,000,000 / 1,000	10^{12}	10^{18}	10^{24}	10^{33}

The concept of multi-bit trees was further extended by Bachteler et al. (2013) as a new blocking method for record linkage. In their approach they used Bloom filters to hold the data. The string values in the attributes are first converted into sets of q-grams, which are then mapped into a Bloom filter using a double hashing mechanism. The generated Bloom filters are then partitioned into separate bins according to the number of bits set to 1. All the Bloom filters in each bin are stored in a multi-bit data structure. A given Bloom filter will be queried against all Bloom filters that are stored in the multi-bit tree and at each node the similarity is calculated to find matches. According to the experiments conducted it was noted that the proposed blocking approach is performing better than existing blocking methods such as standard blocking, canopy clustering and sorted neighborhood approaches, while scaling linearly in terms of total running time for construction and querying of a multi-bit tree.

3 Problem Statement

As mentioned in Section 1, generating the candidate record sets for multiple databases becomes computationally expensive when the number of parties is increasing. In such situations, methods or techniques to reduce the comparison space are needed. In the record linkage process these methods are referred to as blocking or indexing methods (Christen 2012*b*). Such methods identify reduced sets of candidate records for comparison and classification, by keeping true matching records in sets of candidate records while removing as many of the true non-matching record sets as possible.

When the number of parties is increasing, more sophisticated indexing mechanisms are required. A larger number of blocks with a small number of records, or a smaller set of blocks with a large number of records, will still require more comparisons. This problem is highlighted in Table 1 to illustrate the number of candidate record sets generated for different number of parties with various datasets and block sizes (by assuming all blocks have the same size).

Table 1 shows how the number of generated candidate record sets grows exponentially with the number of parties involved in a multi-party protocol, and how even with very small sized blocks (e.g. 10 records per block per party) the number of candidate record sets becomes prohibitively large to be practically feasible.

One major problem in currently available indexing solutions is that not enough control is available over the block sizes. Generating different sizes of large number of blocks makes the comparison step even more problematic and requires more computational time. To overcome this problem, in our protocol we provide a parameterized solution, where the user can control the size of the blocks that will be generated. The protocol is based on an indexing mechanism that can generate blocks of records in a balanced tree data structure. Each participating party will have a similar tree data structure constructed holding blocks of records on leaf nodes. The construction of the tree is done in a secure manner without exchanging any information about the records that are held by each party.

4 Tree Based Scalable Indexing for PPRL

In this section we provide details on how the proposed indexing mechanism works. We highlight how attribute values are encoded into Bloom filters and how they are used to construct the tree data structure to hold the blocks. First we will explain the details of the building blocks that are needed for the construction of the index.

4.1 Building Blocks

4.1.1 Bloom Filters

Bloom filters are data structures proposed by Bloom (1970) for checking element membership in any given set (Broder & Mitzenmacher 2004). A Bloom filter is a bit vector of length m, where initially all the bits are set to 0. In order to map an element into the domain between 0 and $m-1$ of the Bloom filter, k independent hash functions $h_1, h_2, \ldots, h_k$ are used. Furthermore, to store n elements of the set $S = \{s_1, s_2, \ldots, s_n\}$ into the Bloom filter, each element $s_i \in S$ is encoded using the k hash functions and all bits having index positions $h_j(s_i)$ for $1 \leq j \leq k$ are set to 1.

4.1.2 Q-grams

A q-gram (also known as n-gram) is a character substring of length q in a string (Christen 2012*a*). Often string values are prefixed and suffixed, which is also known as padding, with a special character of length $q-1$ before they are converted into q-grams (Bachteler et al. 2013). This padding character helps to emphasize the first and last characters of a string. A q-gram of length 2 is known as a *bigram* or *digram* and a q-gram of length 3 is known as a *trigram*. A string s of length c contains $l = c - q + 1$ q-grams (Christen 2012*a*). For an example, the string "THILINA" can be padded with character '_' and the resulting *bigram* ($q = 2$) set is $\{\_T, TH, HI, IL, LI, IN, NA, A\_\}$.

In our approach we used q-gram sets of the QIDs to convert these QIDs into Bloom filters. First, the selected QID values of a given record are converted into a q-gram set. Then each q-gram set is stored in a Bloom filter by using k hash functions. This is repeated for all records in the dataset. Figure 1 illustrates the transformation of a record's QID value into a Bloom filter.

4.1.3 Optimal Parameter Settings for Bloom Filters

A crucial aspect that affects all three challenges (quality, scalability and privacy) of PPRL approaches that are based on Bloom filters is the parameter settings used to generate the Bloom filters. In this section we describe our choice of parameters, which follows

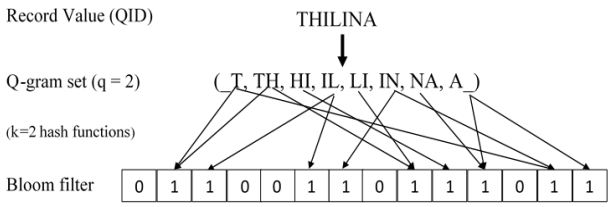


Figure 1: Mapping of a string value into a Bloom filter of $m = 14$ bits by using $k = 2$ hash functions.

earlier Bloom filter based PPRL approaches (Schnell et al. 2009, Durham 2012, Durham et al. 2013, Vatsalan & Christen 2012).

For a given Bloom filter length, m, and number of elements (e.g. q-grams) to be inserted into the Bloom filter, n, the optimal number of hash functions, k, that minimizes the false positive rate, r, can be calculated as (Mitzenmacher & Upfal 2005)

$$k = \frac{m}{n} ln(2), \tag{1}$$

leading to a false positive rate of

$$r = \left(\frac{1}{2^{ln(2)}} \right)^{m/n}. \tag{2}$$

We can calculate the value for n by analyzing the content of a dataset to be used for a PPRL project by calculating the average number of q-grams that are generated from a record (i.e. we convert attribute values into q-grams as described above and count how many q-grams are generated on average for a record).

The value of m determines how much memory and communication will be required in our PPRL protocol. For a given m, we can calculate k based on n as calculated from the datasets. For a certain dataset and n, the larger m the larger k will be, with larger values of k requiring more computation as more hash values need to be calculated and mapped into a Bloom filter for each record.

While k and m determine the computational aspects of our approach, quality and privacy will be determined by the false positive rate r. A higher value for r will mean a larger number of false matches (i.e. a set of non-matching records classified to correspond to the same entity), and thus lower quality. At the same time, a higher false positive rate r will also mean improved privacy, as false positives mean an adversary cannot be absolutely sure that a certain Bloom filter corresponds to a certain record (Schnell et al. 2009, Durham 2012, Vatsalan & Christen 2012).

It was proven (Mitzenmacher & Upfal 2005) that a Bloom filter should ideally have half its bits set to 1 (i.e. 50% filled) to achieve the lowest possible false positive probability for given values of n, m and k. Equations 1 and 2 in fact lead to a probability that a bit in a Bloom filter is set to 1 as $p = e^{-kn/m} = 0.5$ (Mitzenmacher & Upfal 2005). For PPRL this is important, because the bit patterns and their frequencies in a set of Bloom filters can be exploited by a cryptanalysis attack (Kuzu et al. 2013). Such an attack exploits the fact that Bloom filters that are almost empty can provide information about rare q-grams and thus rare attribute values.

In our experimental evaluation we will set the Bloom filter parameters for our approach according to the discussion presented here and following earlier Bloom filter work in PPRL (Schnell et al. 2009).

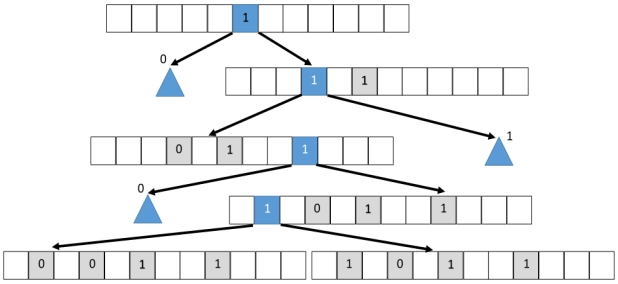


Figure 2: Single-bit tree : The blue squares represent the bits chosen for the given node, while the light gray squares mark bits chosen at an ancestor. The blue triangles represent sub-trees that are not shown.

4.1.4 Single-bit Tree Data Structure

A single-bit tree is a binary tree data structure that can be used to store information of a set of bit vectors (Kristensen et al. 2010). The construction of the tree starts from the root node, where all bit vectors are assigned to this node. At each node in the tree a position in the bit vectors is chosen to best split all the children of the node into two parts of equal size, which in turn will keep the tree as balanced as possible. All bit vectors with a 0 at that position are stored in the left subtree while all bit vectors with a 1 are stored in the right subtree. This division is continued recursively until all the bit vectors in a given node are the same, or all the bit positions have been used for the construction. Figure 2 shows an example of a single-bit tree.

It is not directly apparent how best to choose which bit position to split the data on at a given node when building the tree data structure. The selection of the best splitting bit position requires information about all the bit vectors held by a given node. This requires to view all the child bit vectors in a given node and select a bit position which contains 0 in half of the children and 1 in the other half.

The continuation of the recursive division is based on the bit vectors available in a given node and bit positions used in each parent node. Other than these two factors, in our protocol we provide a parametric solution to control this recursive division. The construction of the single bit tree data structure is described in more detail in Section 4.2.2.

4.1.5 Secure Summation Protocol

The secure summation protocol is a method used in secure multi-party computation, which has been used in several record linkage approaches (Clifton et al. 2002, Rashid et al. 2009). Secure multi-party computation was first introduced by Yao (1986) with the idea of performing computations securely such that at the end of the computation no party knows anything except its own input and the final results of the computed function (O'Keefe et al. 2004, Lindell & Pinkas 2009, Cheng et al. 2010). The secure summation protocol allows multiple cooperating parties to compute a sum over their individual data without revealing their data to the other parties.

The idea behind the secure summation protocol can be described as follows (Clifton et al. 2002, Karr et al. 2004). Protocol 1 shows the steps involved. Assume there are P parties with each one having a secret input a_i, where $1 \leq i \leq P$. The parties want to compute the summation of these inputs. Initially party P_1 chooses a large random number r and then

Protocol 1: Basic secure summation protocol

Input:
- **P** : Number of parties
- $\mathbf{a_i}$: A secret input, $1 \leq i \leq P$

Output:
- s: Final sum , where $s = \sum_1^n a_i$

1: Party P_1 generates a random number r
2: Party P_1 computes partial sum $s_1 = a_1 + r$
3: Party P_1 sends s_1 to P_2
4: **for** ($i = 2$ to P) **do**
5: Party P_i computes partial sum $s_i = s_{i-1} + a_i$
6: **if** $i = P$ **then**
7: Party P_i sends s_i to party P_1
8: **else**
9: Party P_i sends s_i to party P_{i+1}
10: Party P_1 computes final sum $s = s_P - r$
11: Party P_1 sends final sum s to other parties.

adds this random number to his input a_1. Then party P_1 sends this value to party P_2. Since R is random, party P_2 learns effectively nothing about a_1.

Party P_2 adds his value a_2 to $r+a_1$, and sends the result to party P_3. This process repeats until all the parties have added their values and the partial sum $s = r+a_1+\ldots+a_P$ is received by the first party. Then the first party subtracts r from s and the resulting sum is distributed to all the other parties. Once the computation is finished each party only knows the total sum, from which they are unable to derive the other parties' information.

4.2 Blocking Approach

The previous sections provided details about the building blocks, which are needed for the construction of the blocking mechanism, and here we will elaborate in detail how the single-bit trees can be used as a blocking mechanism in the multi-party PPRL context. The construction of the index for a dataset of an individual party contains two main phases.

1. Generate Bloom filters for the records in the dataset.
2. Construct the single-bit tree by using the generated Bloom filters. This phase can be further extended into three sub phases, which are:
   (a) Perform secure summation to find the best splitting bit position.
   (b) Split the set of Bloom filters.
   (c) Generate the child nodes of the tree.

Each party needs to follow these phases to construct the single-bit tree for their own dataset. The overall indexing protocol, which includes all these phases mentioned above, is outlined in Protocol 2.

4.2.1 Generation of the Bloom Filters

Before the construction of the trees, the set of records needs to be encoded into Bloom filters as given in line 1 in Protocol 2. All parties need to agree upon a bit array length m (length of the Bloom filter); the length (in characters) of grams q, the k hash functions, and a set of attributes (blocking key attributes) that are used to link the records. As per step 1 in Protocol 2, each party needs to iterate over its dataset and each record needs to be encoded. Based on the optimal parameter settings calculated as described in Section 4.1.3, the blocking key values in a record are encoded

Protocol 2: Single-bit tree indexing protocol

Input:
- $\mathbf{D_i}$: Dataset belonging to party P_i
- A: Set of selected attributes
- s_{min}: Minimum bucket size
- s_{max}: Maximum bucket size

Output:
- T_i : Single-bit tree (the tree structure for dataset D_i)

Phase 1 :
1: B = generateBloomfilters(D_i, A)

Phase 2 :
2: $T_i.root$ = makeNode(B)
3: $q = [T_i.root]$ // Initialization of queue
4: **while** $q \neq \emptyset$ **do**
5: $n = q$.pop() // Get the current node

Phase 2.a:
6: R = generateRatios(n) // Generate local bit ratios
7: R_g = secureSummation(R) // Get ratios globally
8: b = getBestBit(R_g)

Phase 2.b:
9: n_0, n_1 = splitNode(n, b)

Phase 2.c:
10: **if** ($|n_0.B| \geq s_{min}$) AND ($|n_1.B| \geq s_{min}$) **then**
11: $n.left = n_0$ // Add left child
12: $n.right = n_1$ // Add right child
13: **if** ($|n_0.B| \geq s_{max}$) **then** // If blocks too large
14: q.push(n_0) // add to queue
15: **if** ($|n_1.B| \geq s_{max}$) **then** // If blocks too large
16: q.push(n_1) // add to queue

17: *return* T_i

into the Bloom filter using k hash functions. This encoding will be performed for all records in the dataset. Each party needs to generate the set of Bloom filters from their own dataset before proceeding to construct their tree.

4.2.2 Construction of the Trees

In the second phase of the protocol, each party can construct their single-bit tree by using the generated Bloom filters from their dataset. The construction of a tree for an individual party is described based on the lines of Protocol 2.

Before starting the construction process all the parties need to agree upon the two parameters of *minimum bucket size* (s_{min}) and *maximum bucket size* (s_{max}). These parameters specify the minimum and maximum number of records that need to be included in a block (bucket), respectively. The use of these parameters is elaborated in the relevant phases of the protocol below. Figures 3 to 7 illustrate the steps of Protocol 2 with an example for three parties.

- **Phase 2 : Initialization**
  As the initial step of the construction, a root node is created and the list of Bloom filters is assigned to the respective root node as the node data. A queue will be created to hold the nodes, where nodes are created at each iteration in the tree construction. Initially the root node is assigned into the queue. The iterations will continue until the queue becomes empty as per lines 2 to 5 in Protocol 2.

Party A

Bloom filter set A

A1	0	1	1	0	0
A2	1	0	1	0	1
A3	1	0	0	0	0
A4	0	1	1	1	0
A5	0	1	0	0	1
A6	1	1	0	1	0
A7	0	1	1	1	0
A8	0	1	0	0	0

Abs Diff from 0.5

1/8	1/4	0	1/8	1/4

Party B

Bloom filter set B

B1	1	0	0	1	1
B2	1	0	1	0	1
B3	0	0	1	1	0
B4	1	1	1	1	1
B5	1	1	0	0	0
B6	0	0	1	0	1
B7	0	0	0	1	1
B8	1	0	1	1	1

Abs Diff from 0.5

1/8	1/4	1/8	1/8	1/4

Party C

Bloom filter set C

C1	0	1	1	1	0
C2	0	1	0	1	0
C3	1	0	1	1	0
C4	1	1	0	0	1
C5	0	0	1	0	0
C6	1	0	0	1	0
C7	0	1	1	0	1
C8	0	1	0	1	0

Abs Diff from 0.5

1/8	1/8	0	1/8	1/4

Figure 3: Bloom filter generation and calculation of 0/1 bit ratios and absolute differences from 50% filled (Phase 2 in Protocol 2).

Random vector (R) =

10	5	12	13	6

10.125	5.25	12.0	13.125	6.25

→

10.25	5.5	12.125	13.25	6.5

→

10.375	5.625	12.125	13.375	6.75

Secure sums:

0.375	0.625	0.125	0.375	0.75

Final absolute differences from 50% filled:

0.125	0.208	**0.042**	0.125	0.25

Figure 4: Secure summation of absolute differences and selecting best bit for splitting (Phase 2.*a* in Protocol 2).

Party A

Bloom filter set A '0'

A3	1	0	0	0	0
A5	0	1	0	0	1
A6	1	1	0	1	0
A8	0	1	0	0	0

Abs Diff from 0.5

0	1/4	–	1/4	1/4

Party B

Bloom filter set B '0'

B1	1	0	0	1	1
B5	1	1	0	0	0
B7	0	0	0	1	1

Abs Diff from 0.5

1/6	1/6	–	1/6	1/6

Party C

Bloom filter set C '0'

C2	0	1	0	1	0
C4	1	1	0	0	1
C6	1	0	0	1	0
C8	0	1	0	1	0

Abs Diff from 0.5

0	1/4	–	1/4	1/4

Random vector (R) =

12	4	–	15	11

12.0	4.25	–	15.25	11.25

→

12.167	4.417	–	15.417	11.417

→

12.167	4.667	–	15.667	11.667

Secure sums:

0.167	0.667	–	0.667	0.667

Final differences from 50% filled:

0.056	0.222	–	0.222	0.222

Figure 5: Calculation of absolute differences and secure summation on sub-sets where bit 3 is 0 (Next iteration of Phase 2.*a* in Protocol 2).

Party A

Bloom filter set A '1'

A1	0	1	1	0	0
A2	1	0	1	0	1
A4	0	1	1	1	0
A7	0	1	1	1	0

Abs Diff from 0.5

1/4	1/4	–	0	1/4

Party B

Bloom filter set B '1'

B2	1	0	1	0	1
B3	0	0	1	1	0
B4	1	1	1	1	1
B6	0	0	1	0	1
B8	1	0	1	1	1

Abs Diff from 0.5

1/10	3/10	–	1/10	3/10

Party C

Bloom filter set C '1'

C1	0	1	1	1	0
C3	1	0	1	1	0
C5	0	0	1	0	0
C7	0	1	1	0	1

Abs Diff from 0.5

1/4	0	–	0	1/4

Random vector (R) =

16	8	–	7	14

16.25	8.25	–	7.0	14.25

→

16.35	8.55	–	7.1	14.55

→

16.6	8.55	–	7.1	14.8

Secure sums:

0.6	0.55	–	0.1	0.8

Final differences from 50% filled:

0.2	0.183	–	**0.033**	0.267

Figure 6: Calculation of absolute differences and secure summation on sub-sets where bit 3 is 1 (Next iteration of Phase 2.*a* in Protocol 2).

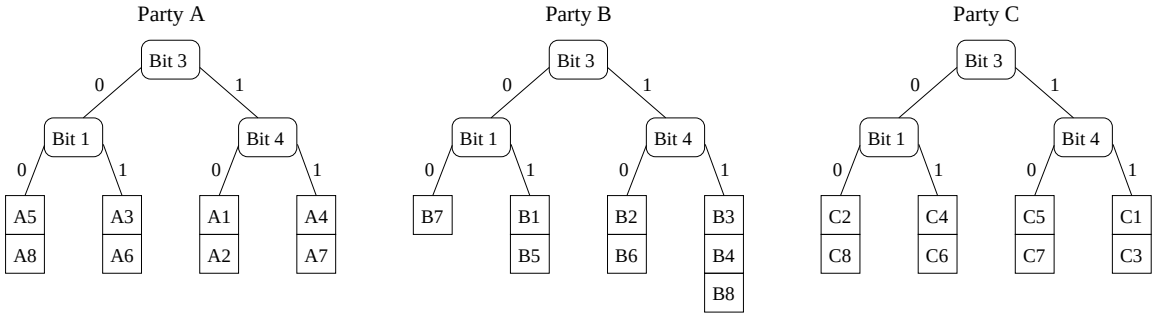


Figure 7: The resulting single-bit trees as generated by the example in Figures 3 to 6 with blocks across the three parties.

- **Phase 2.a : Find best bit position for splitting**
  At each iteration the node that is available at the front of the queue is processed. By processing the data available in this node, each party needs to generate a vector of length m that contains the values of ratios between the number of 0's and 1's for each bit position in the Bloom filters, as is calculated using Equation 3:

$$f_{ij} = abs(0.5 - \frac{o_{ij}}{l}), \tag{3}$$

  where f_{ij} is the ratio value of bit position i of party P_j, o_{ij} is the number of 1's in position i, and l is the number of Bloom filters processed in a given node.

  The bit positions that are having a value of 1 in half of the Bloom filters are given the lowest ratio value of 0, and the bit positions that are having 1's or 0's in all the Bloom filters are given the highest ratio value of 0.5. This processing is shown in lines 6 and 7 in Protocol 2.

  Once all parties have computed the ratio vectors of the bit positions locally based on their individual node data, a common bit position needs to be selected as the best bit for splitting the set of Bloom filters for the child nodes in the next level. For computing this global bit position, we extend Protocol 1 to securely compute the summation of these vectors of ratios, where each party P_j has as private input a vector f_j of length m, and the random value r in Protocol 1 is extended to a random number vector of the same length as the ratio vector. Once the secure summation step (line 7 in Protocol 2) is finished the globally summed ratio vector is used to find the best splitting bit position:

$$i_{best} = argmin\{i : (\frac{P}{2} - \sum_{j=1}^{P} f_{ij})\} \tag{4}$$

  Once each party receives the globally summed ratio vector, the best splitting bit position (i_{best}) is selected to divide the data which are available in the current node, as shown in line 8 of Protocol 2. This best splitting bit position is selected by ranking the globally summed ratio vector according to the summed values, where the bit position with the lowest sum gets the highest rank as shown in Equation 4.

- **Phase 2.b: Split the set of Bloom filters**
  The selected global best bit position is used to split all the Bloom filters of the current node into two portions. All the Bloom filters that contain a 0 in the best bit position are assigned to the left portion of the list and all others are assigned to the right portion. These portions are assigned to two new tree nodes, which are to be processed in the next iterations.

- **Phase 2.c : Generate the child nodes**
  According to the selected best bit position, these two portions may contain an uneven number of Bloom filters, i.e. one portion contains a smaller number of records. The disadvantage of such a division is that sensitive information can be revealed about the blocks that are having a smaller number of records, where an adversary can potentially re-identify individual records (Sweeney 2002). As a solution we propose the *minimum block size* (s_{min}) to guarantee that every block in the tree structure contains at least s_{min} records. After splitting, if any of the portions in the resulting lists contain less than s_{min} records, then these portions will not be assigned to any block in the given node. Instead the two portions are merged and the resulting list is included as a relevant block in the parent of the current node (line 10 in Protocol 2). All parties need to merge the two portions for the current iteration, which is informed by using the secure summation protocol as explained in Phase 2.*a*.

  If these portions contain a number of records greater than s_{min}, then these newly created nodes are assigned as child nodes to the current node, i.e. the node with the left portion becomes the left child of the current node and the other becomes the right child respectively (lines 11 and 12 in Protocol 2).

  One important consideration in the tree construction phase is to have control over the number of blocks created in the tree. We provide a parameter *maximum block size* (s_{max}) for this purpose. This allows the user to control the maximum number of records that can be contained in a block, which indirectly controls the number of iterations that occur when creating subtrees. After splitting the current node data, each portion is checked against the value of s_{max}. If any of the two portions contains less than s_{max} records, then a new block (bucket) is created to hold the relevant portion and is assigned to the current node. If the number of records is greater than s_{max}, then these two child nodes are added to the queue for future splitting (lines 13 to 16 in Protocol 2). Therefore the continuation of iterations is decided based on the number of blocks that are generated.

5 Analysis of the Protocol

In this section we analyze our protocol in terms of complexity, privacy, and quality of blocking.

5.1 Complexity

In this section we analyze the computational and communication complexities of our blocking protocol in terms of a single party. Let us assume there are N records in the dataset with each having an average of n q-grams. In the Bloom filter generation all the records are encoded using k hash functions. This encoding process is applied to all the records in a linear manner. Therefore the Bloom filter generation for a single party is of $O(k \cdot n \cdot N)$.

In the second phase of the protocol the single-bit tree construction starts once the records are encoded into N Bloom filters. In our protocol the parameter s_{max} is used to control the number of blocks, which indirectly controls the number of levels generated in the tree data structure. If s_{max} is equal to N, then the number of levels in the tree becomes 1 (all the records are assigned to the root node), while if s_{max} is equal to $N/2$ then the tree will have two levels (root node with two child nodes), and so on. When s_{max} is equal to 1 the single-bit tree is constructed with $log_2(N)$ levels. Therefore the number of levels in a single-bit tree can be calculated as $log_2(N/s_{max})$. At each level of the tree a total of N records are processed, where all child nodes in a given level hold a total of N records. Therefore the insertion of N Bloom filters into a single-bit tree requires a computational complexity of $O(N \cdot log_2(N/s_{max}))$.

In our protocol, the parties only need to communicate with each other in order to perform the secure summation protocol to find the best bit for splitting the data of the nodes, and to check if block sizes are less than s_{min}. This requires communication in the creation of each node. By assuming each party directly connects to other parties, the distribution of the final sum to P parties requires P messages for each node in the tree data structure, each of size m where m is the length of a Bloom filter. Therefore the entire protocol has a communication complexity of $O(m \cdot P \cdot 2^{N/s_{max}})$ for P parties. The computation of the secure summation protocol requires each party to process the data of the node in a given iteration, which needs a set of Bloom filters to be scanned for each bit position to get a count of the number of 1's. Therefore line 8 of our protocol has a computational complexity of $O(m \cdot N)$ for each level in the tree.

5.2 Privacy

In our protocol we assume each party follows the honest-but-curious (semi-honest) adversary model (Al-Lawati et al. 2005, Scannapieco et al. 2007), where each party follows the steps of the protocol while trying to find as much as possible about the data from the other parties. Privacy is a main factor that needs to be considered to evaluate the amount of information a party can learn from the data from the other parties when they communicate during the protocol. In our protocol, the parties communicate with each other to compute the global best bit position for splitting. For the exchange of ratio values of the Bloom filters our protocol uses a secure summation protocol.

During the secure summation, each party sums their ratio vector with the partial resulting vector sent by the previous party, but will not be able to learn any information about the ratio values of the previous party since the random vector is only known to the party that initiated the protocol. Once the initiated party received the final partial sum vector, he subtracts the random vector from the summed values, but is not capable of deducing anything about the other parties' ratio vector values. At the end of our blocking protocol, the set of blocks are generated and private linkage can be conducted on each respective block by using a private matching and classification technique (Atallah et al. 2003, Ravikumar et al. 2004, Vatsalan & Christen 2012, Durham et al. 2013), which should not reveal any information regarding the sensitive attributes and non-matches.

Our protocol performs a generalization strategy on the blocks that makes the re-identification from the perturbed data not possible (Sweeney 2002). The parameter *minimum block size* (s_{min}) is used to guarantee that every block in the tree structure contains at least s_{min} records. This ensures all blocks that are generated have the same minimum number of records, which makes a dictionary attack, where an adversary hash-encodes values from a large publicly available dataset using existing hash encoding functions, or a frequency attack, much more difficult (Vatsalan et al. 2013*b*).

5.3 Quality

The quality of our protocol is analyzed in terms of effectiveness, which requires all similar records to be grouped into the same block, and efficiency, which requires the number of candidate record sets generated to be as small as possible while including all true matching record sets (Vatsalan et al. 2013*b*). By assuming each block contains s_{max} records and there are b blocks in each tree for P parties, the number of candidate record sets generated by our approach is $((s^P_{max})b)$. The parameter s_{max} decides the number of child nodes that are created in the single-bit tree, which in turn controls the number of blocks generated. If the value of s_{max} is large, the number of blocks generated is reduced by the protocol, while a smaller s_{max} value provides a single-bit tree with more blocks. An optimal value for s_{max} needs to be set by considering factors such as the dataset size and the number of parties, such that both effectiveness and efficiency are achieved while guaranteeing sufficient privacy as well.

6 Experimental Evaluation and Discussion

We evaluated our protocol by performing experiments using a large real world database. In the following sub-sections we provide details on the datasets that we used for our experiments, implementation details of the proposed indexing protocol, and the evaluation measures used in the experiments.

6.1 Datasets

To provide a realistic evaluation of our approach, we based all our experiments on a large real-world database, the North Carolina Voter Registration (NCVR) database as available from `ftp://alt.ncsbe.gov/data`. This database has been used for the evaluation of various other PPRL approaches (Vatsalan et al. 2013*a*, Durham et al. 2013). We have downloaded this database every second month since October 2011 and built a combined temporal dataset that contains over 8 million records of voter's names and addresses.

We are not aware of any available real-world dataset that contains records from more than two parties that would allow to us evaluate our multi-party approach. We therefore created, based on the real NCVR database, a series of sub-sets as described next.

To allow the evaluation of our approach on different number of parties, with different dataset sizes, and with data of different quality, we used and modified a recently proposed data corruptor (Christen & Vatsalan 2013) to generate various datasets with different characteristics based on randomly selecting records from the NCVR database. During the corruption process we keep the identifiers of the selected and modified records, which allows us to identify true and false matches and therefore calculate various blocking quality and complexity measures as will be discussed in Section 6.3.

Specifically, we selected sub-sets from the full NCVR database for 3, 5, 7 and 10 parties, that contain 5,000, 10,000, 50,000, 100,000, 500,000 and 1,000,000 records, respectively. In each of these sub-sets, 50% of records were matches, i.e. half of all records occur in the sub-sets of all parties. We then applied various corruption functions on different numbers (ranging from 1 to 3) of randomly selected attribute values which allows us to investigate how our approach can handle 'dirty' data. We applied various corruption functions, including character edit operations (insertions, deletions, substitutions, and transpositions), and optical character recognition and phonetic modifications based on look-up tables and corruption rules (Christen & Vatsalan 2013).

We also created groups of datasets where we included a varying number of corrupted records into the sets of overlapping records (ranging from 0% to 100% corruption, in 20% steps). This means that a certain percentage of records in the overlap were modified for randomly selected parties. Therefore, some of these records are exact duplicates across some parties in a group, but only approximately matching duplicates across the other parties in the group. This simulates for example the situation where three out of five hospitals have the correct and complete contact details (like name and address) of a certain patient, while in the fourth and fifth hospitals some of the details of the same patient are different.

From the created datasets we extracted four attributes commonly used for record linkage: Given name, Surname, Suburb (town) name, and Postcode. These four attributes were used for generating the Bloom filters for the experiments. In our experiments we set the Bloom filter parameters as $m = 1000$ bits, $k = 30$, and $q = 2$ by following earlier Bloom filter based work in PPRL (Schnell et al. 2009).

6.2 Implementation

We implemented a prototype to evaluate the performance of our protocol using the Python programming language (version 2.7.3). We implemented prototypes for single-bit tree construction with a recursive approach and a loop iteration approach. The run time was measured for both of these approaches on different datasets of various sizes. The results showed that the tree construction using the iterative approach runs faster than the recursive approach. So we based all the experiments using the single-bit tree construction on the iterative approach. All the experiments were run on a server with 64-bit Intel Xeon (2.4 GHz) CPU, with 128 GBytes of main memory and running Ubuntu 12.04. The programs and test datasets are available from the authors.

Table 2: Average memory (MBytes) requirement per party.

Data set	BF Construction	Tree construction
5,000	20	23
10,000	33	47
50,000	112	117
100,000	340	370
500,000	970	987
1,000,000	1,920	1,958

We performed a comparison with a phonetic based blocking approach as a baseline to measure the level of privacy provided by our indexing approach. In the phonetic blocking we used Soundex (Christen 2012*b*) as encoding function for the Given name, Surname and Suburb attributes, while for the Postcode attribute the first three digits of a Postcode value are used as the blocking key.

6.3 Evaluation Measures

We evaluated our protocol in terms of complexity, blocking quality, and privacy for different sizes of the dataset, different number of parties, and different block sizes. We measured the runtime for generating Bloom filters and for the single-bit tree construction to evaluate the time complexity of blocking. The reduction ratio (RR) and pairs completeness (PC) are used to evaluate the blocking quality, which are standard measures to assess the efficiency and the effectiveness of blocking (Christen 2012*a*), respectively. The RR and PC are calculated as Equations 5 and 6:

$$RR = 1 - \frac{BL_{CS}}{T_{RS}} \tag{5}$$

$$PC = \frac{BL_{TM}}{T_{TM}} \tag{6}$$

where BL_{CS} is the number of candidate record sets generated by blocking, BL_{TM} is the number of true matching candidate record sets generated by blocking, T_{TM} is the number of total true matching record sets in the datasets, and T_{RS} is the total number of record sets.

According to Vatsalan et al. (2014), a standard and normalized measure to quantify privacy based on simulated attacks (Kuzu et al. 2013) has so far not been studied. Vatsalan et al. (2013*a*) introduced a set of disclosure risk (DR) measures that can be used to evaluate and compare different private blocking and PPRL solutions. To evaluate the privacy of our protocol we use the measure *probability of suspicion* (P_s), which is defined for a value in an encoded dataset as $1/n_g$, where n_g is the number of values in a global dataset (G) that match with the corresponding value in an encoded (protected) dataset D.

This measure of P_s is normalized between 0 and 1, where 1 indicates that the value in D can be exactly re-identified with a value in G based on one-to-one matching, and 0 means a value in D could correspond to any value in G and therefore it cannot be re-identified. Based on the normalized P_s values for each value in D, the *maximum disclosure risk* and the *mean disclosure risk* (Mean) of D can be calculated as the maximum value of P_s ($max(Ps)$) of any value in D, and as the average risk ($\sum(Ps)/|D|$) by considering the distribution of P_s of all values in D, respectively.

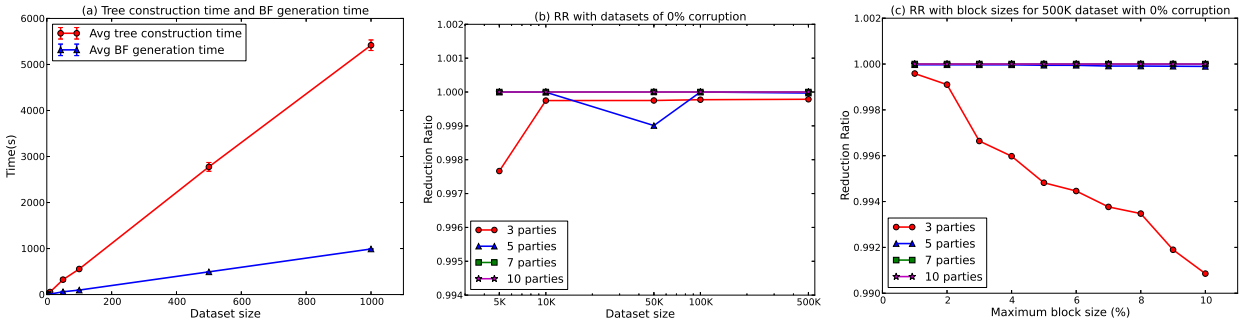


Figure 8: (a) Average time for tree construction and Bloom filter generation, (b) Reduction ratio (RR) with dataset size, and (c) RR with different block sizes.

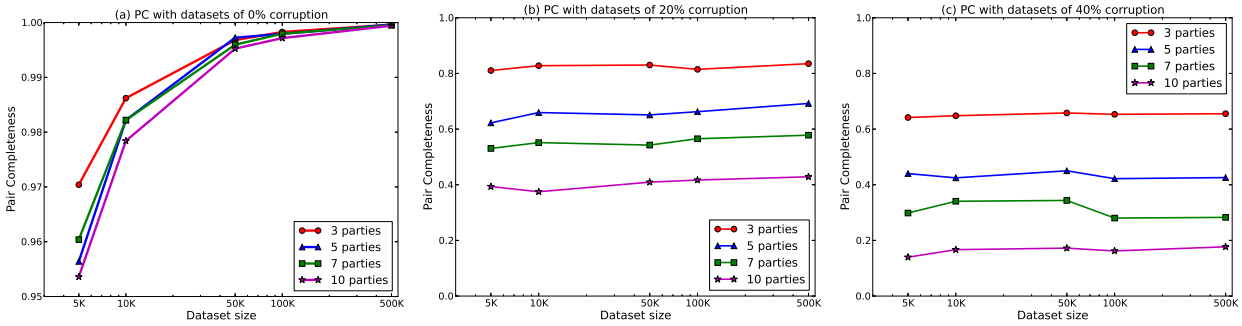


Figure 9: Pair completeness (PC) with dataset sizes for (a) 0% corruption, (b) 20% corruption, and (c) 40% corruption.

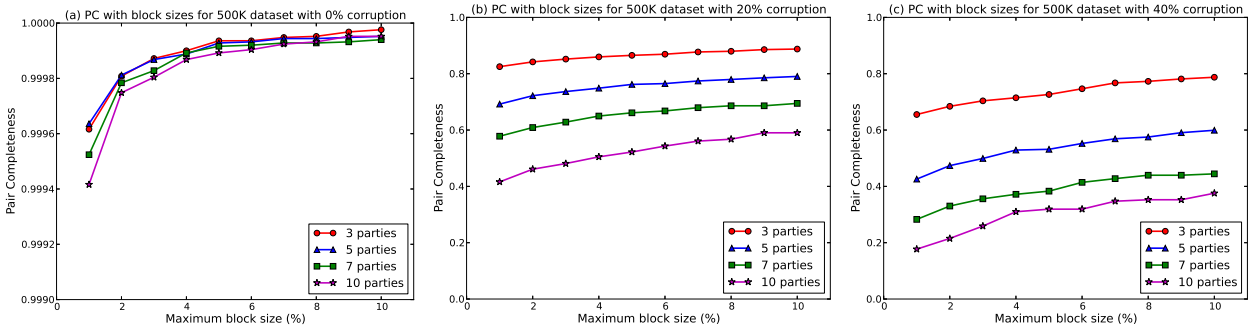


Figure 10: Pair completeness (PC) with different block sizes for (a) 0% corruption, (b) 20% corruption, and (c) 40% corruption.

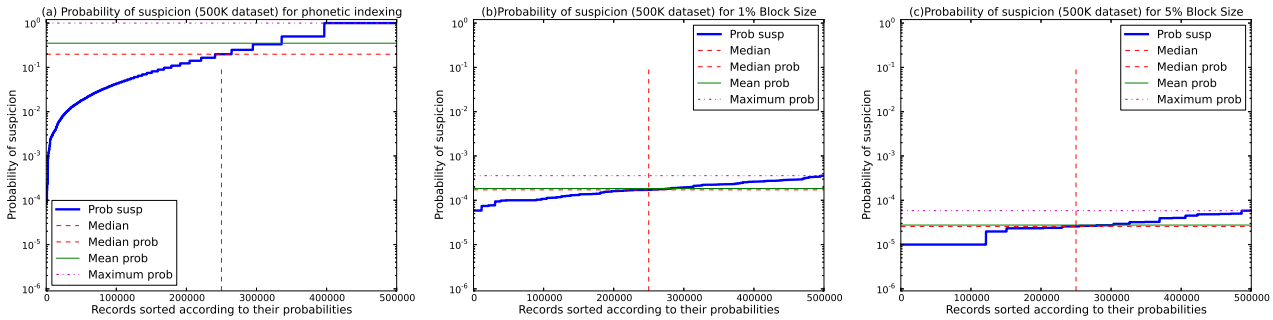


Figure 11: Probability of suspicion (P_s) in the 500K dataset with (a) phonetic indexing, (b) single-bit tree with 1% block size, and (c) single-bit tree with 5% block size.

6.4 Discussion

Figure 8 shows the scalability of our approach in terms of the average time required for generating Bloom filters and the single-bit tree construction time for a single party, and the reduction ratio with different sizes of the dataset and blocks. As expected the tree construction time increases linearly with the dataset sizes. Reduction ratio remains nearly 1 for different dataset sizes and for different number of parties, which illustrates our approach is reducing the

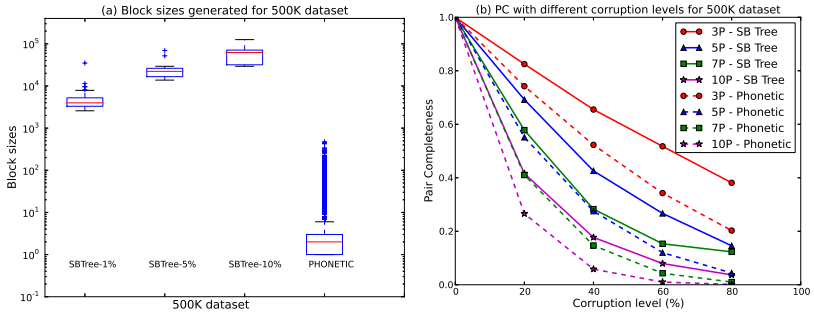


Figure 12: (a) Block sizes generated by phonetic indexing, and single-bit tree with 1%, 5%, and 10% block sizes and (b) Pair Completeness (PC) with different corruption levels for phonetic indexing and single-bit tree.

number of total candidate record sets that need to be compared dramatically. Table 2 shows that our approach can be run with less than 2 GBytes of memory to construct a tree even for one million records.

Figure 9 illustrates the pair completeness (PC) of our approach with different dataset sizes. It shows that PC slightly increases with dataset size (Figure 9(a)), which indicates that the blocks of the single-bit tree contain more true matching records. It is noted that the blocking quality of our approach is affected by the quality of the data and PC is decreasing rapidly with the number of parties with low quality data, as illustrated in Figures 9(b) and 9(c).

As shown in Figure 10, PC increases when the size of the blocks is increased. The block sizes are controlled with the s_{max} parameter, where high PC values are achieved with high s_{max} values. Figures 10(b) and 10(c) illustrate the increment of the PC values with different quality levels of the data for different block sizes.

Privacy is a main aspect of any indexing mechanism in PPRL. We compute P_s values for a single party by assuming all trees contain similar block structures. As shown in Figure 11(b) our approach is having a maximum P_s value less than 0.0001 for each individual, which indicates a record in a block can be matched to more than $10,000$ values in a global dataset (under the worst case assumption of global dataset G is being equal to the linkage dataset). It can be noted that our approach is providing significantly better privacy compared to the phonetic indexing approach that is having a maximum P_s of 1 as shown in Figure 11(a). Figure 11(c) illustrates that better privacy can be achieved with larger block sizes.

We compare our approach with the phonetic indexing in terms of the blocks sizes generated and how the PC is changing with the quality of the data as shown in Figure 12. It shows that the phonetic indexing approach creates a large number of blocks of size 1 (Figure 12(a)). This makes the phonetic based approach not suitable for PPRL, because these records can be exactly re-identified by using a value in G based on one-to-one matching. Figure 12(b) shows that our approach achieves higher PC values than the phonetic indexing even with low quality data.

7 Conclusion

In this paper, we presented a novel indexing protocol for multi-party privacy-preserving record linkage based on Bloom filters and single-bit tree data structures. Each party constructs the single-bit tree index based on the Bloom filters generated on their dataset, and the parties communicate with each other to compute the best bit positions to be used for construction by using a secure summation protocol. The proposed approach was validated by an experimental evaluation, where we performed the experiments on different datasets with a size of up-to one million records. The evaluation results indicated that our approach is scalable with both the size of the databases to be linked and the number of parties. Our approach also outperforms a phonetic indexing approach in terms of privacy and quality of blocking. The blocks which are generated can finally be compared using private comparison and classification techniques to determine the similar record sets in different databases.

We plan to extend our protocol with different tree data structures, which can reduce the number of levels of the trees and by using more bits for the splitting of the tree nodes. We will also investigate the parallelization of the algorithm, which can further improve the performance of our protocol. A limitation in our approach is the assumption of the semi-honest adversary model which is not applicable for some real-world applications. Privacy can be compromised when some parties are malicious which requires more secure communication techniques than the adopted secure summation protocol. We aim to extend this protocol for different adversary models for allowing it to be employed in real-world PPRL applications.

References

Al-Lawati, A., Lee, D. & McDaniel, P. (2005), Blocking-aware private record linkage, *in* 'ACM IQIS', Baltimore, pp. 59–68.

Atallah, M., Kerschbaum, F. & Du, W. (2003), Secure and private sequence comparisons, *in* 'ACM WPES', pp. 39–44.

Bachteler, T., Reiher, J. & Schnell, R. (2013), Similarity filtering with multibit trees for record linkage, Technical report, Working Paper WP-GRLC-2013-02, German Record Linkage Center, Nüremberg.

Bloom, B. (1970), 'Space/time trade-offs in hash coding with allowable errors', *Communications of the ACM* **13**(7), 422–426.

Broder, A. & Mitzenmacher, M. (2004), 'Network applications of Bloom filters: A survey', *Internet mathematics* **1**(4), 485–509.

Cheng, C., Luo, Y.-L., Chen, C.-X. & Zhao, X.-K. (2010), 'Research on secure multi-party ranking problem and secure selection problem'.

Christen, P. (2012*a*), *Data Matching – Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*, Data-Centric Systems and Applications, Springer.

Christen, P. (2012*b*), 'A survey of indexing techniques for scalable record linkage and deduplication', *IEEE TKDE* **24**(9), 1537–1555.

Christen, P. & Vatsalan, D. (2013), Flexible and extensible generation and corruption of personal data, *in* 'ACM CIKM', San Francisco, pp. 1165–1168.

Churches, T. & Christen, P. (2004), Blind data linkage using n-gram similarity comparisons, *in* 'PAKDD', Sydney, pp. 121–126.

Clifton, C., Kantarcioglu, M., Vaidya, J., Lin, X. & Zhu, M. (2002), 'Tools for privacy preserving distributed data mining', *SIGKDD Explorations* **4**(2), 28–34.

de Vries, T., Ke, H., Chawla, S. & Christen, P. (2011), 'Robust record linkage blocking using suffix arrays and Bloom filters', *ACM TKDD* **5**(2).

Durham, E. (2012), A framework for accurate, efficient private record linkage, PhD thesis, Faculty of the Graduate School of Vanderbilt University, Nashville, TN.

Durham, E. A., Toth, C., Kuzu, M., Kantarcioglu, M., Xue, Y. & Malin, B. (2013), 'Composite bloom filters for secure record linkage', *IEEE TKDE* .

Fellegi, I. P. & Sunter, A. B. (1969), 'A theory for record linkage', *Journal of the American Statistical Society* **64**, 1183–1210.

Hawashin, B., Fotouhi, F. & Truta, T. (2011), A privacy preserving efficient protocol for semantic similarity join using long string attributes, *in* 'ACM PAIS', Uppsala, Sweden.

Inan, A., Kantarcioglu, M., Ghinita, G. & Bertino, E. (2010), Private record matching using differential privacy, *in* 'EDBT', Lausanne.

Karakasidis, A. & Verykios, V. (2011), 'Secure blocking+secure matching = secure record linkage', *Journal of Computing Science and Engineering* **5**, 223–235.

Karr, A. F., Lin, X., Sanil, A. P. & Reiter, J. P. (2004), Analysis of integrated data without data integration, Vol. 17, Chance, pp. 26–29.

Kristensen, T. G., Nielsen, J. & Pedersen, C. N. (2010), 'A tree-based method for the rapid screening of chemical fingerprints', *Algorithms for Molecular Biology* **5**(1), 9.

Kuzu, M., Kantarcioglu, M., Durham, E. A., Toth, C. & Malin, B. (2013), 'A practical approach to achieve private medical record linkage in light of public resources', *Journal of the American Medical Informatics Association* **20**(2), 285–292.

Lai, P., Yiu, S., Chow, K., Chong, C. & Hui, L. (2006), An efficient Bloom filter based solution for multiparty private matching, *in* 'International Conference on Security and Management'.

Lindell, Y. & Pinkas, B. (2009), 'Secure multiparty computation for privacy-preserving data mining', *Journal of Privacy and Confidentiality* **1**(1), 5.

Mitzenmacher, M. & Upfal, E. (2005), *Probability and computing: Randomized algorithms and probabilistic analysis*, Cambridge University Press.

O'Keefe, C. M., Yung, M., Gu, L. & Baxter, R. (2004), Privacy-preserving data linkage protocols, *in* 'ACM WPES', Washington DC, pp. 94–102.

Rashid, S., Brijesh, K. & Mishra, D. K. (2009), Privacy preserving k-secure sum protocols, *in* 'Computer Science and Information Security', Vol. 6, pp. 40–46.

Ravikumar, P., Cohen, W. & Fienberg, S. (2004), A secure protocol for computing string distance metrics, *in* 'PSDM held at IEEE ICDM', Brighton, UK, pp. 40–46.

Scannapieco, M., Figotin, I., Bertino, E. & Elmagarmid, A. (2007), Privacy preserving schema and data matching, *in* 'ACM SIGMOD', pp. 653–664.

Schnell, R., Bachteler, T. & Reiher, J. (2009), 'Privacy-preserving record linkage using Bloom filters', *BMC Medical Informatics and Decision Making* **9**(1).

Song, D., Wagner, D. & Perrig, A. (2000), Practical techniques for searches on encrypted data, *in* 'IEEE Symposium of Security and Privacy', Oakland, pp. 44–55.

Sweeney, L. (2002), 'K-anonymity: A model for protecting privacy', *International Journal of Uncertainty Fuzziness and Knowledge Based Systems* **10**(5), 557–570.

Vatsalan, D. & Christen, P. (2012), An iterative two-party protocol for scalable privacy-preserving record linkage, *in* 'AusDM, CRPIT 134', Sydney, Australia.

Vatsalan, D., Christen, P., O'Keefe, C. M. & Verykios, V. S. (2014), 'An evaluation framework for privacy-preserving record linkage', *Journal of Privacy and Confidentiality* **6**(1), 35–75.

Vatsalan, D., Christen, P. & Verykios, V. S. (2013*a*), Efficient two-party private blocking based on sorted nearest neighborhood clustering, *in* 'ACM CIKM', San Francisco, pp. 1949–1958.

Vatsalan, D., Christen, P. & Verykios, V. S. (2013*b*), 'A taxonomy of privacy-preserving record linkage techniques', *Elsevier Journal of Information Systems* **38**(6), 946–969.

Yakout, M., Atallah, M. & Elmagarmid, A. (2012), 'Efficient and practical approach for private record linkage', *ACM JDIQ* **3**(3), 5.

Yao, A. (1986), How to generate and exchange secrets, *in* 'Foundations of Computer Science', IEEE, pp. 162–167.

Detecting Digital Newspaper Duplicates with Focus on eliminating OCR errors

Yeshey Peden
ICT Officer
Department of Public Accounts
Ministry of Finance
Thimphu, Bhutan
ypgyeltshen@gmail.com

Richi Nayak
Associate Professor
Higher Degree Research Director
Science and Engineering Faculty
Queensland University of Technology
r.nayak.qut.edu.au

Abstract

With the advancement in digitalization, archived documents such as newspapers have been increasingly converted into electronic documents and become available for user search. Many of these newspaper articles appear in several publication avenues with some variations. Their presence decreases both effectiveness and efficiency of search engines which directly affects user experience. This emphasizes on development of a duplicate detection method, however, digitized newspapers, in particular, have their own unique challenges. One important challenge that is discussed in this paper is the presence of OCR (Optical Character recognition) errors which negatively affects the value of document collection. The frequency of syndicated stories within the newspaper domain poses another challenge during duplicate/near duplicate detection process. This paper introduces a duplicate detection method based on clustering that detects duplicate/near duplicate digitized newspaper articles. We present the experiments and assessments of the results on three different data subsets obtained from the Trove digitized newspaper collection.

Keywords: clustering, duplicate document detection, OCR errors, feature selection

1 Introduction

The Australian Newspaper Digitization Program (ANDP) has initiated the digitization of newspapers archives before the copyright act. ANDP has provided free access to this data, relevant to Australia via the Trove search engine[1]. A significant problem with digitisation of archived newspapers is the presence of identical and nearly identical documents in the resulting collection. These duplicate documents are not only an annoyance to users as search results, but, they also decrease efficiency of search engines (Uyar, 2009). The processing of these duplicate documents and results is not only time consuming, but, their presence also does not add any value to the information presented to users. Duplicate document detection has gained research interest as it assists in search engines in increasing the effectiveness and storage efficiency (Uyar, 2009). While carrying out this task, consideration of the domain/application is very important. For example, in plagiarism detection scenario, even if a sentence or a paragraph of one document is found in another document the two documents could be seen as near-duplicates. This definition of near-duplicate may be looser in other domains as comparison to news domain (Hajishirzi, Yih and Kolcz, 2010).

Newspaper articles exhibit distinct characteristics (Smeaton, Burnett, Crimmins and Quinn's, 1998). Almost all news stories have an element of a continuum and news stories usually evolve over time. A large proportion of articles are related to previously published articles. There exists a dependency between news stories which cannot be ignored when navigating news archives. These dependencies need to be acknowledged while looking for duplicate newspaper articles. If two documents are identified as near identical articles, when in fact one contains new information, this potentially leads to loss of information (Gibson, Wellner and Lubar, 2008). In this paper, we explored document clustering as a technique to identify and generate links between related news stories.

Another significant problem with the use of digitized collections in search engine is OCR (Optical Character recognition) errors. These errors are non-existing words which result due to incorrect letter recognition. OCR helps resolve the growing problem of searching intended information from the digital archives; however, historical newspapers are different from the recent newspapers in image quality, type fonts, ruby characters, noise, and language usage. The direct use of these digitized collections, without any processing for quality improvement, cannot provide satisfactory results (Shima, Terasawa and Kawashima, 2011). Newspapers especially old ones are certain to have OCR errors no matter how well the digitization is done. The reasons include the

[1] http://trove.nla.gov.au/

condition of the source material, the typeface used, the contrast between print and paper and many more. (DLC, 2011).Whereas such challenges do not surface in other web documents,

An abundance amount of work exists on identifying duplicated online web pages (Conrad, Guo and Schriber, 2003; Radlinski, Bennett and Yilmaz, 2011), but, none can be found on digitized newspapers with particular consideration to OCR errors. While identifying exact duplications of plaintext documents has been more straight forward (Gibson, Wellner and Luber, 2008), identifying near duplicates has been challenging (Alonso, Fetterly and Manasse, 2013). The task of defining near duplicates and setting a "resemblance" threshold (a threshold that classifies near identical and non-identical documents) is difficult and varies across the domains. The errors resulted due to OCR software application not only have the potential to affect the clustering solutions but it also makes it difficult to set a threshold and identifying redundancy.While it will be almost impossible to detect exact duplicates, it will be most difficult to decide at what level of threshold to discard the near duplicates. At times, the small differences between the two near duplicate documents might have important information to reveal. Moreover two documents might be near duplicate but a lot of words might not match due to OCR errors. This will result in falsely classifying the documents as not duplicates.

This paper is a step towards finding the solution to the problems associated with digitized newspaper articles. We propose a solution to deal with unique problems such as eliminating OCR errors and detecting redundant news stories which may negatively affect the search engine performance and the user experience. We propose a methodology using a pre-processing method to eliminate OCR errors and clustering algorithms. Evaluation of the clusters is done using both internal and external measures. The proposed method is based on the assumption that the closest neighbours within a cluster can represent near duplicates or exact duplicates.

The remainder of this paper is set out as follows. Section 2 will describe the approach taken. This is followed by experiments, results and evaluation in section 3 and finally ending with the conclusion.

2 The Proposed Clustering based Method

The proposed document duplication detection method starts with initial pre-processing that includes removal of all non-alphanumeric characters and stop words (i.e. extremely common words which are not valuable) and stemming of the terms to its root form using Porter's stemmer (Porter, 1980). This is followed by identification of the more common words and OCR error using a simple method consisting of frequency count and threshold cut-off. A sample of raw data from Trove dataset displayed in Figure 1 contains non-existing words such as "luetnc", "uecrioi", "gnnitttt" and "oiilitioiiali" which resulted due to incorrect letters.

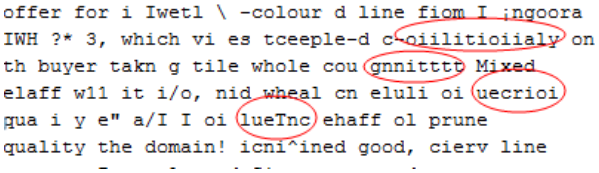


Figure 1: A sample of raw data from Trove dataset prior to cleaning

For each term in the document collection, Term Frequency (TF) is calculated. TF is the number of times the term occurs in the document collection D.

TF (t, D) = f (t, D)

Selection of a threshold cut-off is critical. Setting a higher threshold may lead to elimination of valuable words. Setting the threshold too low may still leave the noise or OCR errors in the collection. It is not possible to get a perfect threshold cut-off that leads to elimination of all the noise while keeping all the important words. The aim is to find a cut-off that might not lead to complete elimination of the OCR errors but, it should remove most of the common ones, and it should also lead to a reduced word space while keeping the important terms. This will reduce the computational time and, additionally, lead to a more accurate clustering solution.

The decision on the threshold cut-off is made by plotting the frequency distribution of the terms. Since the number of terms is too big to be put in one plot, random points are picked for visualization shown in Figure 2. The x-axis of the graph represents the number of times (n) a term occurs within the document collection and the y-axis shows the terms that occurs n times. It is seen that a total number of 842385 terms occurs one time and also the maximum count of times a term occurs is 35,717.

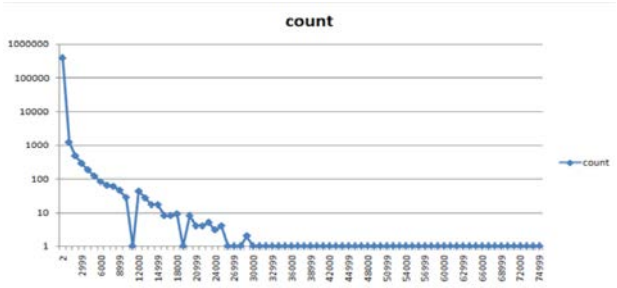


Figure 2: A graph presenting frequency distribution of terms-random points picked due to largeness of data

It can been seen in Figure 3 that the words identified as results of OCR error in Figure 2 actually occur only one time in the entire document collection. Such rarely occurring terms are usually filtered out by setting a correct threshold.

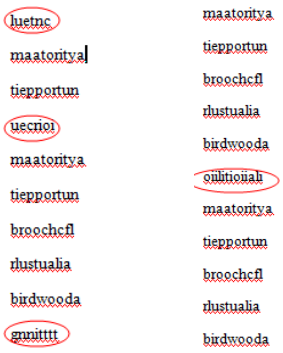


Figure 3: A sample list of stemmed terms occurring one time in the complete document collection

Once pre-processing is completed the document collection is represented in the Vector Space Model (VSM) which represents documents as a feature vector of the words that appear in the documents of the collection (Jin, Chai and Si 2005). Each feature vector is presented with the term-frequency * inverse-document-frequent (tf * idf) weighting. This weighting will consider a term important if it occurs frequently within a document and does not occur so frequently in the collection. This model becomes input to the clustering process.

A choice of clustering algorithm depends upon many factors. Experimental studies in the literature have often portrayed hierarchical clustering to be better than the partitional K-means, in terms of clustering quality but inferior in terms of time complexity (Steinbach, Karypis and Kumar, 2000). In this paper, we apply the repeated bisections partitional clustering approach (Karypis, 2002) to deal with the large dimensions. The method is a K-way clustering solution that is computed by performing a sequence of K-1 repeated bisection on the data instances where K is the desired number of clusters. In addition to having low computational requirements, this bisecting K-means approach has a time complexity which is linear in the number of documents. It is found that as K increases, there is an increment in the optimization of the criterion functions as well.

Using a nearest neighborhood method that uses cosine similarity, for each object (a newspaper article) within a cluster, its ten nearest neighbors is found. If the similarity between the object and its nearest neighbor is within the acceptable threshold, it is considered as identical or near-identical. It is assumed that any near duplicates of that particular document should be its nearest neighbour within the same cluster.

3 Experiments and Evaluation

This section focuses on assessing the effectiveness of the proposed document duplication detection method. We first present the datasets and evaluation measures used in experiments. The results from the pre-processing phase of the experiment will be presented next. This is followed by discussion and analysis of the results obtained from the clustering phase. Finally, we will present and analyze the results of duplicate identification obtained from the cluster evaluation phase.

We use the clustering tool CLUTO to perform clustering (Karypis, 2002). CLUTO is chosen because it is easy to use and is able to operate on very large dataset with respect to number of documents as well as number of dimensions. CLUTO has many clustering algorithms and each of these algorithms has different scalability characteristics. Karypis (2002) has shown that the most scalable method in terms of time and memory is vcluster's repeated-bisecting algorithm that uses cosine similarity function.

3.1 Dataset

The Trove data set consists of a total number of 77,841,027 documents. The documents fall into six categories which are: Literature; detailed lists, results and guides; Advertising; Articles; Family Notices; and Others. The main focus of the study is on the Article category which consists of a total of 58,549,810 documents (as shown in Table 1). The experiment is conducted on three different data subsets selected from three different time period of year 1921 (called as Dataset 1), year 1880 (called as Dataset 2) and year 1862 (called as Dataset 3). The data collection for the period of 1921 is randomly chosen while the years 1862 and 1880 collection are chosen based on queries identified by historians as having duplicates. It is based on the fact that there are duplicates of those particular news articles. For example, running the search query "kipper billy daring attempt" on Trove website is supposed to produce at least three duplicate copies. For each dataset, a two months of data snapshot has been used for clustering experiments.

Category Name	**#Documents**
Literature	12289
Detailed Lists, Results, Guides	7516250
Advertising	11058829
Article	**58549810**
Family Notices	703846
Others	3
Total:	**77841027**

Table 1: Total number of documents under each category

The study is based on the premise that, in archived news-stories in late 1800s and early 1900s, the duplication in news occurs within the two months of the release of the original news story and not beyond that. This assumption has been made based on discussion with historians involved in the project, as well as, based on the facts that the stories used to deliver via telegraphs and printing and delivery process used to take long. Today with partial replacement of newspapers by web news and network television news which gets updated in real time, most duplication would happen on the same day. Even with newspapers, most publication happens daily or weekly. A search completed on Trove search engine for both queries "Kelly expiated his career in crime" and "kipper billy daring attempt" showed that news duplication does not go beyond two months. These two queries were for news that happened in 1860 and 1862 respectively.

In the original data collection, the news articles include some advertisement. There was no heuristic way to identify these ads and distinguish them with the news stories. A simple method is applied to identify and eliminate any ads occurring within the data collection. An initial clustering is conducted on the data collection after

a simple pre-processing of stemming and general stop word removal. Once the clusters are obtained, the best clusters on the top are manually observed. It has been seen that any cluster that has few number of documents and that has very high intra-similarity tends to be ads most of the time. The exact same ads might repeat in several newspapers and appear during extended periods. This might not allow efficient identification of duplicate news articles. Any document within these clusters which is identified as an ad is removed from the collection. This process slightly improves the efficiency of the clustering solution.

3.2 Evaluation Measures

The quality of the clusters obtained is measured using two measures: internal quality measure and external quality measure. The internal quality measure is based on the examination of the internal similarity (ISIM) that is the average similarity between the objects of each cluster and the output value of the criterion function used. The ISIM has values between 0 and 1. Good clusters have ISIM and output value of the criterion function closer to 1. In contrast a bad cluster has ISIM and output value of the criterion function closer to 0. This is also known as the quantitative measure.

In addition to the internal measures, an alternative qualitative measure is required to evaluate the effectiveness of the clustering algorithm in grouping the similar digitized news articles together. This leads to finding out how successfully the near duplicate articles are detected. An extrinsic method using ground truth is applied. The ground truth for this study is a collection of information retrieved from the Trove search engine based on certain queries. A simple process of manual observation is applied to the results retrieved by the query in order to identify duplicates related to that query. Then the cluster solution is taken for further evaluation.

3.3 Effectiveness of pre-processing

This section discusses the effect of pre-processing phase in removing the OCR errors and reducing word space. An assumption is made that if a terms occurs too many times in the overall document collection, the term is not important and does not provide any valuable information while if a terms occurs very few times such as once, the term is an OCR error term. This conclusion is drawn from manual observations of the data. A cut-off threshold is set to identify too frequent and too infrequent terms.

Experiment on Dataset 1 (Table 2) shows that a number threshold is better suited than a % threshold even though it is not a perfect solution by itself. A % threshold leads to elimination of too many terms which could lead loss of valuable words. It has been seen that increasing the threshold cut-off leads to maximum elimination of OCR errors but it also increases the risk of removing important terms.

Input file	Min & Max Threshold culled	Matrix out put (doc, term, matrix density)
Original data collection	Max:8000, Min:1	115755, 398858, 12249111
Data collection: ads removed	Max:8000, Min:1	115635, 398826, 12482467
Data collection: ads removed	Max:8000, Min:2	115635, 229043, 12178014
Original data collection	MIN:1% ; MAX:70%	115755, 2408, 684543
Original data collection	NONE	115755, 1869496, 13861605

Table 2: Dataset 1(1921) - Comparison of different versions of the data collection with different threshold cut offs

Observation of Table 2 shows that it does reduce the word space of the document collection, but, the reduction is not too great. Moreover comparing the threshold cut offs MIN=1 and MIN=2, the latter proves to be a better choice than the former. The “Min” indicates how many minimum documents the term should appear in, and the “Max” shows how many maximum documents, the term should appear in. Note that this processing is conducted after the standard stop-word removal and stemming so the dataset does not contain many stop-words or rare words. This culling would ensure that errors associated with OCR are removed.

It can be seen from Table 3 that the total number of documents in the 1880 collection is 28717. The original data without any MIN and MAX threshold culled has 1045656 unique terms and the density is 7340510. By eliminating the most frequent and infrequent (OCR errors) terms from the original data using a simple MIN and MAX threshold cut off, the number of unique term has reduced to 130579 and the dimensionality has also been reduced.

Input Data	MIN & MAX Threshold culled	Matrix out put
Original data	Max:8000, Min:2	28717 x 130579 x 6274642
Original data	None	28717 x 1045656 x 7340510

Table 3: Dataset 2 (1880) - Comparison of the original input data with the data from which the frequent and infrequent terms (OCR errors) have been removed

Similar results can be seen for Dataset 3 in Table 4.The identification and elimination of approximately 87.49% of total number of terms in the 1880 data collection (as shown in Table 5) and 86.79% of total number of terms in the 1862 data collection (as shown in Table 6), as infrequent/OCR errors, confirm the concern stated earlier with regard to OCR errors. Figure 4 displays the top ten OCR errors in the three different datasets that are removed by our method.

Input File	Min & Max Threshold culled	Matrix out put (doc, term, density)
Orginal Data	Max:8000, Min:2	16696, 103172, 4347401
Orginal Data	None	16696, 782569, 5137437

Table 4: Dataset 3 (1862) - Comparison of the original input data with the data from which the OCR errors have been removed

Total No of unique terms	MIN Threshold	Terms occuring<=MIN Threshold	Terms occuring more than 8000 times	Good usable terms
1045656	2	914886=87.49%	1	130769=12.51%

Table 5: Dataset 2 (1880) - The most frequent, infrequent and good usable terms

Total No of Unique terms	Threshold	Terms occuring<=Threshold	Terms occuring more than 8000 times	Good terms
782569	2	679221=86.79%	1	103347=13.21%

Table 6: Dataset 3 (1862) - The most frequent, infrequent and good usable terms

jrticiefl	pretiid	1t1l
bxohaagw	pretiic	gehmhnh
flezv	flev	vollkommerr
uildltion	1roiu	flez
flg5	flet	icliarfca
iimurbvd	tenipemtui	ptcherd
mfksr	amenlldmenlldmenltmenlld	riwuh
asrest	fler	ttfdrutahd
supuppliosup	sansllst	flep
lrrlgate	flel	sanult
Dataset 1 (1921)	Dataset 2 (1880)	Dataset 3 (1862)

Figure 4: Top 10 OCR errors removed by the defined process in all three datasets respectively

3.4 Internal Evaluation of Cluster Solutions

For Dataset 1, it is seen that the best clusters were produced by a combination of K=2000, method=RBR and criterion function=I2 (as shown in Table 7 and 8). For Dataset 2, Table 9 and 10 shows that combination of K=600, method=RB and criterion function=I1 produced a better solution compared to the rest. For Dataset 3, it is clear from results in Table 11 and 12 that K=400 produces a better clustering solution compared to K=300. Looking at the methods, if based on the output value of the criterion function, RBR methods combined with criterion function I2 outperformed the other combinations. Otherwise if based on the average ISIM value, RB with I1 seems to be performing better than the rest.

Although some combination of the methods outperformed the other, the quality of overall clustering solution is not very high. For a good solution, it is expected that the average ISIM value and the output value of the criterion function at least be 0.5 and above. These values are obtained just around 0.5. An external measure based on ground truth is required to evaluate the solution further and discover the effectiveness of the methodology.

K value	MIN Threshold Cut off	No of Docs Clustered	OUTPUT
100	1	115630 of 115635	I2=2.68e+04
100	2	115630 of 115635	I2=2.74e+04
1000	1	115630 of 115635	I2=4.21e+04
1000	2	115630 of 115635	I2=4.28e+04
2000	1	115630 of 115635	I2=4.79e+04
2000	2	115630 of 115635	I2=4.88e+04

Table 7: Dataset 1, cluto clustering solution for method=rbr; -sim (similarity measure)=cosine; criterion function=I2

Method	Criterion Function	SIM	Output
RB	I2	Cos	I2=3.75e+04
RBR	I2	Cos	I2=4.80e+04
RB	I1	Cos	I1=1.39e+04
RBR	I1	Cos	I1=1.68e+04

Table 8: Dataset 1, comparison of two different methods and two criterion functions in CLUTO using K=2000

Method	Criterion Function	SIM	OUTPUT	AVG no of Docs in each cluster	ISim(AVG)
RB	I2	Cos	I2=1.04e+04	57.428	0.17413
RBR	I2	Cos	I2=1.1202e+04	57.428	0.17938
RB	I1	Cos	I1=4.35e+03	57.428	0.329976
RBR	I1	Cos	I1=4.88e+03	57.428	0.31893

Table 9: Dataset 2, result summary for K=500

Method	Criterion Function	SIM	OUTPUT	K(AVG)	ISim(AVG)
RB	I2	Cos	I2=1.08e+04	47.85667	0.181992
RBR	I2	Cos	I2=1.16e+04	47.85667	0.190278
RB	I1	Cos	I1=4.64e+03	47.85667	0.345382
RBR	I1	Cos	I1=5.20e+03	47.85667	0.327673

Table 10 : Dataset 2, result summary for K=600

Method	Criterion Function	SIM	OUTPUT	AVG no of Docs in each cluster	ISim(AVG)
RB	I2	Cos	I2=6.29e+03	55.65	0.192987
RBR	I2	Cos	I2=6.64e+03	55.65	0.193087
RB	I1	Cos	I1=2.78e+03	55.65	0.34794
RBR	I1	Cos	I1=3.01e+03	55.65	0.335223

Table 11 : Dataset 3, result summary for K=300

Method	Criterion Function	SIM	OUTPUT	AVG no of Docs in each cluster	ISim(AVG)
RB	I2	Cos	I2=6.63e+03	41.7375	0.2023825
RBR	I2	Cos	I2=7.00e+03	41.7375	0.207855
RB	I1	Cos	I1=3.05e+03	41.7375	0.345963
RBR	I1	Cos	I1=3.32e+03	41.7375	0.337478

Table 12: Dataset 3, result summary for K=400

3.5 External Evaluation of Cluster Solutions

The three datasets are further evaluated using an extrinsic method. The quality of cluster is evaluated by finding out how successfully the near duplicate articles can be detected using the cluster outputs. Analysing the results obtained using Dataset 1 (as shown in Table 13), it is established that when the cosine value is equal to 1, two documents are exact duplicates and when the cosine value is below 1 and above 0.4, the documents could be near duplicate.

Document ID	Cosine Value
83949238	1
70768975	0.602139
66634151	0.589655
16882523	0.58184
51105364	0.578736
27953167	0.568314
79392563	0.496901
80494852	0.484702
92886510	0.481333
20456292	0.474761

Table 13: Dataset 1 - Nearest neighbors of document id 83949238

Human observation shows that all 10 nearest neighbours obtained for that document can be called near duplicates but the low cosine values resulted because of the OCR errors, and due to the fact that the same information could have been presented using different words.

We have not yet used a semantic model to incorporate the semantic words in the matching process. With Dataset 2, based on results from Table 14, it is understood that when cosine value is 1 or 0.9999, the document is an exact duplicate and the documents are near duplicate if the cosine value is 0.8 and above. The documents are more of an update on the news if the threshold is 0.4 and above. It is seen that that better observation can be made from dataset 2 due to presence of less OCR errors.

Document ID	Cosine value	Comments
13476046	1	exact duplicate
13483531	0.957245	near duplicate
13477812	0.945398	near duplicate
78918195	0.90304	near duplicate
813385	0.842917	near duplicate
65379447	0.54443	contains information about Ned Kelli's execution but it is more of an update of the news with more details
65960937	0.50208	contains information about Ned Kelli's execution but it is more of an update of the news with more details
77590206	0.492915	contains information about Ned Kelli's execution but it is more of an update of the news with more details.
89686423	0.479583	contains information about Ned Kelli's execution but it is more of an update of the news with more details
2984082	0.44524	there is near duplicate information but because of the use of different terms, due to OCR errors and due to presence of other news, cosine similarity value is low

Table 14: Dataset 2 -Nearest neighbors of document id 13476046

With Dataset 3, from the results in Table 15 it is found to be difficult to set a cosine value threshold for identifying near/exact duplicates. For a particular document, human observation proved that some of its nearest neighbours obtained are near duplicate information. But with the presence of OCR errors, the cosine value is calculated lower than it should be.

Document ID	Cosine Value	Comments
4604263	1.0	
13225704	0.72897	Near duplicate. Could have had higher cosine value if other news stories were not merged in it.
59790774	0.665777	Near duplicate but cosine value is lower because of too many OCR errors
13225448	0.319914	Near duplicate but it just contains a shortsummary news. No details given
18687137	0.294292	Near duplicate. Could have had higher cosine value if other news stories were not merged in it
4604821	0.291291	Related news but not near duplicate
18687088	0.249018	Contains a brief summary of the news. No details and other news stories are merged in it
60509989	0.221332	Contains a brief mention of the kipper billy news(update) but because of the OCR errors and also it contains other news stories merged in it
79976917	0.106073	Similar news but not related
90253966	0.090444	Completely different news

Table 15: Dataset 3 - Nearest neighbours of document id 4604263

While the methodology has been successful, to some extent, in detecting near/exact digitized news documents, it is clear that the pre-processing phase could have been improved further. The pre-processing phase did not completely eliminate the OCR errors and no consideration was given to incorporate semantic models to deal with synonyms and hyponyms. These flaws had impact on the clustering solution and therefore clear observation could not be made at the end. Anyway this project was a first step forward and, with more sophisticated pre-processing, improved results can be obtained.

4 Conclusion

This paper presented a clustering based duplicate detection method for digitized newspaper stories. Three different data subsets have been employed to test the method. Correction of OCR errors has been found a significant issue dealing with digitized collection of the archived documents. Employing a simple method of threshold cut-off to eliminate the OCR errors has not been found to be a perfect solution. While for each dataset, some combination of the methods outperformed the other, the overall clustering solution was not found to be optimal. This would have been due to the reason that there has not been 100% removal of OCR errors from the digitized news article collection. The risk of removing important terms prevented the removal of 100% of the errors. It is found that increasing the number of clusters leads to better quality clusters but it has also been established that increasing the value of the cluster number beyond a certain point leads to clusters containing very few documents which does not serve the purpose of keeping all similar news documents together.

The clusters were further evaluated by using an extrinsic nearest neighborhood method that was used to find the 10 nearest neighbor for each document/object in a cluster.

In future, to make the duplicate detection more robust and efficient, a query dependant method can be explored and adapted. More time can be spent on the pre-processing phase of the methodology to find a way to eliminate the OCR (Optical Character recognition) errors without removing any valuable terms. The simple threshold cut off method can be improvised into a more sophisticated method. Likewise for dimensionality reduction, the same simple method was used. Other methods such as the PCA

(Principal Component Analysis) (Indhumathi and Sathiyabama, 2010) and SVD (Singular Value Decomposition) can be explored and experimented with as well.

It would also be important to look into other clustering methods. The methodology developed in this study adapted the bisecting partitional approach which does not consider a new incoming document. Methods such as adaptive K-means clustering allow clusters to grow without depending on the initial selection of cluster representation. It would be interesting not only to explore the different software, tools and methods but it will also be interesting to expand the study to understand the evolution of news stories and enhance user search experience.

5 Acknowledgement

We would like to acknowledge CRC Smart Services and Prof Kerry Raymond for facilitating data acquisition from National Library of Australia. We acknowledge Prof Paul Turnbull, Dr Sangeetha Kutty, Sharon Pingi and Behzad for constructive discussion and assistance in processing the data. Yeashey will like to thank AusAID scholarship and AusAID student contact officers in QUT for their full support.

6 References

Alonso, O., Fetterly, D. and Manasse, M. (2013): Duplicate news story detection revisited. In *Information Retrieval Technology*: 203-214, Springer Berlin Heidelberg.

Conrad, J. G., Guo, X. S., and Schriber, C. P. (2003, November): Online duplicate document detection: signature reliability in a dynamic retrieval environment. In *Proceedings of the twelfth international conference on Information and knowledge management*: 443-452, ACM.

DLC (2011, May). A brief history of tactile writing systems for readers with blindness and visual impairments.http://www.dlconsulting.com/digitization/the-unique-challenges-of-newspaper-digitization.

Gibson, J., Wellner, B., and Lubar, S. (2008): Identification of Duplicate News Stories in Web Pages. In *Workshop Programme*: 26.

Hajishirzi, H., Yih, W. T., and Kolcz, A. (2010, July): Adaptive near-duplicate detection via similarity learning. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*: 419-426, ACM.

Indhumathi, R., and Sathiyabama, S. (2010): Reducing and Clustering high Dimensional Data through Principal Component Analysis. *International Journal of Computer Applications*, 11:8.

Jin, R., Chai, J. Y., and Si, L. (2005, August): Learn to weight terms in information retrieval using category information. In *Proceedings of the 22nd international conference on Machine learning*: 353-360, ACM.

Karypis, G. (2002): CLUTO-software for clustering high-dimensional datasets. http://gloros.dtc.umn.edu/gkhome/cluto/cluto/download.

Porter, M. F. (1980): An algorithm for suffix stripping. *Program: electronic library and information systems*, *14*(3): 130-137.

Radlinski, F., Bennett, P. N., and Yilmaz, E. (2011, February): Detecting duplicate web documents using click through data. In *Proceedings of the fourth ACM international conference on Web search and data mining*: 147-156, ACM.

Shima, T., Terasawa, K., and Kawashima, T. (2011, September): Image processing for historical newspaper archives. In *Proceedings of the 2011 Workshop on Historical Document Imaging and Processing*: 127-132, ACM.

Smeaton, A. F., Burnett, M., Crimmins, F., and Quinn, G. (1998, March): An architecture for efficient document clustering and retrieval on a dynamic collection of newspaper texts. In *BCS-IRSG Annual Colloquium on IR Research*.

Steinbach, M., Karypis, G., and Kumar, V. (2000, August): A comparison of document clustering techniques. In KDD workshop on text mining, Vol. 400: 525-526.

Uyar, E. (2009): *Near-duplicate news detection using named entities* (Doctoral dissertation, BILKENT UNIVERSITY).

The Schema Last Approach to Data Fusion

Neil Brittliff and Dharmendra Sharma

University of Canberra

Faculty of Education, Science, Technology and Mathematics

Dharmendra.Sharma@canberra.edu.au

Abstract

Big Data presents new challenges that require new and novel approaches in order to resolve the problems associated with the variability and variety of data obtained from multiple sources. This paper focuses on how to manage variety and the eclectic nature of big data using a technique known as 'Schema Last'. The 'Schema Last' approach is a frame work which defers the application of a descriptive model until it is required. This paper also provides a formal definition of the 'Schema Last' methodology and demonstrates the effectiveness over the more traditional Extract-Transform-Load methodologies employed in many organizations. The 'Schema Last' approach can be used as input to Map Reduction, Index creation and various data mining techniques. Ultimately, the Schema Last approach provides the frame-work to 'fuse' semi-structured data into a single coherent view.

1 Introduction

The Australian Crime Commission (ACC) established the Fusion Data Section in 2010. The section was tasked to collect and analyze data. Legislative powers were granted to the ACC that enables data to be obtained under their coercive powers. These powers do not extend to the format or structure of the data.

The approach taken by the Australian Crime Commission was to model a variety of data sets received via the data collection processes. The ACC could then utilize the collected data for the following purposes:

- assess the behavior of known entities.
- determine any new leads based on the behavior of known entities.
- develop models that may indicate criminal activity within the Australian community.

The *schema last approach* (SLA) is a technique to model data contained within a Big Data store. This approach allows data models to be specified on an on-demand basis and as new knowledge became available, this can be used to respecify the schema definitions. This approach only works if the schema's specification is independent to the data storage. The SLA addresses the issues raised by [Klaus-Dieter Schewe, 2013] in 2012 concerning data quality, interpretation and modification of data as a result of the *cleansing* process.

The size of data sets received by the Australian Crime Commission can range from small to extremely large. Initially, *data cleansing* or the transformation process as part of Extract-Transform-Load (ETL) was the approach taken to convert the raw data into a sanitized form that was loaded into relational tables.

The value proposition for each data source can be quite different. For example, data sets may be classified as low or high signal data sources. Low signal data sources in themselves do not provide any useful indicators but could be used to confirm an entity's address, data of birth and property ownership. An entity membership of low signal data source in itself is not a form of intelligence. High signal data sources can be further analyzed and an entity's membership may indicate a potential threat or unlawful activity that would require further analysis. An example of low and high signal data for attendance at a fairground would be residents of town compared to the entry list of partons.

As the number of data sources continued to increase the ACC came to the realization that the existing ETL process was taking too long and therefore delaying the time taken to analyze the data. Therefore, a new approach had to be found to reduce the demands placed upon the Fusion Data Centre.

1.1 Motivation

The initial implementation utilized a normalized relational table structure. This structure was specifically designed to capture the following *entity* information that included:

- person: name, date-of-birth, and address details.
- organization: organization name, address, Australian business number (ABN), and Australian Company Number (ACN).
- information pertaining to the relationship between two or more entities.

Any data item that did not fall within the structure was discarded and therefore the model could only capture simple identity information. In addition, transaction, time series data or highly linked data could not be captured within this structure even though most relational database products allow for the modification of existing table structures.

There were instances where data definitions within the schema did not adequately describe the source data. In this case a person within the Fusion Data

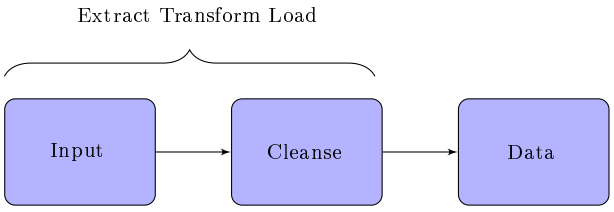


Figure 1: Data Cleansing

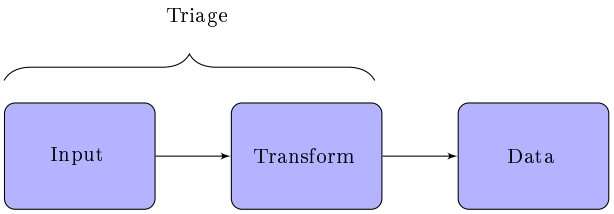


Figure 2: Data Triage

Centre would determine the most appropriate field within the relational structure and perhaps alter the raw data item to comply to the field's schema definition.

This whole entire data ingestion process was seen by the management of the fusion data centre as cumbersome and error prone. Therefore an alternate solution had to found to replace the existing ingestion process.

1.2 Data Triage and ETL

Data Cleansing is the transformation of data from a non-canonical to a canonical state. ETL involves the transformation of data into a state suitable for ingestion into a database. This usually requires the standardization of data fields from the original source to a new target format.

For example dates may be transformed from **mm/dd/yyyy** to **dd/mm/yyyy** or addresses may be required to have the postcode field removed. Extraneous attributes within a field or the entire field are lost due to the cleansing process. For example a person's salutation (MR, MRS, DR, *et cetera*) if contained within a field may be required to be removed to comply to the name field specification. Often the ETL process can lose data or perhaps pervert the original data in some way and may in turn reduce overall quality of the data [Rajaraman, 2014]. As argued by N. Brierley, T. Tippetts and P. Cawley:

> "Formal data cleansing can easily overwhelm any human or perhaps the computing capacity of an organization." [N. Brierley and Cawley, 2014.]

This problem was also identified by Vincent Burner in 2007:

> "that the data volume may overwhelm the Extract Transform Load process and that *data cleansing* may introduce unintentional errors." McBurney [2007]

Data Triage is a different approach to ETL in that the raw value of the data is always maintained throughout the transformation process. The data is loaded into the data stores **verbatim** unless there are structural transformation issues with the original data source.

To summarize the differences between the Data Triage and ETL:

- Data Triage does not alter the original data value or format whilst ETL may alter a field's content.
- Data Triage will not eliminate any data field contained within the original data source whilst the result of an ETL load process will eliminate fields that do not comply to a fixed schema.

The purpose to cleanse or not cleanse can best be expressed utilizing the **Beliefs**, **Desires** and **Intentions** [Bratman, 1999] methodology based upon Michael Bratman's theory of human practical reasoning [Castanedo, 2011]. Michael Bratman's theory can be applied to data analysis where the data modeler takes the following into consideration:

Beliefs Beliefs provide the *inference rules* to process and manipulate the incoming data. These rules can be captured and reused for new data sets or reapplied to existing data sets.

Desires Desires represent how the data can best be used or processed. Desires represent the motivation behind the data and are generally expressed as an accomplishment. For example, this data can be used to determine if this person is what is commonly expressed as 'on the move' or 'up to no good'.

Intentions Intentions represent the deliberate use of the data and how the data is to be used. For example, the data may obtained so that it correctly enhances the intelligence surrounding a particular criminal organization.

However, there is a cost to data cleansing:

- Inconsistent processing where cleansing involves human assessment.
- The elimination or removal of unwanted tokens within a field can lead to the introduction of errors and inconsistencies.
- The introduction of human judgment which in turn may lead to errant assumptions that result in decisions based on false or misleading data interpretation.

Jimmy Lin and Dmitiry Ryaboy state:

> "That a major problem for the data scientist is to *flatten the bumps* as a result of the heterogeneity of data." [Lin and Ryaboy, 2013]

1.3 Data Provenance

Data provenance is the ability to retain the lineage between the original and processed data. This is an important attribute of data quality as Paulo Pinheiro da Silva states:

> "without acknowledging data provenance the quality of data will decline." [Paulo Pinheiro da Silva, 2007]

Paul Groth, co-chair W3C Provenance Working Group said:

> "The provenance of information is crucial in deciding whether information is to be trusted, how it should be integrated with other diverse information sources, and how to give credit to its originators when reusing it,.." [Oracle - Release, 2013]

2 Data Integration and Matching

Many challenges face all Big Data implementations with the ability to combine eclectic data sources into a single view being the most significant. Over time, as new knowledge about the data source becomes apparent, this in turn may lead to a greater understanding of the data structure. This is where SLA allows one to adapt the model to conform to a new understanding of the data. To facilitate the integration process a universal schema can be applied to all data sources which in turn can be used to provide a *common* view to the data sources. It is important characteristic of SLA that it does not change any individual data values.

3 Data Fusion and the Schema

Data fusion is the process where data sources are combined into a single view. This view can be used as input to various analytical and data-mining techniques and still utilize the SLA schema from the ingestion process.

The schema may change over time and this will not affect the raw data the schema represents. A SLA schema can be used for the following purposes:

- Provide a consistant view of the data source contained within the *big data* repository.
- Assist in the creation indexes and formulate index strategies.
- Identify portions of data source suitable for extraction and further analysis.
- An input **map** to a *Hadoop map reduction* task.

3.1 Schema First Frameworks

The history of database implementations including the *codaysl* network model, hierarchical database systems such as Information Management System (IMS) by IBM and more recent relational database implementation require a schema definition to be established before any data can be stored. Changes can be made to the schema afterward but will ultimately alter the data represented by the schema. For example, an addition of a column to a schema within a relational table will result in a *null* value for every row contained within that table. If a column is removed from the schema then all the associated data for the deleted column is lost. The issue was identified by IBM in 2011:

> "The schema is not meta-data but *combines* the data with a structural representation. Therefore, changes to the schema results in changes to the data even if it is only the addition of a *null* field value to each record." [IBM, 2011].

Schema protocols like Thrift, Avro, CORBA, DCOM and XML all have a schema definition language and fall into the category of schema first whereby the schema is fixed and difficult to change after inception.

Figure 3: Schema Last Models

3.2 The Schema Last Model

The model is a logical group of fields. A model may hold fields contained within another model. Models that do not hold fields from other models are said to be **distinct** (figure 4). Models containing all the fields held within another model are said to be encapsulated (figure 6). There is no restriction on the number of encapsulated or distinct models contained within a SLA schema. Models may share fields within the same schema definition (figure 5).

The same model definition may reappear in another data source schema. It is also possible that two or more data sources have identical SLA schema definitions.

3.3 The Field Label

Each field within the SLA schema definition must have an associated label. The default value for the label should be the name given to the field within the original data source. If there is no name within the original data source then a descriptive name is assigned to the label.

3.4 The Field Domain

Each field may have an associated domain where the domain determines the field's potential range of values. A typical domain would be **name** where the **name** may contain either a person or organization name. If the field were only to contain only a person name then **person-name** would be the field's domain. Other domains may be: phone-number, complete-address, contact-address. It is important to restrict or manage the number of domains and remove any ambiguity amongst domain specifications. A domain should not be confused with a primitive data type (string, integer, float, *et cetera*) and a primitive data type is not descriptive enough to be used as a domain.

3.5 Ontological Support

Not many data modeling products have Ontology support [Maxim, 2013], however W3C has specified Ontology Web Language (OWL) which defines a comprehensive ontological language [W3C, 2012]. OWL allows the definition of hierarchical ontological structures based on the RDF specification. SLA can also assist in the ontological description of the data in regards to domain classification as shown in figure 8

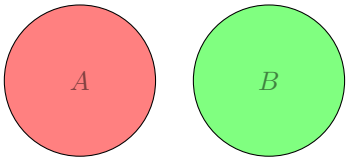


Figure 4: Distinct Models

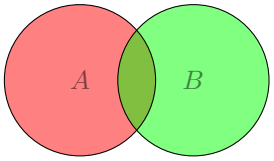


Figure 5: Overlapping Models

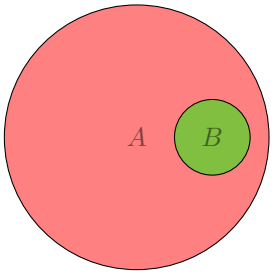


Figure 6: Encapsulated Model

and OWL could be used to describe the ontological structure of the domain.

4 Formal Description

SLA provides a formal *framework* to describe the process and the language necessary to describe a data source. The process provides an inherent feedback loop and is an iterative process. It is always possible to 'start over again' if the SLA schema definition is proved to be incorrect.

The SLA process consists of the following phases:

1. **Data source triage**: The coercion of data into a form where it can be uploaded into the data repository. No data is lost or changed during the **triage** process.
2. **Schema specification**:
   (a) The labeling of all fields contained within a data source.
   (b) The association of each field with a specific domain.
   (c) The creation of models and associated field memberships.
   (d) Schema storage and version management.
3. **Suitability**: Determine the schema is an accurate representation of the data source.
4. **Application**: Create indexes based on the schema defined in step 2.
5. **Verification**: Ensure that by application of the Schema that erroneous indexes are not created and that the models defined within the Schema correspond with the data contained within the data source.
6. **Fuse**: Identify entities which are common within the data source.
7. **Resolve**: The construction of a '*single source of truth*' to represent an entity (generally a person or organization) within the entire data repository.

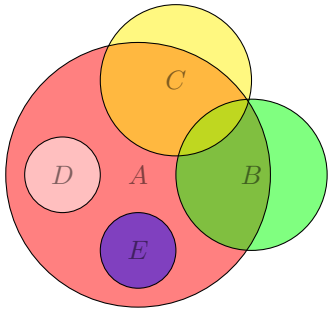


Figure 7: Multiple Encapsulated Models

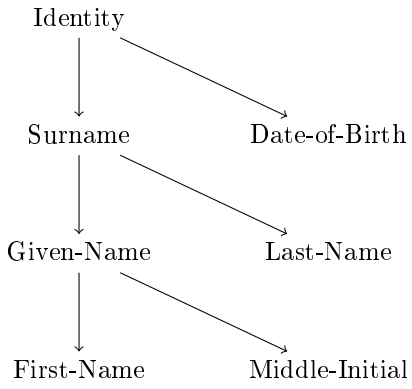


Figure 8: Ontological Support

The Schema definition is formally represented in RDF N3 form[W3C Group, 2014]. RDF blank nodes are used to group the fields that are contained within a model. Figure 11 is a simple schema definition containing three fields and two models.

4.1 Schema Storage Considerations

There is no prescriptive persistent storage technology and there is an advantage to store the Schema with the raw data.The resultant removal of the raw data will lead to the destruction of the data's associated Schema. The Minerva reference implementation stores the data as an RDF list structure. The Schema is then stored within the same RDF graph alongside the RDF list.

4.2 Map Reduction

Map Reduction is a strategy that has become popular to process large data-sets. To summarize, the *map* phase maps the input data into a known structure and the reduction phase summarizes the data. Map Reduction has become a tool of choice amongst data mining practitioners [Rajaraman, 2014]. The Map Reduction approach can take full advantage of the SLA in that the SLA schema can be used as the input map.

5 Data Fusion

Data Fusion is the process of integration of multiple data data sources into a single view. Federico Castanedo defines data fusion as:

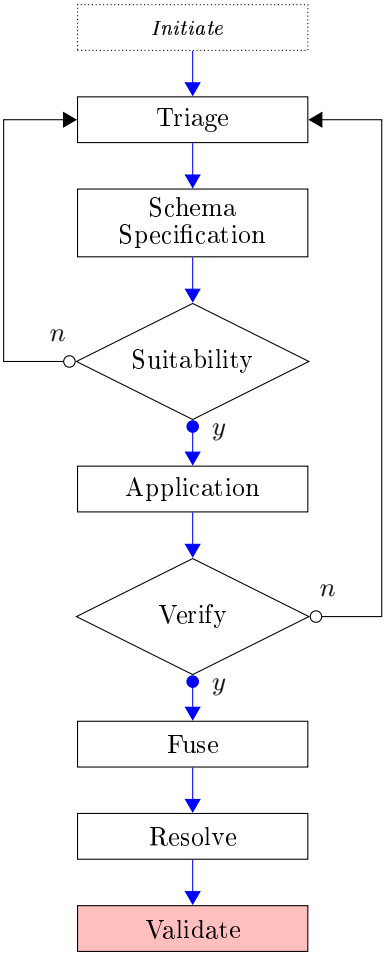


Figure 9: Schema Last Process

> "The integration of data and knowledge from several sources is known as data fusion." [Castanedo, 2013]

The SLA provides the framework to achieve this, in that a consistent modeling technique is applied to all the data sources contained within the repository. The *fused* data can then be modeled again and the result in itself can become another data source. The Data Fusion consolidation provides a unique perspective to the Big Data repository. It is essential to refer back to the original raw data source in order that the correct data provenance is maintained.

5.1 Indexing Strategies

Indexes are a crucial part of any Big Data implementation and it is important that the indexes are independent to the stored data. The SLA schema defines the models and structure of the indexes that will be used to query the data contained within the repository. The indexes can be tailored to a specific purpose for example: an index may be a person's name and allow for typographical errors (see figure 12). [Rajaraman, 2014, Christen, 2012].

The SLA Schema can be used as the basis to generate any number of index strategies. For example, SOLR which is a document index system based on Lucene is a high performance text based indexing engine. SOLR indexing definitions can utilize the SLA schema that contains sufficient meta-data to define SOLR filters and tokenizers.

```
@prefix rdf: <http://www.w3.org/...#>
@prefix rdfs: <http://www.w3.org/...#>
@prefix schema: <http://www.sla.org.au/...#>

<file:/schema.ttl> schema#:schema
 [
 <schema#://model>
    [
     rdf:#_1 _:b1 ;
     rdf:#_2 _:b2 ;
     rdf:_#3 _:b0
   ];
 <schema#://model>
   [ rdf#:_0
    _:b0 , _:b2 , _:b1 ;
     rdf#:_1
    _:b0 , _:b2
   ]
 ] .
_:b0 rdfs#:domain "document-id" ;
     rdfs#:label "document-name" .
_:b1 rdfs#:domain "first-name" ;
 rdfs#:label "given-name" .
_:b2 rdfs#:domain "last-name" ;
 rdfs#:label "surname" .
```

Figure 10: Schema Last Definition in N3 format

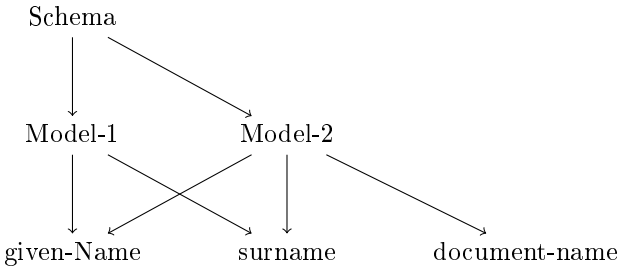


Figure 11: Schema Definition - Visual Representation of figure 10

5.2 Classifiers and Stochastic Attributes

A row may contain one or more models and a model can represent an entity that may be an individual or company [Rajaraman, 2014]. Each data source could be assigned additional attributes or classifiers in the form of *meta-data*. Classifiers are used to describe the nature of the data source and how the data source was originally obtained. Additional attributes would also add value to the data source; for example: Geo-spatial coordinates, time of ingestion and an intelligence ratings.

5.3 Meta-data

Additional meta-data in the form of name/value pairs or tags are used to annotate the data source or models contained within the data source (see figure 13). This includes:

- the security classification (Secret, Highly Protected, Unclassified),
- the name of the organization that supplied the data
- any special data handling requirements.

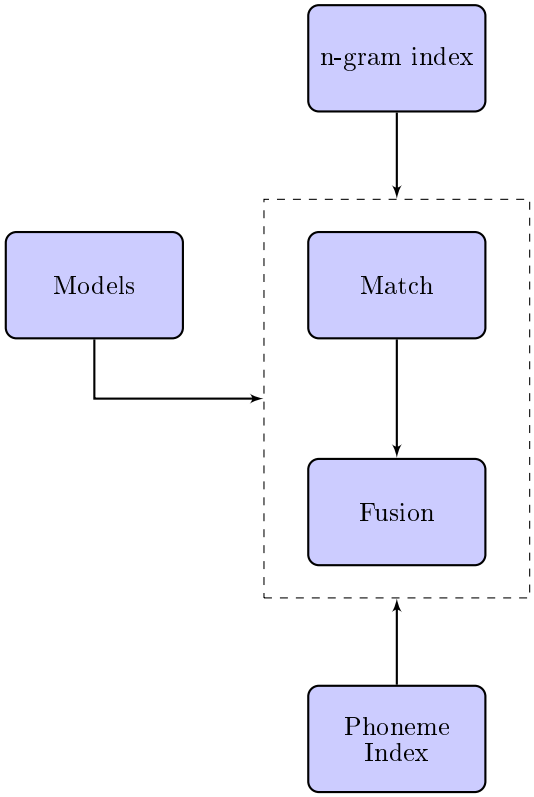


Figure 12: Index Strategies and Schema Last

Meta-data were also used to describe any associations amongst data sources within the data repository. For example, if two data sources are related in some way then this relationship could be captured within descriptive meta-data tags.

It is essential to conform to, or at the very least map to, known standards for meta-data and that the meta-data is consitent across all data sources, rather than using proprietary or homegrown schemes. In addition the meta-data scheme(s) are defined within a thesaurus that has been developed for the provision of a common vocabulary across the data sources.

> "Good meta-data conforms to community standards in a way that is appropriate to the materials in the collection, users of the collection, and current and potential future uses of the collection." [NISO, 2007]

5.4 The Single Source of Truth

Part of the Identity Resolution process is to formulate the *golden record* which is a consolidated view of an individual or organization. The consolidated view in turn can contain a *match* score between the two matched entities. It is important that the new knowledge that pertains to each record can be reapplied in the construction of the consolidated view. The consolidated record can be queried by the user and this will contain all the current information relevant to the entity.

6 Initial Findings

The 'Schema Last' approach to data fusion has been in operation within the ACC since 2012 and during that time there has been over a thousand data sources processed that utilized this technique. Initial results have shown a dramatic reduction in the time taken to

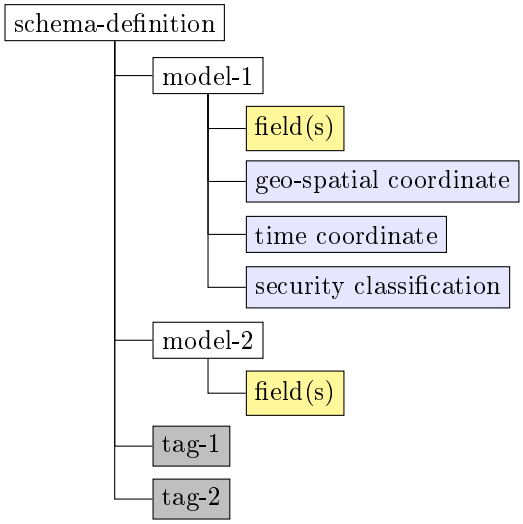


Figure 13: Model Description including Attributes and Classifiers

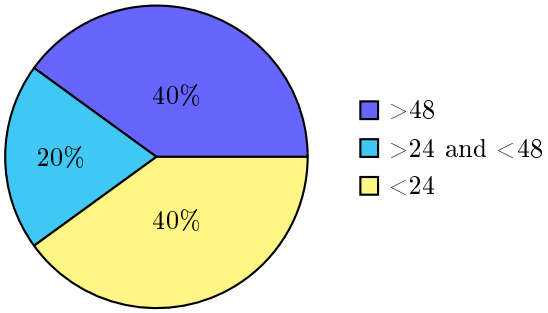


Figure 14: Time taken in hours to cleanse 50 Data Sources

collate data sets received by the ACC. One particular data set took over three months to cleanse; whilst utilizing SLA this particular data set only took minutes to upload and model. The reaction by the analysts within the ACC has been positive and they have commented on how vital information was lost due to the previous ETL process. SLA has allowed the data source modeling process to be delegated to unskilled staff which has allowed the skilled data analysts to focus on model development and target detection. The improvements are shown in both figure 14 and figure 15.

In addition, it took only **thirty** domains to describe every field contained within the data sources.

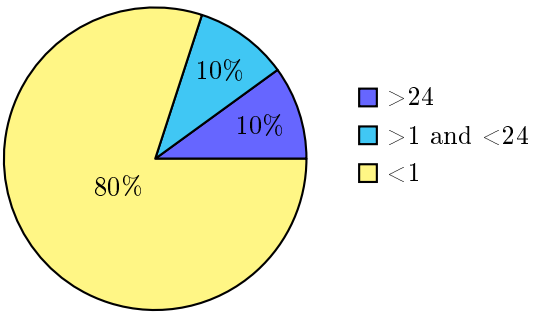


Figure 15: Time taken in hours to triage 1010 Data Sources

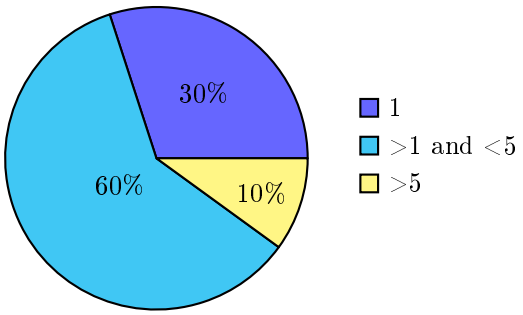


Figure 16: The number models required to describe a data source

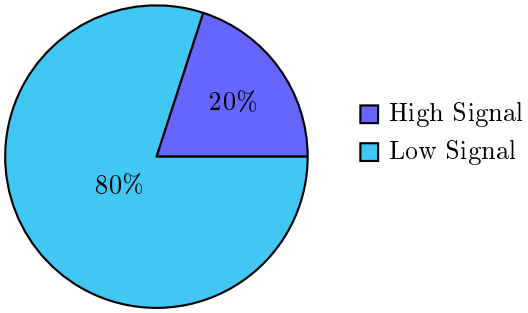


Figure 17: The Ratio of High to Low Data Sources in terms of number

The domains types were deliberately restricted. For example, the domain *document identifier* was used to describe any unique identifier within a data source. There were no detectable overlaps as a result of this generalized classification of domains.

The Minerva Schema Designer 19 was developed to allow users to create and modify SLA Schema Definitions. The schema models could be saved along side the data or separately as a file. The interface was deliberately kept simple to minimize user training requirements and reduce modeling errors.

The reaction from users of the application was positive. Other Australian government departments have shown interest with the *schema last approach* and have expressed a desire to run a proof-of-concept within their respective organizations.

6.1 Future Work

The resultant 'fused' data has raised a number of concerns among the data analysts, their greatest concern is the ever increasing number of results returned from search queries. To determine a *single source of truth* for an entity has proved to be a challenge especially when there is no unique identifier to link one entity across multiple data sources. The 'schema last approach' is first step towards this goal and work can be done in this area to improve the accuracy in establishing links between *like* entities within different data sources.

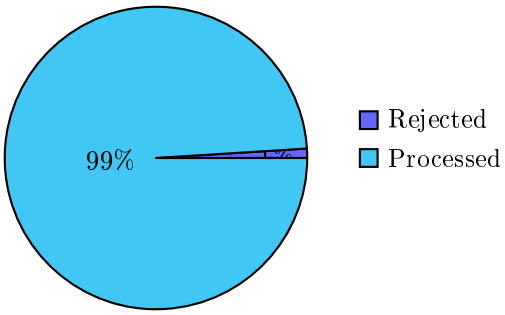


Figure 18: The ratio of processed to rejected data sources

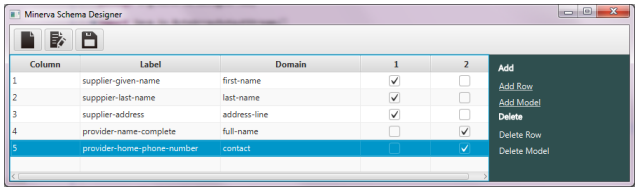


Figure 19: Minerva Schema Designer

7 Conclusion

The ACC now has a tool to address the eclectic nature of the data sent to them. SLA can be used to define index strategies, provide the *map* in Map Reduction and the form foundation for data mining. The formal definition of the schema syntax allows for the **interchange** of models and the **sharing** of meta-data among organizations and institutions.

SLA has provided the platform to *fuse* data into a consolidated view and to resolve issues associated with variability and variety of data when obtained from multiple sources.

References

Michael E. Bratman. ***Beliefs, Desires and Intentions.*** **CSLI Publications, 1999.**

Federico Castanedo. A multi-agent architecture based on the bdi model for data fusion in visual sensor networks. ***Journal of Intelligent & Robotic Systems*****, 62(3-4):299–328, 2011. ISSN 0921-0296. doi: 10.1007/s10846-010-9448-1. URL** `http://dx.doi.org/10.1007/s10846-010-9448-1`**.**

Federico Castanedo. A review of data fusion techniques, 2013. URL `http://dx.doi.org/10.1155/2013/704504`**.**

Peter Christen. ***Data Matching.*** **Springer, 2012.**

Paulo Pinheiro da Silva. Propagation and provenance of probabilistic and interval uncertainty in cyberinfrastructure-related data processing and data fusion. ***DEPARTMENTAL TECHNICAL REPORTS (CS)*****, (UTEP-CS-07-56), 11 2007. URL** `http://digitalcommons.utep.edu/cgi/viewcontent.cgi?article=1224&context=cs_techrep`**.**

W3C Working Group. Resource description framework (rdf), 2 2014. URL `http://www.w3.org/RDF/`**.**

IBM. Terminology: Dynamic- vs. fixed-schema databases, July 2011. URL `http://www.dbms2.com/2011/07/31/dynamic-fixed-schema-databases/`**.**

Qing Wang Klaus-Dieter Schewe. Knowledge-aware identity services. *Knowledge and information systems*, 36:335–357, 2013.

Jimmy Lin and Dmitriy Ryaboy. Scaling big data mining infrastructure: The twitter experience. *SIGKDD Explor. Newsl.*, 14(2): 6–19, April 2013. ISSN 1931-0145. doi: 10.1145/2481244.2481247. URL http://doi.acm.org/10.1145/2481244.2481247.

Bakaev Maxim. Ontology to support web design activities in e-commerce software development processmore. 2013. URL http://www.academia.edu/929051/Ontology_to_Support_Web_Design_Activities_in_E-Commerce_Software_Development_Process.

Vincent McBurney. 17 mistakes that etl designers make with very large data, 2007. URL http://it.toolbox.com/blogs/infosphere. http://it.toolbox.com/blogs/infosphere/17-mistakes-that-etl-designers-make-with-very-large-data-19264.

T. Tippetts N. Brierley and P. Cawley. Data fusion for automated non-destructive inspection. *Proceedings of the RSPA*, 2014. URL http://rspa.royalsocietypublishing.org/content/470/2167/20140167.abstract.

NISO. A framework of guidance for building good digital collections, 2007. URL http://www.niso.org/publications/rp/framework3.pdf.

Anand Rajaraman. *Mining of Massive Datasets*. 2014. URL http://infolab.stanford.edu/~ullman/mmds/book.pdf.

Oracle Press Release. Oracle implements w3cs standard for data provenance. 2013. URL http://www.oracle.com/us/corporate/press/2028860.

W3C. Owl 2 web ontology language structural specification and functional-style syntax, December 2012. URL http://www.w3.org/TR/owl2-syntax/.

A Triple Store Implementation to support Tabular Data

Neil Brittliff and Dharmendra Sharma

University of Canberra

Faculty of Education, Science, Technology and Mathematics

Dharmendra.Sharma@canberra.edu.au

Abstract

The acceptance of Triple Stores and the adoption of Triple Store as an alternative to relational database technology has not met the expectations demanded by the Big Data community. In addition, most triple store implementations are unable to store and extract tabular data as fast as many of the alternate Big Data solutions that are currently available. What is missing is a Triple Store implementation that can contain both graphical and tabular data. The SPARQL property path extension which was introduced into the SPARQL 1.1 specification provides the capability to retrieve tabular data through the traversal of unbounded RDF list structures. However, there are no currently available Triple Store implementations that fully support the SPARQL 1.1 property path extension and therefore are unable to traverse large RDF lists using the SPARQL language. This paper will describe a triple store implementation project targeting the property path extension utilizing features found in most columnar storage databases.

1 Motivation

Motivations to bridge Tabular data with Graphical Data abound. For example, within the intelligence community data is often received in a tabular format and this data needs to be stored and retrieved in the same format for evidentiary purposes. The RDF linked list structure provides the specification to store tabular data within an RDF graph. The introduction of the SPARQL 1.1 property path extension provides a means to retrieve tabular data without any loss of format and structure. Once the data has been stored within RDF lists then additional rows can be linked together to create other directed graph structures. The SPARQL 1.1 property path extension implementation is proving to be problematic amongst the various triple store implementations. As described by Lei Gai and Wei Chen [Lei Gai, 2014]

> "path queries are of common interest in OSN (Online Social Networks) analysis for the discovery of complex relations among entities. Despite the scalability and flexibility provided by the RDF model, Path queries performs poorly and a lack of efficient implementation in existing RDF management."

This paper will discuss how a columnar database with the additional support of the **Apache Jena** framework which provides SPARQL language support can store and retrieve both unbounded tabular data and directed graph structures. A reference implementation called Minerva was developed to demonstrate the feasibility a triple store that supports the property path extension and utilizes a columnar database to store RDF structures.

2 Introduction

The triplet is a data structure composed of a subject, predicate and object where the predicate binds the subject to the object (figure (1)). The subject and predicate must conform to the Uniform Resource Specification (URI) as described by Tim Berner-Lee in 1991 [Berners-Lee, 1991].

$$Subject \xrightarrow{\text{Predicate}} Object$$

Figure 1: Triplet Structure

The URI which is also referred to as a **resource** which provides the mechanism to link or associate one or more triplets. The object component of the the triplet serves dual purposes. If the object contains a **resource** this identifies a directed link between this triplet and other triplets within the graph (figure 2). However, if the object contains a literal value then this is the data associated with the triplet. If the object does contain a literal value then additional information is stored to identify the literal's primitive type whether the literal is a string, date or number.

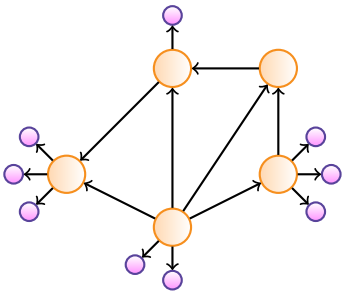


Figure 2: A directed graph structure

A *blank resource* is a special resource type similar to the URI used to group triplets together. A blank

resource can only be used as subject or object resource identifiers.

A *graph identifier* is an extension to the triplet specification. Like the subject and predicate, the graph identifier is represented by a resource. A graph identifier provides a mechanism to group large numbers of triplets together. A graph identifier is sometimes called the triplet's *context*.

The W3C specification Resource Definition Framework (RDF) is a formal representation of triplet structures [W3C Group, 2014]. RDF can represent many different types of information. For example, the RDF necessary to describe the structure required to capture the ingredients of the simple food stuff chutney are shown in figure 3 and visually represented in figure 4.

```
@prefix ex: <http://example.org/>.
@prefix rdf: <http://www.w3.org/...#>
ex:Chutney ex:hasIngredient ex:item1.
ex:Chutney ex:hasIngredient ex:i.
ex:item1 rdf:value ex:greenMango.
ex:item1 rdf:amount "50gm".
ex:item2 rdf:value ex:relish
ex:item2 rdf:amount "20gm".
```

Figure 3: RDF to describe the contents of chutney

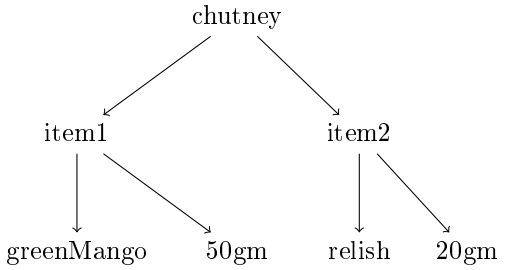


Figure 4: A visual representation of chutney

W3C has defined a language specification SPARQL (**S**imple **P**rotocol **a**nd **R**DF **Q**uery **L**anguage) which is an RDF triplet querying language [W3C Group, 2008]. SPARQL provides a specification to retrieve triplets based on one or more graph patterns. An example of a SPARQL query is shown in figure 5. The SPARQL language specification also extends to the format in which the triplets are returned and this includes: XML, N3 and JSON. The basic SPARQL retrieval patterns are shown in table 1. The pattern as described can form part of a union, intersection or filter within a SPARQL query. For a full description of the SPARQL language specification refer to the W3C specification [W3C Group, 2014].

2.1 RDF Data Structures

The Resource Definition Framework has identified and defined four major data structures:

- **Bag** - is a collection of triplets with no specific order.
- **Sequence** - is a collection of triplets where the order matters.
- **Alternate** - is a collection of triplets however only one triplet can ever be selected.
- **List** - is a generic specification for a *linked* list structure (see figure 7 as an example of a RDF linked list).

```
select ?subj ?pred ?obj
  where {
    {
      ?subj ?pred ?obj.
      ?store <fusion://map> ?subj.
    } union {
      ?subj ?pred ?obj.
      ?store <fusion://schema> ?subj.
    } union {
      ?subj ?pred ?obj.
      ?store <fusion://model> ?subj.
    }
}
```

Figure 5: SPARQL Language Example

Subject	Predicate	Object	Description
○	Any	Any	Retrieve the predicate and object for a given subject.
○	○	Any	Retrieve all subject and predicates for a given object.
Any	○	Any	Retrieve all triplets that contain a given predicate
Any	Any	○	Return all subjects and predicates for a given object
Any	○	○	Return all subjects for a specific object and predicate combination
○	Any	○	Return all the predicates for a given subject and object
Any	Any	Any	Return all the triplets contained with in a graph
○	○	○	Determine if a specific triple pattern exists within a graph

Table 1: Basic Retrieval Patterns

```
@prefix ex: <http://example.org/>.
@prefix rdf: <http://www.w3.org/...#>
ex:list1 rdf:first [
    ex:artist "Basement Jaxx".
    ex:year "2004".
    ex:song "Good Luck".
  ].
ex:list1 rdf:next _:a01.
_:a01 rdf:first [
    ex:artist "Groove Armada".
    ex:year "1996".
    ex:song "Superstylin".
  ].
_:a01 rdf:next _:a02.
_:a02 rdf:first [
    ex:artist "Fat Boy Slim".
    ex:year "2002".
    ex:song "Weapon of Choice".
  ].
_:a02 rdf:next _:a03.
_:a03 rdf:first [
    ex:artist "Daft Punk".
    ex:year "2013".
    ex:song "Get Lucky".
  ].
_:a03 rdf:next _:a04.
_:a04 rdf:first [
    ex:artist "Pharrell Williams".
    ex:year "2013".
    ex:song "Happy".
  ].
_:a04 rdf:next rdf:nil.
```

Figure 6: RDF List represented in N3 format

In addition to the standard RDF data structures, the triple store can represent and store any linkage pattern [Pascal Hitzler, 2010] (figure 2 an example of an undirected graph).

The RDF list structure is designed to accommodate large ordered lists. The list has a node that represents the list's head and is terminated by the **nil** RDF resource which represents the list's tail [W3C Group, 2014]. Each RDF node within the list contains links to both the data and to the next item within the list. These links are represented by the RDF *first* and RDF *next* predicates respectively. 7.) There is no restriction to the number of nodes contained within a linked list, however the standard implementation supported by **Apache Jena** will attempt to load the entire linked list structure into memory and will fail if this exceeds the amount of available memory.

SPARQL 1.1 introduced the *property path expression* (see table 2) which simplified the navigation of the RDF list structure. The *property path expression* can be used to search any directed graph and is particularly useful to extract items from the RDF list structure. Furthermore, the *property path expression* can be useful to perform simple Social Network Analysis by techniques including determining the degrees of separation between any two nodes within a graph, centrality measurements and k-core graph analysis. The rationale for the introduction of the *property path expression* as stated by the W3C SPARQL working group is as follows:

> "Property paths allow for more concise expression of some SPARQL basic graph patterns and also add the ability to match arbitrary length paths." [W3C, 2010]

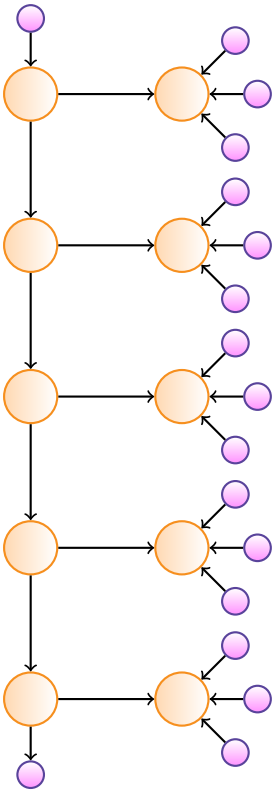


Figure 7: A visual representation of figure 6

Table 2 summarizes the extension.

3 RDF/SPARQL Implementations

An excellent example of a SPARQL implementations is the **Apache Jena** framework. The Apache Jena framework was originally developed by Hewlett Packard in their Bristol laboratories in 2010 and subsequently given to the Apache Foundation. Since leaving incubation status in April 2012, Apache Jena is now a top-level product. The framework contains a SPARQL parser (Jena ARQ) which transforms the SPARQL query into one or more triple patterns (see table(1)). This provides the abstraction layer between the logical structure represented by the triplet and the physical storage implementation. Jena is not the only library that provides this functionality. Other implementations providing similar functionality include: Aduna's Sesame library ***openrdf*** and the **Redland librdf** library. Aduna's Sesame library also provides an abstraction layer between the physical and logical and supports numerous triplet storage implementations. Both Apache Jena and Aduna's Sesame are both written in Java, however the **Redland librdf** library is developed specifically for the 'C' and 'C++' computer languages.

W3C has also defined a REST style protocol to communicate to SPARQL implementations W3C [2008]. All three libraries support the W3C SPARQL REST protocol specification.

4 SQL Implementations

Translating SPARQL to SQL can generate complex SQL statements. The paper by Eric Prud'hommeaux and Alexandre Bertails from **W3C** describe the process how this could be achieved [Prudhommeaux and

Syntax Form	Matches
uri	URI or a prefixed name. A path of length one.
^elt	Inverse path (object to subject).
(elt)	A group path *elt*, brackets control precedence.
elt1 / elt2	A sequence path of *elt1*, followed by *elt2*
elt1 ^ elt2	Shorthand for *elt1* / ^*elt2*, that is elt1 followed by the inverse of *elt2*.
elt1 | elt2	A alternative path of elt1, or elt2 (all possibilities are tried).
elt*	A path of zero or more occurrences of *elt*.
elt+	A path of one or more occurrences of *elt*.
elt?	A path of zero or one *elt*.
elt{n,m}	A path between n and m occurrences of *elt*.
elt{n}	Exactly n occurrences of *elt*. A fixed length path.
elt{n,}	n or more occurrences of *elt*.
elt{,n}	Between *0* and n occurrences of *elt*.

Table 2: Property Path Expressions

Bertails, 2010]. In their paper the authors suggest to use the SQL union and group clauses but omit to discuss the recursive select clause available in some SQL implementations. Translated SPARQL queries largely rely on the SQL optimizer for any performance gains and as the Eric Prud'hommeaux and Alexandre Bertails state:

> "...that such a translation may not benefit from the capabilities of the SQL optimizer."[Prudhommeaux and Bertails, 2010]

This was also confirmed by **IBM Research** which developed a strategy to store and retrieve triplets utilizing a DB2 back-end:

> "SPARQL queries, as we illustrate in this paper. Such queries often have deep, nested sub-queries whose inter-relationships are lost when optimizations are limited by the scope of single triple or individual conjunctive patterns. To address such limitations, we introduce a hybrid two-step approach to query optimization."[Mihaela A. Bornea, 2012]

In addition, the SPARQL specification has been expanded to include a property path that usually requires a form of recursion to implement. Usually relational databases are not well suited to store graph structures. This is largely due to the inconsistent approach taken by the various database implementations to handle recursive queries. For example Postgres uses the SQL language extension **WITH RECURSIVE** to navigate recursive table structures but Oracle provides identical functionality **START-FROM** and **CONNECT-BY** (see figure 8). However there are no SPARQL to SQL translators that take advantage of these SQL recursive extensions.

5 Non-Relational Triple Store implementations

There are a number of non-relational implementations that provide some support for triple storage

POSTGRES:

```
WITH RECURSIVE t(n) AS (
        VALUES (1)
UNION ALL
        SELECT n+1 FROM t WHERE n < 100 )
SELECT sum(n) FROM t;
```

ORACLE:

```
SELECT name, SUM(salary) "Total" FROM (
        SELECT CONNECT_BY_ROOT name, Salary
                FROM Emps
                        WHERE dept_id = 110
                CONNECT BY PRIOR Emp_id = mgr_id)
GROUP BY name;
```

Figure 8: SQL recursive queries various platforms

and SPARQL retrieve. Table 3 summarizes the well known non-relational triple store implementations.

Classification	Product	Comments
Columnar	Cassandra HBase Accumulo	Limited RDF implementations
Key Value	Berkley DB Dynamo Redis Riak	Becoming more popular as an RDF platform. Oracle has developed an RDF store for their Berkley DB (NOSQLDB) product
Graph Database	Neo4j OpenLink Virtuoso	Some RDF Support via **Tinkerpop**
RDF Native	Bigdata	Big Data can operate stand-alone or using the Sesame SAIL

Table 3: Comparison of Triple Storage Implementations

6 Evaluation of Existing Implementations

The comparisons in this paper are based on the Lehigh University Benchmark (LUBM) software tools M. Schmidt [2009]. The tests were performed on a Oracle SPARC machine with 128GB and eight 2.2GHz processors. The Operating system is Solaris 11 running Postgres 9.2. and Oracle database version 11.2 G with the Spatial/RDF option activated. The Sesame version from **Aduna** at the time was 2.7.8 that utilize the Tomcat 7 servlet engine. All the below implementations utilized Aduna's Sesame as the RDF front end triple provider. Each have their own SAIL implementation where the SAIL (**S**torage **A**nd **I**nference **L**ayer) is a low level System API (SPI) for an RDF store and inferencer.

The Oracle **Sail** implementation was not compatible with the tested version of Sesame and a minor modification of Oracle **SAIL's** Java code was required to prevent a Java exception thrown on initiation.

There were two main requirements to be considered in order to establish a viable triplet store repository. The first requirement was that the RDF triples could be loaded quickly and the memory foot print was restricted to 250 megabytes of main memory. The second requirement was that the entire triple store

could be unloaded as a **n3** file with a memory foot print less than 250 megabytes.

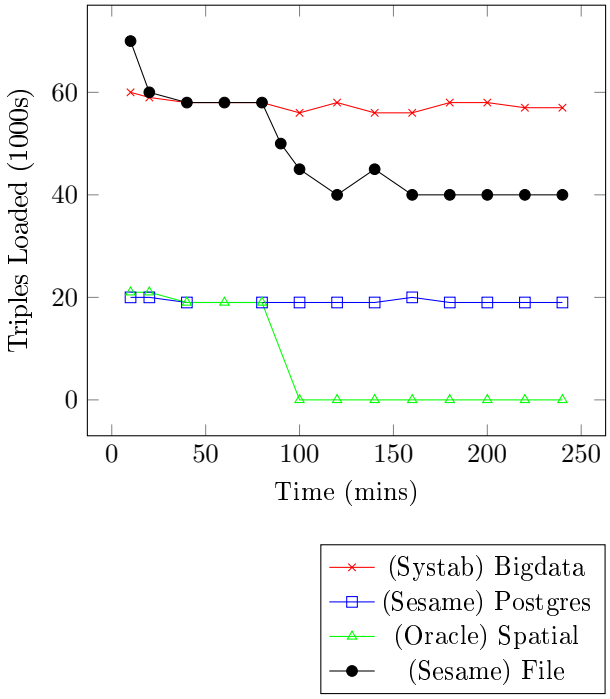


Figure 9: Comparison of Triple Store Implementations (Load)

Except for the Oracle implementation which failed after only 90 minutes on both the load and extraction test, all other tested implementations were able to load the LUBM RDF triplets. During the Oracle extraction test, the Oracle's Performance Monitor reported an unusual high number of locks on several tables that contained the triplets and this may have contributed to the upload test failure. Overall, all implementations were **unable** to extract triplets without incurring a non-trivial memory leak. In addition, the overall performance of the various implementations declined over time.

The SPARQL property path expression support was also tested and the results are described in figure 12. In accordance to the requirements, the only implementation that showed any promise was the Sesame Postgres implementation. The documentation from both *Systap's Bigdata* and *Sesame's native file storage* did indicate that property path support was available within their implementation. However, both implementations reported a *syntax error* when the property path clause was specified within the SPARQL select expression.

7 The Minerva Project

The poor performance of the Aduna Sesame product lead to the initiation of the Minerva project which was an attempt to improve upon existing implementations specifically targeting the RDF list ingestion and extraction using the SPARQL property path extension. Taking the experience from the previous tests, this series of implementations was designed to reach the specified memory and storage footprints. This implementation was developed at the Australian Crime Commission as a proof of concept to determine if the triple store supported by the DataStax Cassandra columnar database was feasible The Aduna Sesame product was discarded in favour of Apache Jena which

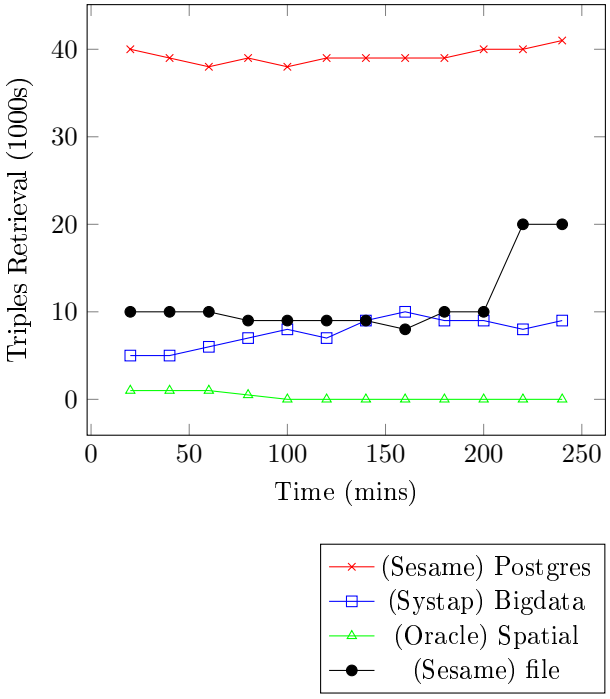


Figure 10: Comparison of Triple Store Implementations (Retrieval)

provided property path support. Cassandra was chosen as the back end database because of its ability to distribute work load amongst multiple nodes and its general availability on most platforms.

Cassandra consists of the following key components: [DataStax, 2014]

- Cluster: a container for one more *keyspaces*. A Cassandra instance may contain only one cluster however the cluster may be distributed across one or more computers.
- Keyspaces: a container for *column families*. A keyspace is analogous to a schema in a relational database.
- Column families: analogous to a tables in a relational database which may contain unlimited number of rows and up to four billion columns. Each row within a column family must have a single unique key to identify the row.
- Columns: A column is name-value pair. The name can contain 64 Kilobytes of data and there is no real limitation to the size of the value.

Cassandra has the unique feature where the columns are stored in a nominated order (alphabetic, numeric, date or self-defined). Cassandra also has the concept of a *super* column which is a column that may contain any number of sub-columns. Support for alternate access to data is provided through *secondary* indexes. There is no restriction to the number of secondary indexes a column family may contain, however it is recommended that the secondary indexes are of low cardinality.

Keyspaces were used to group triplets into graphs. The Jena ARQ parser provided the graph node and this was used to determine the Cassandra keyspace that contained the triplets. There is no documented limit to the number of keyspaces allowed within a Cassandra instance. HBase shares many of the same attributes with Cassandra except that there is no support for keyspaces.

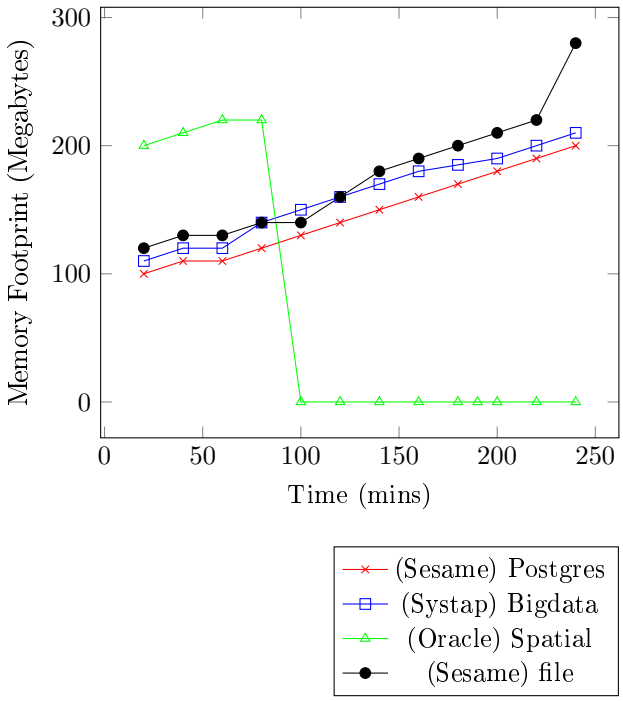


Figure 11: Memory Footprint on extraction

Property Path	elt*	elt{,5}	elt{,10}
Sesame (Postges)	✗	✓	✗
Sesame (File)	✗	✗	✗
Oracle Spatial	✗	✗	✗
Systap Bigdata	✗	✗	✗

Figure 12: SPARQL 1.1 Property Path Support

7.1 Storage and Processing Strategies

The final design determined there were three possible storage and processing strategies which are:

- Strategy 1 [s1] : A column family for each predicate within a graph. The column family would comprise of two columns which contain both the subject and object respectively. Secondary indexes would be used to select particular rows if only one the subject or object is known. There is an assumption that there will only be a limited number of predicates within a graph. An MD-5 hash key of both the subject and predicate would be the key to the row. Therefore, a total of five column families are required to store the triplets which are shown in figure 6.
- Strategy 2 [s2] : This strategy comprised of three column families which are SPO (Subject,Predicate,Object), OSP (Object,Predicate,Subject) and PSO (Predicate,Subject,Object) to store a triplet's resources. The PSO column family takes advantage of Cassandra's capability to store any number of columns within a row. The MD-5 hash was dispensed with in favour of the target artifact value contained within the triplet. The SPO column family would be the *subject*, the OPS column family would be both the *object* and the *predicate* for the PSO column family. There was an exception for the OPS column family in that an MD-5 hash value was used as the key if the *object* represented a literal. There were no secondary indexes used in this implementation (see 14).
- Strategy 3 [s3] : A column family to contain all the literal values for a specific subject identified by the predicate (Subject, Predicate, Literal). Each row within this column family is referred to as a Node. There are two additional column families for every predicate within the graph and these column families are in the form of SPO (Subject,Predicate,Object) and OSP (Object,Predicate,Subject). For the SPO column families if the object is a literal value then a placeholder - the asterisk character - is used to indicate that the value is stored within a Node column family. Otherwise, the subject and object resources are rows within SPO and OPS column families where the predicate value determines the column family name (see 15).

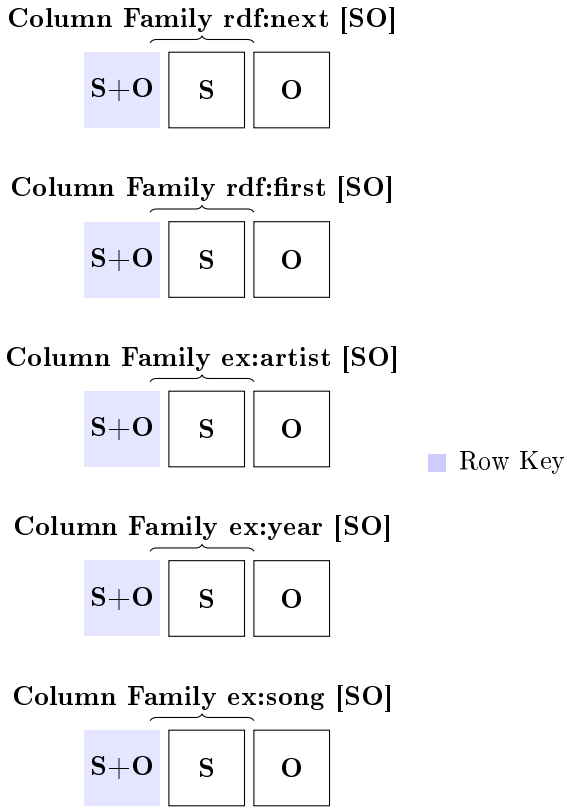


Figure 13: Strategy 1: Column Family Structure

7.2 Load Performance

All tests were performed on the same hardware as the previous evaluations but were using the latest version of Cassandra which at the time was 2.0.9. The **Thrift** protocol was used to communicate to Cassandra via an **Apache Hector** proxy. The load figures clearly demonstrate that the second strategy [s2] clearly outperformed the third strategy [s3] which outperformed the first strategy [s1] (see figure 16). The LUBM test contained approximately 28 predicate *uris* which meant there was an additional 24 column families and 56 secondary indexes required to support *s1* over *s2*. *s3* also required the same 24 additional column families but did not require the secondary index support.

The *s1* ingestion test was redone without the secondary indexes and subsequently the performance

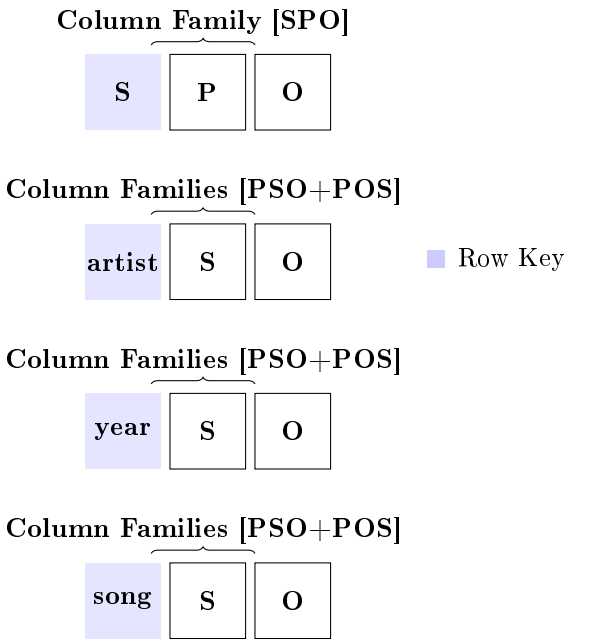


Figure 14: Strategy 2: Column Family Structure

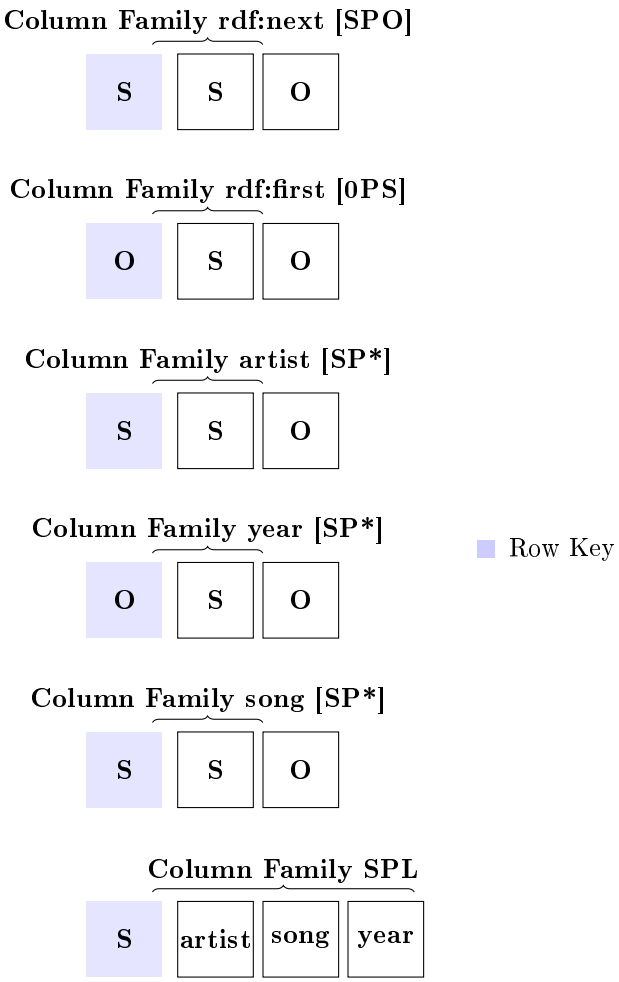


Figure 15: Strategy 3: Column Family Structure

improved dramatically. This confirmed it was secondary index construction that caused the performance degradation, however without secondary indexes this makes *s1* unusable.

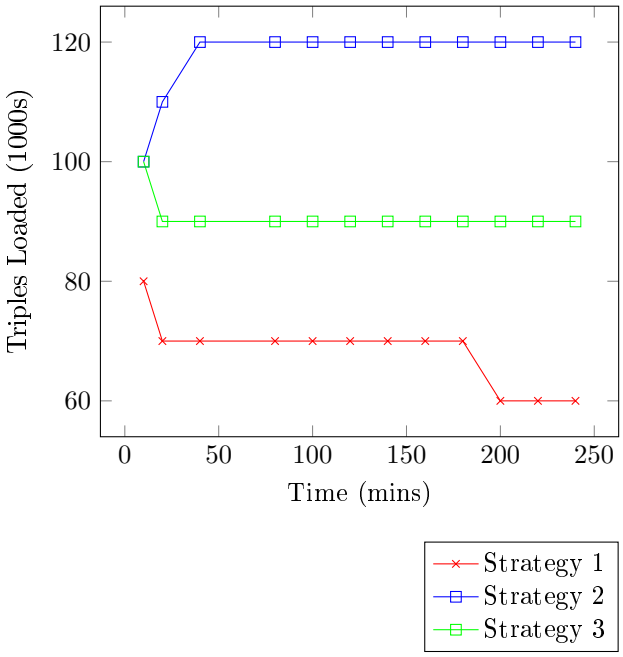


Figure 16: Minerva - Load Performance

7.3 Extraction Performance

All strategies could support the nine possible triplet search patterns (see 1). For *s1*, if the predicate is unknown and a subject or object is specified then all column families have to be searched for any matching triplets. For *s2*, the presence of the **PSO** column family that was keyed on the predicate removed the need to search multiple column families. s3 could utilize the column family name which happened to be the predicate *uri* to efficiently traverse the RDF list structure (see 17). There was no detectable memory leaks with either the *s1*, *s2* or *s3* implementations during both the ingestion and extraction tests (see 18).

7.4 Property Path Support

The unbounded property path expression's performance was problematic for both *s1*, *s2* and not so with *s3* (see figure 19). To improve performance of *s2* the *link* triplets involved in RDF list structures were placed in a separate *column family* with two rows keyed by the **rdf:first** and **rdf:last**. Each link is assigned an ascending number which is used as the column identifier. This meant that triplets can only ever be appended to an RDF list, therefore if insertions are required anywhere within the list structure other than at the end then the entire list would have to be reconstructed.

7.5 Implementation Acceptance

The decision was made to deploy *s3* over *s1* and *s2*. Utilizing s*3*, tabular data in excess of 60 million rows was loaded in approximately 5 hours and extracted in 6 hours.

The initial reaction to triple store implementation has been positive amongst the test audience. However

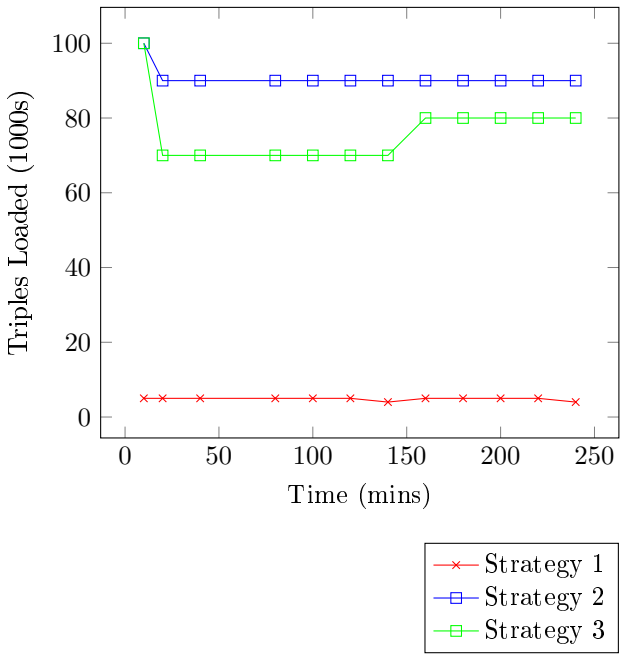


Figure 17: Minerva - Extraction Performance

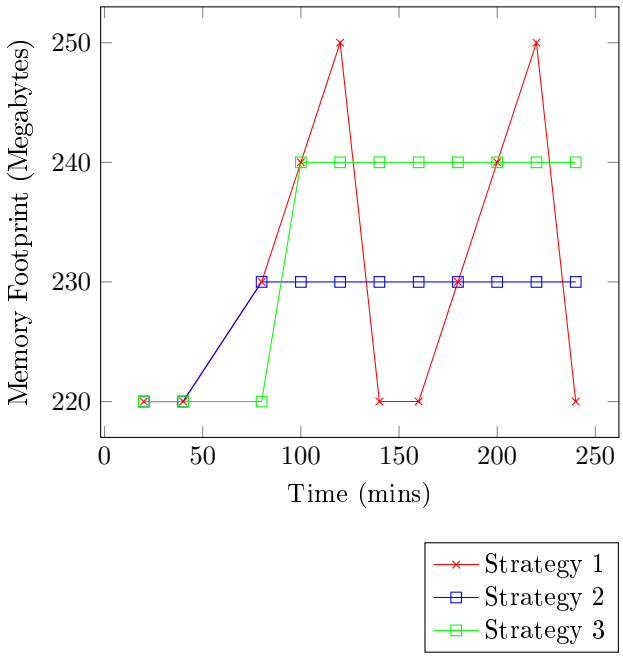


Figure 18: Minerva - Memory footprint on extraction

Property Path	elt*	elt{,5}	elt{,10}
s1	✔	✔	✔
s2	✔	✔	✔
s3	✔	✔	✔

Figure 19: Minerva - Tested Property Path Expressions

Layout Type	Storage Schema
Simple	3 tables each indexed by subjects, predicates and objects
Vertically Partitioned (VP)	For every unique predicate, two tables, each indexed by subjects and objects
Indexed	Six tables representing the six possible combinations of a triple namely, SPO, SOP, PSO, POS, OSP and OPS
Hybrid	Simple + VP layouts
Hash	Hybrid layout with hash values for nodes and a separate table containing hash-to-node mappings

Table 4: University of Texas - Identified Storage Alternatives

it took some time to train the users to familiarize themselves with the SPARQL query language syntax and how best to use the property path expression to query directed graph structures.

7.6 Other RDF/Columnar implementations

The University of Texas researched the possibility that HBase could support a high performance Triple Store. The paper identified five possible schema structures and subsequently tested each schema structure in turn. (see figure 4) Vaibhav Khadilkar [2012]

The paper concluded that:

- To support high performance triplet extraction it is better to use *wide* rows to contain triplets.
- It was not necessary to index literal values contained within an object.

Both IBM Research DB2 and University of Texas HBase implementation were unable to implement the following:

- Support for the SPARQL 1.1 property path extension.
- Provide any form of graph/context support.

8 New Developments

Oracle has two triple store implementations, where one utilizes the Oracle Spatial option which stores the triplets in a number of relational tables, the other implementation stores the triplets in a Oracle NOSQLDB key/value store. Only their NoSQLDB implementation comes with a native SPARQL optimizer.

IBM research has investigated the possibility of utilizing their DB2 database system to provide a platform as a triple store. IBM does state:

> "For future work, we are preparing a study on insertion, bulk load and update performance and we are planning to extend our system to support the SPARQL 1.1 standard (including property paths)." Mihaela A. Bornea [2012]

9 Conclusion

Columnar storage solutions such as HBase and Cassandra are suitable candidates to store triplets.The *proof-of-concept* implementation - Minerva - was able

to hold and retrieve unbounded tabular data. The performance characteristics of *s1* was such that it was not a suitable contender. Overall *s3* was the superior strategy for storing tabular data utilizing the **rdf list** structure because it did not need the separate column family required by *s2*. In addition, this strategy did not require the *wide rows* required by *s2*. In addition to tabular data, Minerva *s2* was able to contain any directed graph structure, process the LUBM query test suite and implement the SPARQL 1.1 property path expression. At the time of testing, Minerva conclusively outperformed currently available commercial and open source triple store implementations.

References

Tim Berners-Lee. Uri specification. w3c, 1991. URL `http://www.w3.org/Addressing/URL/uri-spec.html`.

DataStax. Cassandra technical manual, August 2014. URL `http://www.datastax.com/`.

W3C Working Group. Sparql query language for rdf, January 2008. URL `http://www.w3.org/TR/rdf-sparql-query/`.

W3C Working Group. Resource description framework (rdf), 2 2014. URL `http://www.w3.org/RDF/`.

Zhichao Xu Changhe Qiu Tengjiao Wang Lei Gai, Wei Chen. Towards efficient path query on social network with hybrid rdf management. 2014. URL `http://arxiv.org/pdf/1405.6500v2.pdf`.

G. Lausen C Pinkel M. Schmidt, T. Hornung. A sparql per-formance benchmark. *ICDE*, pages 222–223, 2009.

Anastasios Kementsietsidis Kavitha Srinivas Patrick Dantressangle Octavian Udrea Bishwaranjan Bhattacharjee Mihaela A. Bornea, Julian Dolby. Building an efficient rdf store over a relational database. Technical report, IBM Research, 2012. URL `https://cs.uwaterloo.ca/~gweddell/cs848/papers/Bornea.pdf`.

Sebastian Rudolph Pascal Hitzler, Markus Krotzsch. *Foundations of Semantic Web Technologies*. CRC Press, 2010.

Eric Prudhommeaux and Alexandre Bertails. A mapping of sparql onto conventional sql, 2010. URL `http://www.w3.org/2008/07/MappingRules/StemMapping`.

Bhavani Thuraisingham Paolo Castagna Vaibhav Khadilkar, Murat Kantarcioglu. Jenahbase: A distributed, scalable and efficient rdf triple store. *The University of Texas at Dallas*, 2012.

W3C. Sparql protocol for rdf, January 2008. URL `http://www.w3.org/TR/rdf-sparql-protocol/`.

W3C. Sparql 1.1 property paths, 2010. URL `http://www.w3.org/2009/sparql/docs/property-paths/Overview.xml`.

Factors Influencing Robustness and Effectiveness of Conditional Random Fields in Active Learning Frameworks

Mahnoosh Kholghi[1,2], Laurianne Sitbon[1], Guido Zuccon[1], Anthony Nguyen[2]
[1]Science and Engineering Faculty, Queensland University of Technology, Brisbane 4000, Queensland, Australia
[2]The Australian e-Health Research Centre, CSIRO, Brisbane 4029, Queensland, Australia
{m1.kholghi, laurianne.sitbon, g.zuccon}@qut.edu.au, anthony.nguyen@csiro.au

Abstract

Active learning approaches reduce the annotation cost required by traditional supervised approaches to reach the same effectiveness by actively selecting informative instances during the learning phase. However, effectiveness and robustness of the learnt models are influenced by a number of factors. In this paper we investigate the factors that affect the effectiveness, more specifically in terms of stability and robustness, of active learning models built using conditional random fields (CRFs) for information extraction applications. Stability, defined as a small variation of performance when small variation of the training data or a small variation of the parameters occur, is a major issue for machine learning models, but even more so in the active learning framework which aims to minimise the amount of training data required. The factors we investigate are a) the choice of incremental vs. standard active learning, b) the feature set used as a representation of the text (i.e., morphological features, syntactic features, or semantic features) and c) Gaussian prior variance as one of the important CRFs parameters. Our empirical findings show that incremental learning and the Gaussian prior variance lead to more stable and robust models across iterations. Our study also demonstrates that orthographical, morphological and contextual features as a group of basic features play an important role in learning effective models across all iterations.

Keywords: active learning, robustness, effectiveness, conditional random fields, Gaussian prior variance, concept extraction.

1 Introduction

Concept extraction is a significant initial step in any information extraction system and includes recognising meaningful entities and assigning them to predefined classes (e.g., person, organization, location; in the medical domain: problem, test, treatment) (Nadkarni et al., 2011). In this paper we use datasets and tasks in the clinical domain, where the concepts to be extracted are clinical concepts and the documents are clinical records.

The three main approaches to extract target concepts and entities from free text resources are dictionaries, rules and machine learning. Target entities usually appear as multi-token sequences in the text; these often cannot be captured directly using only lexical resources (Gurulingappa, 2012; Meystre et al., 2008; Roberts, 2012). For example in this sentence from the i2b2/VA 2010 dataset (Uzuner et al., 2011):

"*She had a workup by her neurologist and an MRI revealed a C5-6 disc herniation with cord compression and a T2 signal change at that level.*"

"*a C5-6 disc herniation*" is a multi-token concept of type "*problem*".

Manually creating resources or rules for dictionary and rule-based approaches is not only expensive and time-consuming, but also is a complex, error prone task. Additionally, these approaches are not adaptable and scalable to other domains and languages (Gurulingappa, 2012; Meystre, et al., 2008; Nadkarni, et al., 2011; Roberts, 2012).

Machine learning-based approaches have been extensively leveraged to extract concepts in several information extraction tasks (Jiang, 2012; Piskorski & Yangarber, 2013).

Since they have first been proposed in 2001, Conditional random fields (Lafferty et al., 2001) and in particular linear-chain CRFs, have shown the most promising results among other supervised machine learning algorithms to extract entities and concepts from text, in particular in the clinical domain (Suominen et al., 2013; Uzuner et al., 2008; Uzuner et al., 2010; Uzuner, et al., 2011). This motivates the use of linear chain CRFs in our active learning-based framework. CRFs approaches are supervised and therefore require fairly large amounts of high quality annotated data to build powerful statistical models. Creating the required annotated data to train a supervised model is laborious and expensive due to the necessary manual effort and the domain expert involvement.

Active learning (AL) was introduced to reduce the annotation costs across all supervised machine learning approaches (Settles, 2012) by selectively labelling informative instances and is a well-motivated and promising solution to address the problem of creating costly annotated datasets for training classifiers.

Active learning algorithms are advantageous in machine learning tasks where plenty of unlabelled samples are available and easy to access, but labelled data is scarce and expensive to be prepared. Active learning models are built in an iterative process, unlike other supervised machine learning models. As shown in Figure 1, a first model is built using a supervised algorithm on an initial labelled set, which represents a small portion of the whole annotated data (less than 1%). Then, in an iterative process, "informative" instances are selected using a

query strategy, removed from the unlabelled set and added to the training set to build a new model using the supervised algorithm. The process continues until a stopping point which depends on the task (e.g., reaching at least the same effectiveness as supervised approach). By building a model on informative instances rather than the other instances, the active learning approach guarantees that the highest effectiveness can be yielded by the model.

There are some important elements in the active learning process:

(1) When active learning is performed in real situations, a human annotator labels each selected informative instances just after they have been selected. In this paper, instead, we simulate this activity and use the gold standard annotation to label the selected instances.

(2) In each iteration of *standard* active learning, a model is built from scratch, i.e., without considering the model from the previous iteration. However, at each iteration, it is also possible to build a model by updating the model from the previous iteration in an *incremental* active learning setting.

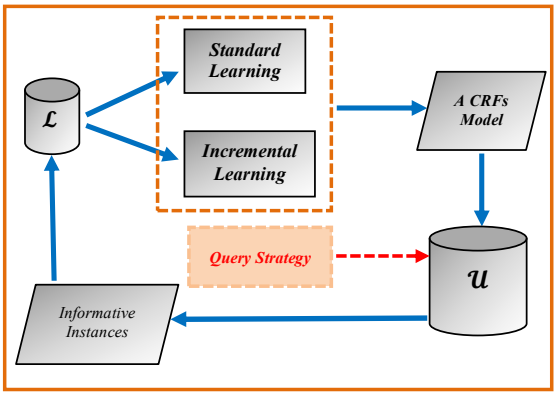


Figure 1: Standard vs. incremental active learning.

The main challenge of active learning approaches is to identify informative instances, and therefore it becomes essential to determine which selection criterion (also called query strategy) is the most suitable for a given task. A number of query strategies have been proposed, e.g., uncertainty sampling (Lewis & Catlett, 1994), query-by-committee (Seung et al., 1992), and information density (Settles & Craven, 2008). Uncertainty sampling (Lewis & Catlett, 1994) is currently the most widely used query strategy across active learning tasks and thus in this paper we will consider only uncertainty sampling to select data instances to label.

The goal of active learning is to maximize the effectiveness of the supervised machine learning model by minimizing the annotation effort. All supervised learning algorithms rely on a number of parameters that are typically tuned on a portion of the existing large set of annotated data (e.g., using cross validation). However, in an active learning framework there is less flexibility to build such tuned set of parameters, therefore the selected supervised algorithm needs to be as robust as possible to small changes in the training set and in its parameters, so that it can be used reliably in the active learning process. In particular, previous work (Nguyen & Patrick, 2014) has observed that some AL framework generated large variation in effectiveness of the models built during successive iterations, rendering the choice of a stopping point difficult. It is therefore essential to identify the parameters of the supervised learning model (here CRFs), the feature set used to train the model and the parameters of the AL framework (here standard vs. incremental) in order to establish what values are the most likely to lead to reliable and stable models. Settles and Craven (2008) have studied the effect of different AL query strategies on the effectiveness of concept extraction from text. They have demonstrated that uncertainty sampling methods (least confidence (Lewis & Catlett, 1994), margin (Scheffer et al., 2001), and entropy (Shannon, 1948)) are computationally the most efficient and, among the tested uncertainty sampling methods, least confidence and sequence entropy achieved better effectiveness compared to others. However, the factors that affect the stability and robustness of the AL models have not yet been investigated. Additionally, there has been no study to measure the impact of the Gaussian prior variance, one parameter of the CRFs model, on the robustness of the classifier.

In this paper we address the following questions: to what extent the AL models are reliable and robust? What factors affect the stability and robustness of the AL models? How different feature sets and parameter values of CRFs influence the robustness of the AL models? How incremental learning can help to build more reliable and robust models within the AL framework?

We answer these questions by conducting an intensive experimental evaluation on data from the i2b2/VA 2010 NLP challenge (Uzuner, et al., 2011) and the ShARe/CLEF 2013 eHealth Evaluation Lab (task 1) (Suominen, et al., 2013). The goal of these challenges is to extract concepts related to medical problems, tests and treatments and disorder mentions, respectively in i2b2/VA 2010 and ShARe/CLEF 2013. We rely on training data from these datasets to train active learning models and leverage the test data to evaluate the robustness of the built models.

The remainder of the paper is organized as follows: Section 2 introduces the feature set, the supervised CRFs approach and the active learning. Section 3 describes our experimental and evaluation settings. Results are reported in Section 4 and discussed in Section 5; Section 6 concludes the paper outlining directions of future investigation.

2 Active Learning Framework

In this section, we explain the features used to describe the data for classification. Then we briefly introduce CRFs and its parameter (Gaussian prior variance). Finally, we explain the query strategy and the incremental active learning settings.

2.1 Features for Conditional Random Fields

Figure 2 shows the feature categories that we use to inform the supervised learning algorithms in both supervised and active learning approaches.

The considered feature sets include rules implemented by regular expressions to identify acronyms, punctuations, capital letters and any combination of digits and letters; suffix and prefix characters with different length (up to 4); character 2-grams, 3-grams, and 4-grams; and a window of three previous and following words.

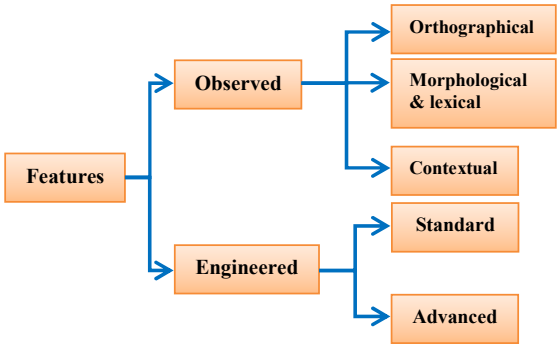


Figure 2: Feature groups.

Engineered features are extracted with the Stanford Part-Of-Speech (POS) tagger (Toutanova et al., 2003) to produce POS tags as standard engineered features. Semantic features comprising of SNOMED CT and UMLS semantic groups as advanced engineered features are obtained using the Medtex system (a medical NLP toolkit) (Nguyen et al., 2009). Here we map semantic group features to "*1*" and "*0*" (present and absent, respectively) for each token. To this aim, we first differentiate our target semantic types from all UMLS and SNOMED CT semantic groups. Target semantic types are specified based on the target concept types required to be extracted (*problem*, *test*, and *treatment* in the i2b2/VA 2010 dataset and *disorder* in ShARe/CLEF 2013). For example, the following UMLS semantic groups represent the disorder concepts: *Congenital Abnormality, Acquired Abnormality, Injury or Poisoning, Pathologic function, Disease or Syndrome, and Mental or Behavioural Dysfunction, Cell or Molecular Dysfunction, Experimental Model of Disease, Anatomical Abnormality, Neoplastic Process, Sign and Symptoms* (Pradhan et al., 2013). We then assign "*1*" to target semantic types and "*0*" to non-target semantic types.

2.2 CRFs and Gaussian Prior Variance

The concept extraction problem requires to assign a sequence of labels $\vec{y} = (y_1, \dots, y_n)$ to a sequence of input tokens $\vec{x} = (x_1, \dots, x_n)$.

Conditional random fields is a probabilistic method for extracting and labelling sequential data. CRFs naturally encode dependencies between different entities of a sequence and typically outperform other supervised learning algorithms (e.g., support vector machines (Joachims, 1998)) in sequence labelling tasks (Li et al., 2008). In this paper we use a first-order linear-chain CRFs as supervised learning algorithm within the active learning framework.

Conditional random fields models measure the conditional probability of the outputs ($\vec{y}$) based on the given inputs ($\vec{x}$) with a set of parameters θ:

$$P_\theta(\vec{y}|\vec{x}) = \frac{1}{Z_\theta(\vec{x})} exp\left(\sum_{i=1}^{n}\sum_{j=1}^{m}\lambda_j f_j(y_{i-1}, y_i, x_i)\right) \quad (1)$$

where $Z_\theta(\vec{x})$ is the normalization factor, $f_j(.)$ are feature functions, and $\theta = (\lambda_1, \dots, \lambda_m)$ represent the parameters to weight the corresponding features. Each $f_j(.)$ is the transition feature function between label state $i-1$ and i on the sequence $\vec{x}$ at position i.

The model parameters θ are estimated by penalized maximum log-likelihood L on some training data T (Tomanek & Hahn, 2009):

$$L(T) = \sum_{(\vec{x},\vec{y})\in T} log\, p(\vec{y}|\vec{x}) - \sum_{i=1}^{m}\frac{\lambda_i^2}{2\sigma^2} \quad (2)$$

Regularization is used to penalize weight vectors with large norm. The regularization parameter ($\frac{1}{2\sigma^2}$) specifies the intensity of the penalty. If θ is modelled using a Gaussian prior, the regularization can be seen as a maximum a posteriori estimation of θ (Lafferty, et al., 2001).

Gaussian prior variance is an important parameter in CRFs, because it prevents over-fitting thus allowing to build reliable and robust models. In particular, the Gaussian prior variance specifies the variance of the feature weights: when the Gaussian prior variance is large, the feature weights deviate more from zero. If the Gaussian prior variance is set to infinite, then the feature weights can assume any real value. The latter case occurs when the values of the feature weights of the learnt model are not constrained by a limit; this results in over-fitting. A generalizable model should then have small feature weights values.

In our experiments, we investigate the effect of the Gaussian prior variance on the robustness and the effectiveness of the learnt AL models.

2.3 Incremental Active Learning and Query Strategy

As shown in Figure 1, in a standard active learning framework, where a pool of unlabelled instances is available, first a supervised model (Θ) is built on an initial small, randomly selected labelled set. Then a batch of informative instances (B) is selected using the query strategy $\varphi^{\Theta}(u_i)$. The query strategy estimates the informativeness of an unlabelled instance $u_i \in \mathcal{U}$ based on the model Θ. The selected batch of instances is removed from the unlabelled set and added to the labelled set to train a new model "from scratch", i.e., without considering the parameters from the previous model. This process continues until a stopping criterion is satisfied.

In an incremental setting, all parameter values, including the feature weights, are kept to be updated in a new iteration. This significantly reduces the training time,

because a model does not have to be trained from scratch at each iteration and the parameters are already initialized in the previous step (Figure 3).

In our experiment, we investigate the difference in stability and robustness of the standard and incremental active learning approaches.

Input:
Set of labeled instances $\mathcal{L}$, set of unlabeled instances $\mathcal{U}$, least confidence query strategy $\varphi_{LC}^{\Theta}(u_i)$, number of instances to be selected in each iteration (batch size) B.

Algorithm:
While stopping criterion not met

1- train Θ model on $\mathcal{L}$
(a) standard learning: train model Θ on $\mathcal{L}$ from scratch
OR
(b) incremental learning: update the model Θ from previous iteration with new $\mathcal{L}$
2- **for** all $u_i \in \mathcal{U}$ **do**:
$\varphi_{LC}^{\Theta}(u_i) = 1 - P_\theta(\vec{y}^*|\vec{u_i})$ (Equation 1 and 4)
End
3- **for** $b = 1$ **to** B **do**:
$u_b = \arg max_{u_i \in \mathcal{U}} \varphi_{LC}^{\Theta}(u_i)$
$\mathcal{L} = \mathcal{L} \cup (u_b, label(u_b))$
$\mathcal{U} = \mathcal{U} - u_b$
End

End

Figure 3: The AL framework based on least confidence and incremental vs. standard learning.

2.3.1 Query Strategy

At each iteration of the active learning loop, we use uncertainty sampling to query the unlabelled instances and select the most informative instances. Informativeness is estimated according to how uncertain the model is about the label of the unlabelled instance (i.e., the classification uncertainty of the model). Instances with the highest uncertainty are selected for labelling and inclusion in the labelled set used for training in the following iteration.

We use Least Confidence (LC) as it is known as one of the most effective uncertainty sampling methods. LC uses the confidence of the latest model Θ with parameters θ in predicting the label $\vec{y}$ of a sequence $\vec{x}$ (Culotta & McCallum, 2005):

$$\varphi_{LC}^{\Theta}(\vec{x}) = 1 - P_\theta(\vec{y}^*|\vec{x}) \quad (3)$$

The confidence of the CRFs model is estimated using the posterior probability described in Equation (1) and $\vec{y}^*$ is the most likely label sequence obtained using the Viterbi algorithm:

$$\vec{y}^* = arg\ max_{\vec{y} \in Y^n} exp\left(\sum_{i=1}^{n}\sum_{j=1}^{m} \lambda_j f_j(y_{i-1}, y_i, x_i)\right) \quad (4)$$

The algorithms describing the active learning framework for both the standard and the incremental settings are shown in Figure 3 using the least confidence query strategy.

3 Experimental Framework

The experimental framework we propose aims to provide a study of the factors that impact stability and robustness, as well as to investigate the effectiveness of the learnt active learning models. Our experimental framework consists of three consecutive steps, as shown in Figure 4:

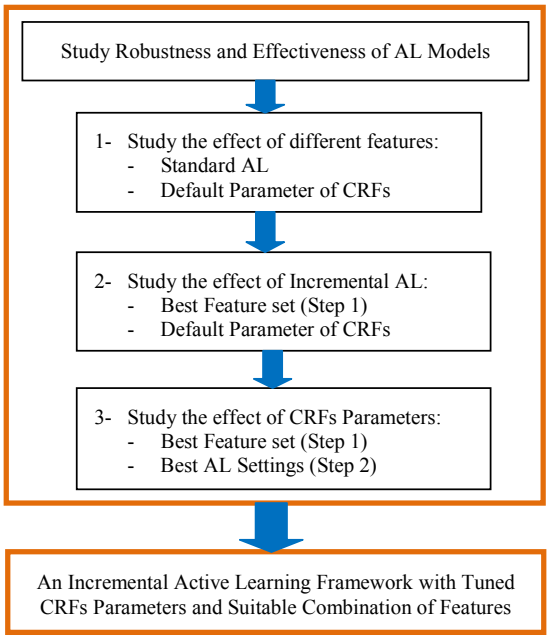


Figure 4: Experimental framework.

1. Investigate the impact of different feature combinations on robustness and effectiveness of the learnt models, when considering the default parameter values for CRFs and the standard AL.
2. Investigate the impact of incremental AL vs. standard AL on robustness and effectiveness of the learnt models, leveraging the best feature combination and the default CRFs parameters.
3. Investigate the impact of the CRFs parameters on robustness and effectiveness of the learnt models when considering the feature set and AL setting that showed the highest effectiveness at steps 1 and 2.

3.1 Dataset

Our experiments leverage data and task definitions from the i2b2/VA 2010 NLP task and the ShARe/CLEF 2013 eHealth Evaluation Lab (task 1). We use the same split of train and test sets defined in the original datasets.

The i2b2/VA 2010 NLP task (Uzuner, et al., 2011) requires to extract medical problems, tests and treatments from clinical reports. The reports used in this task are a combination of discharge summaries and progress notes supplied by three different health providers. The training and testing sets include 349 and 477 reports, respectively. These reports are organized as a collection of phrases and sentences (each report file containing a phrase or sentence per line). After dividing the dataset into phrases and sentences, we obtain 30,673 and 45,025 sequences in the training and test set, respectively.

Table 1: The effectiveness of the supervised and the active learning approach (the one exhibiting the highest effectiveness) with respect to O, A, B, and C feature sets (P = Precision, R = Recall, and F1 = F1-measure) (a) i2b2/VA 2010 (b) ShARe/CLEF 2013.

	Supervised Learning			Active Learning		
	P	R	F1	P	R	F1
O	0.252	0.36	0.296	0.625	0.544	0.581
A	0.725	0.624	**0.671**	0.728	0.63	**0.676**
B	0.416	0.429	0.428	0.64	0.561	0.598
C	0.501	0.466	0.483	0. 654	0.552	0.599

(a)

	Supervised Learning			Active Learning		
	P	R	F1	P	R	F1
O	0.266	0.186	0.219	0.334	0.178	0.232
A	0.419	0.296	**0.347**	0.424	0.298	**0.35**
B	0.257	0.184	0.214	0.324	0.174	0.227
C	0.233	0.182	0.204	0.304	0.174	0.222

(b)

The ShARe/CLEF 2013 eHealth Evaluation Lab (task 1) (Suominen, et al., 2013) requires to extract and identify disorder mentions from clinical free-text notes. The dataset for this task consists of 200 training and 100 test documents, including discharge summaries, electrocardiogram, echocardiogram, and radiology reports from a U.S. intensive care unit. As for the i2b2/VA 2010 dataset, we divided each report from this dataset into sequences based on line breaks. Overall, this produced 2,742 and 2,325 sequences in the training and test set, respectively.

3.2 Evaluation Methodology

We use the MALLET toolkit (McCallum, 2002) to train CRFs classifiers. For AL, the initial labelled set is formed by randomly selecting 1% of the training data. The batch size is set to 200 sequences for i2b2/VA 2010 and 30 for ShARe/CLEF 2013 across all experiments, leading to a total of 153 and 91 batches[1], respectively.

Concept extraction effectiveness is measured by Precision, Recall, and F1-measure. Evaluation metrics are computed on test data using the multi-segmentation evaluator implemented in the MALLET toolkit, which considers segments that span across multiple tokens.

The robustness and stability of AL models are analysed by examining the learning curves of the AL approaches across batches. For each batch, learning curves plot the F1-measure achieved by the AL classifier trained with the data contained in the labelled set up to the considered batch.

To further analyse the robustness of AL models, we also perform 10-fold cross validation experiments on the training data. In these experiments the training set is split in ten random sets; for a given fold, nine are used as labelled train data and one as test data. The effectiveness of active learning in each batch is averaged across ten test such folds.

4 Results

Section 4.1 reports the impact of different feature sets on the robustness of the learnt active learning models. Section 4.2 examines how incremental active learning affects the robustness of the learnt models. Finally, Section 4.3 analyses the impact of the CRFs parameter (Gaussian prior variance) on robustness and effectiveness of the active learning models.

4.1 Effect of Feature Sets

We define the following feature sets to evaluate the effect of different features on supervised and active learning approaches:

- **O** : Only token itself as a feature;
- **A** : Observed features;
- **B** : Standard engineered features (POS tags);
- **C**: Advanced engineered features (SNOMED CT and UMLS semantic groups).

We first consider each **O**, **A**, **B**, and **C** as a separate feature set within the supervised and active learning approaches. Figure 5 demonstrates the learning curve for the active learning approach against the supervised learning effectiveness with different feature groups (**O**, **A**, **B**, and **C**) for both the i2b2/VA 2010 and ShARe/CLEF 2013 datasets. These experiments consider the standard active learning approach with the default parameter values for the MALLET CRFs. Table 1 reports the highest effectiveness achieved by the active learning approach across the batches using different groups of features (**O**, **A**, **B**, and **C**) for both datasets.

The results shown in Figure 5 and Table 1 show that token features, POS tags and semantic features, when used alone, provide similar effectiveness on the ShARe/CLEF 2013 dataset, while POS tags and semantic features are generally superior than tokens alone in the i2b2/VA 2010 dataset. However, all these feature sets provide inferior effectiveness when compared to the observed feature set (**A**). These results suggest that basic linguistic features (**A**) including orthographical, morphological and lexical, and contextual features, are generally more effective than other feature sets (**O**, **B**, and **C**) in both supervised and active learning settings.

From Table 1 we can further observe that the best active learning setting adds improvements for each feature set with respect to the supervised approach, e.g. **O** in i2b2. Also, the best effectiveness of both supervised and active learning approaches is achieved when leveraging observed features (**A**) in learning process.

Figure 5 shows that AL effectiveness varies greatly across batches. Specifically, when the highest effectiveness is achieved, standard AL does not seem to guarantee that effectiveness to be maintained if more

[1] The choice of batch size was done with respect to the number of sentences in each dataset. While this is a parameter which may ultimately influence effectiveness and stability of the learnt model, we do not explore it in the paper and leave it for future work.

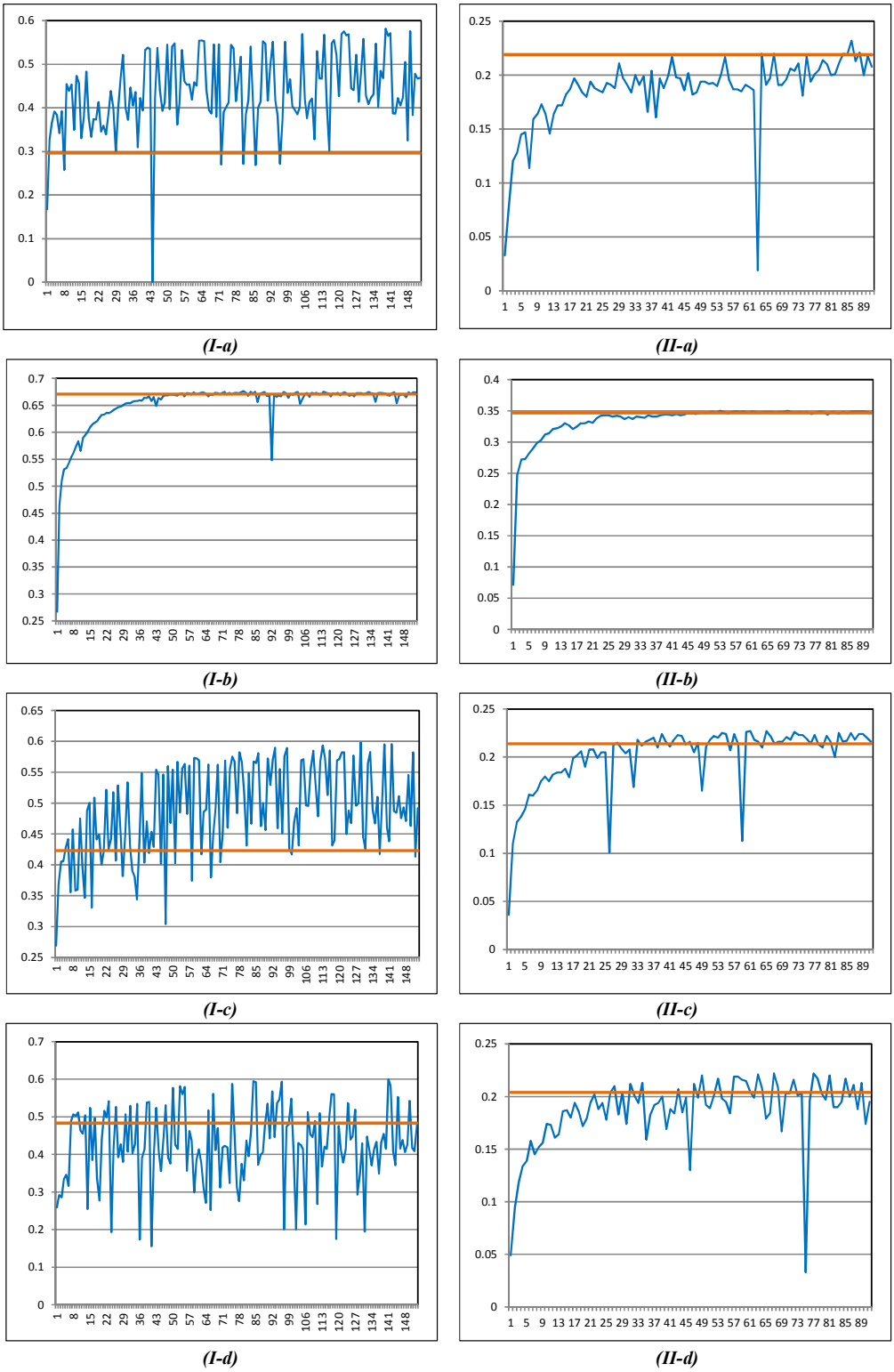
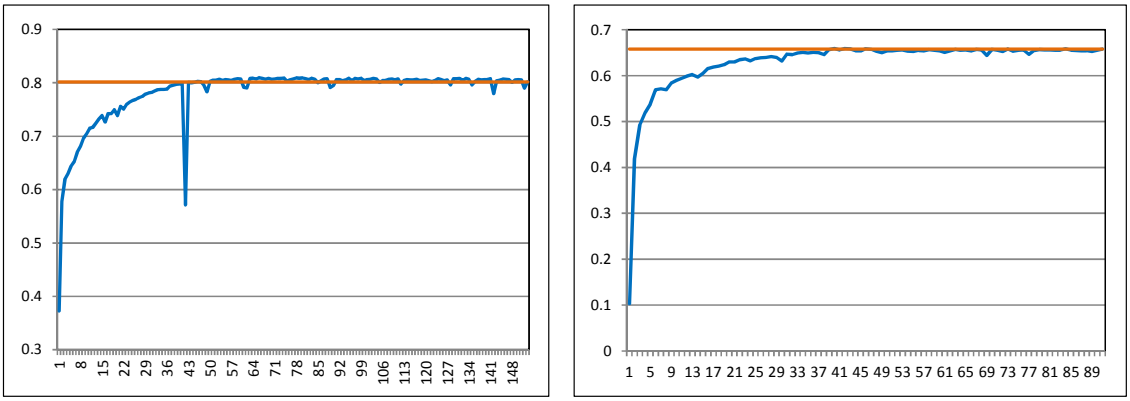


Figure 5: Active learning curves across batches (blue line) and supervised effectiveness (orange straight line). The horizontal axis reports the number of batches used to train the classifier in the AL setting, while the vertical axis reports the value of F1-measure obtained by applying the classifier from the corresponding batch (or the whole training data in the case of the supervised classifier) on the test data (I: i2b2/VA 2010 dataset, II: ShARe/CLEF 2013 dataset, a: O, b: A, c: B, d: C).



(a) **(b)**

Figure 6: Active learning curves across the batches (blue line) and supervised effectiveness (orange straight line), using the combination of whole features. The horizontal axis reports the number of batches used to train the classifier in the AL setting, while the vertical axis reports the value of F1-measure obtained by applying the classifier from the corresponding batch (or the whole training data in the case of the supervised classifier) on the test data (a) i2b2/VA 2010 (b) ShARe/CLEF 2013.

batches are included in the training data, i.e. there are substantial fluctuations (and thus instability) in active learning curve. A sudden decrease in effectiveness between a batch B_i and the subsequent batch B_j suggests that the model learnt on data from up to batch B_i is over-fitted to that labelled set. Hence, the learnt model is not reliable for selecting the informative instances that form the next batch B_j from the unlabelled data. On the other hand, the active learning curve produced by representing data using the observed feature set **A** is smoother than the learning curves observed for other feature sets. This observation shows that feature set **A** leads not only to better effectiveness (Table 1), but also to more robust active learning models that exhibit stability across batches (Figure 5 (I-b and II-b)).

Next, we explore which combination of features provides the highest effectiveness in the active learning settings.

Table 2: The effectiveness of the supervised and the best active learning approach (the one exhibiting the highest effectiveness) using all features (A, B, and C) (P = Precision, R = Recall, and F1 = F1-measure) (a) i2b2/VA 2010 (b) ShARe/CLEF 2013.

	Supervised Learning			Active Learning		
	P	**R**	**F1**	**P**	**R**	**F1**
All	0.816	0.788	0.802	0.824	0.795	0.809

(a)

	Supervised Learning			Active Learning		
	P	**R**	**F1**	**P**	**R**	**F1**
All	0.759	0.581	0.658	0.763	0.58	0.659

(b)

Table 2 and Figure 6 show that combining all considered feature sets (**O**, **A**, **B**, and **C**) provides higher effectiveness than using the individual feature sets alone.

In addition, combining feature sets improves the stability across the active learning batches in both datasets. However, the shapes of the active learning curves suggest that there are other factors, along with the feature sets, that contribute to the robustness and stability of the learnt models.

Figure 7 reports the results obtained when using 10-fold cross validation on the training data for both datasets and when all feature sets are used. In this experiment, active learning effectiveness values are averaged across the testing folds.

The active learning curves reported in Figure 7 are similar to those in Figure 6. Thus, the cross-validation experiments confirm what suggested by the train-test experiments: when all feature sets are combined, the models built using AL are more robust than those built using only one feature set.

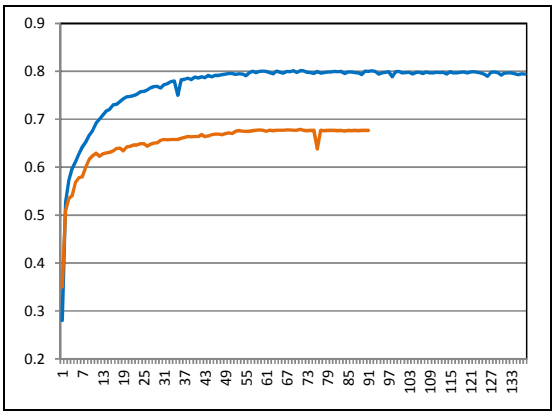
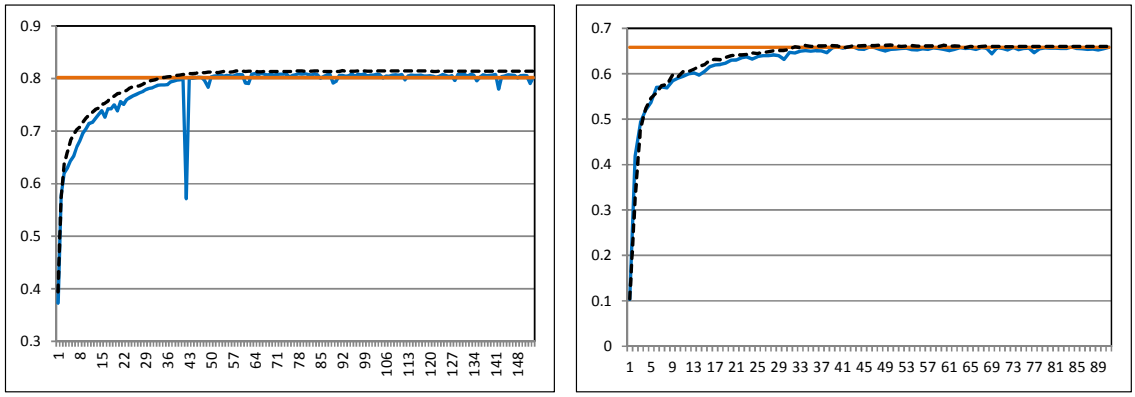


Figure 7: 10-fold cross validation results across the active learning batches on i2b2/VA 2010 (blue curve) and ShARe/CLEF 2013 (orange curve) datasets. The horizontal axis corresponds to the number of batches used for training and the vertical axis reports F1-measure values.



(a) (b)

Figure 8: Standard vs. Incremental active learning (ALCE vs. InALCE). The dashed black curve and blue curve represent the effectiveness of InALCE and ALCE, respectively, and the orange line represents the effectiveness of the supervised classifier. The horizontal axis reports the number of batches used to train the classifier in the AL setting, while the vertical axis reports the value of F1-measure obtained by applying the classifier from the corresponding batch (or the whole training data in the case of the supervised classifier) on the test data (a) i2b2/VA 2010 (b) ShARe/CLEF 2013.

4.2 Effect of Incremental Learning

In this section, we aim to study the effect of standard learning vs. incremental learning on stability and effectiveness of the learnt models within the active learning framework. We leverage the combination of all features and we use the default value of the MALLET CRFs parameters. Incremental active learning is applied throughout the training set, i.e., the values of CRFs parameters are updated in each iteration of active learning. We call this new setting the Incremental Active Learning for Concept Extraction (InALCE) and compare it with the standard active learning framework (ALCE).

Figure 8 reports the F1-measure achieved by ALCE and InALCE compared to the F1-measure obtained by the supervised classifier. Incremental active learning achieves higher effectiveness compared to standard active learning with less training data (i.e. requiring less batches): we suggest that this is because in the incremental active learning approach, the parameters of the learnt CRFs are maintained and updated in the subsequent iteration. While, in standard active learning the CRFs model is built from scratch at each iteration. Incremental learning is less prone to sudden changes in the training data, in particular negative changes. Therefore, the learnt models using incremental active learning in the InALCE framework are more robust rather than the models built using standard active learning in ALCE, as suggested by the smoother learning curve of InALCE when compared to those generated by ALCE. It subsequently leads to more accurate selection of informative instances in each iteration of InALCE and stability across the batches.

4.3 Effect of Gaussian Prior Variance in CRFs setting

As described in Section 2.2, the Gaussian prior variance is an important CRFs parameter as its value influences the robustness of the learnt model. In this section, we investigate the impact of this parameter on the robustness of the AL models built within the incremental AL approach using all features sets (Section 4.1), as this setting provided the highest effectiveness.

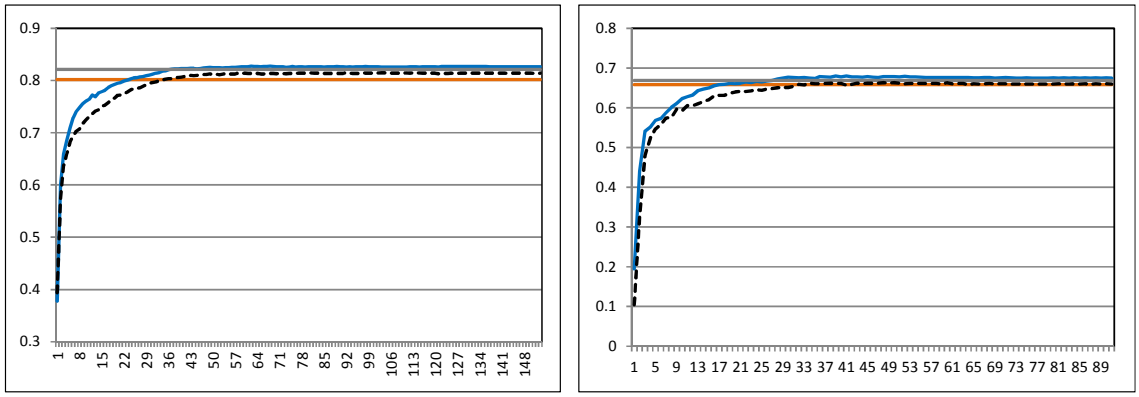
The default value for the Gaussian prior variance in MALLET is 10. Smaller values for the Gaussian prior variance limit the deviation of the feature weights from zero: this often avoids over-fitting. However, if the Gaussian prior variance is zero, then it will force all weights to be zero. To explore the impact of this parameter on the stability of the learnt models, we perform an empirical evaluation of different Gaussian prior variance values between 1 and 10 (with a step of 2). The empirical results suggest that a Gaussian prior variance value of 1 leads to the highest effectiveness for both supervised and active learning. This parameter value also exhibit the smoother AL learning curve with respect to the other tested values, resulting in more robust AL models.

Figure 9 reports the effectiveness for both supervised and active learning in un-tuned (Gaussian prior variance set to the default value of 10) and tuned settings (Gaussian prior variance set to 1). As shown in the figure, incremental active learning with tuned parameter provides the highest effectiveness and the most robustness across batches.

5 Discussion

The results reported in Section 4 show that feature sets, incremental learning and CRFs parameters (specifically, the Gaussian prior variance) play an important role in the stability, robustness and effectiveness of the active learning models learnt across batches.

In this paper we showed that the observed feature set (**A**), which includes orthographical, lexical and morphological, and contextual features, significantly increases the effectiveness of both supervised and active learning classifiers. While POS tags and semantic feature lead to poor effectiveness and unreliable models when used individually, they are useful to augment the data



(a) **(b)**

Figure 9: Incremental non-tuned (dash black curve) vs. Incremental (blue curve) active learning with tuned CRFs parameters (InALCE vs. InALCE-Tun) against the un-tuned (orange line) and tuned (grey line) supervised effectiveness. The horizontal axis reports the number of batches used to train the classifier in the AL setting, while the vertical axis reports the value of F1-measure obtained by applying the classifier from the corresponding batch (or the whole training data in the case of the supervised classifier) on the test data (a) i2b2/VA 2010 (b) ShARe/CLEF 2013.

representation and the highest effectiveness is achieved when combining all feature sets.

Our analysis also demonstrated that incremental active learning not only reduces the amount of training data required (also compared to standard AL), but also leads to more robust and more effective models compared to the standard setting.

Finally, we have shown that the Gaussian prior variance used in CRFs influences both the effectiveness and the stability of the active learning models. The empirical results have demonstrated that tuning this parameter increases the effectiveness of both supervised and active learning models, but it has a minor effect on the stability compared to the influence of feature sets and the incremental setting.

6 Conclusion and Future work

In this paper, we have established that the robustness and the effectiveness of the active learning models for medical concept extraction depend on: feature set, incremental learning setting, and tuning of the supervised classifier parameters. This was demonstrated by conducting a large empirical evaluation on two medical datasets, the i2b2/VA 2010 and the ShARe/CLEF 2013 (task1). The evaluation showed that basic linguistic and lexical features increase the stability and robustness of the learnt models compared to domain specific semantic features. We also studied the effect of incremental learning and the Gaussian prior variance (CRFs parameter), observing that they increase both the effectiveness and the stability of the learnt models on both datasets.

This work represents the first step in analysing the stability and robustness of the learnt active learning models: further work is required to examine the influence of the considered factors for other types of concept extraction tasks.

7 References

Culotta, A., & McCallum, A. (2005). Reducing labeling effort for structured prediction tasks. *Proceedings of the National Conference on Artificial Intelligence (AAAI)* (pp. 746–751): AAAI Press.

Gurulingappa, H. (2012). *Mining the medical and patent literature to support healthcare and pharmacovigilance* (Ph.D. dissertation). University of Bonn, Bonn, Germany.

Jiang, J. (2012). Information extraction from text. In *Mining Text Data* (pp. 11-41): Springer.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. *Proceedings of the 10th European Conference on Machine Learning (ECML-98)* (pp. 137-142): Springer-Verlag.

Lafferty, J. D., McCallum, A., & Pereira, F. C. N. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *Proceedings of the Eighteenth International Conference on Machine Learning (ICML)* (pp. 282-289). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Lewis, D. D., & Catlett, J. (1994). Heterogenous Uncertainty Sampling for Supervised Learning. *Proceedings of the 18th International Conference on Machine Learning* (pp. 148-156): Morgan Kaufmann.

Li, D., Kipper-Schuler, K., & Savova, G. (2008). Conditional random fields and support vector machines for disorder named entity recognition in clinical texts. *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing* (pp. 94-95). Stroudsburg, PA, USA: Association for Computational Linguistics.

McCallum, A. K. (2002). MALLET: A Machine Learning for Language Toolkit. Retrieved from http://mallet.cs.umass.edu

Meystre, S. M., Savova, G. K., Kipper-Schuler, K. C., & Hurdle, J. F. (2008). Extracting information from textual documents in the electronic health record: a review of recent research. *IMIA Year Book of Medical Informatics, 47*(Suppl 1), 128-144.

Nadkarni, P. M., Ohno-Machado, L., & Chapman, W. W. (2011). Natural language processing: an introduction. *Journal of the American Medical Informatics Association, 18*(5), 544-551.

Nguyen, A. N., Lawley, M. J., Hansen, D. P., & Colquist, S. (2009). A simple pipeline application for identifying and negating SNOMED clinical terminology in free text. *Proceedings of the Health Informatics Conference (HIC)* (pp. 188-193): Health Informatics Society of Australia (HISA).

Nguyen, D. H. M., & Patrick, J. D. (2014). Supervised machine learning and active learning in classification of radiology reports. *Journal of the American Medical Informatics Association*.

Piskorski, J., & Yangarber, R. (2013). Information extraction: Past, present and future. In T. Poibeau, H. Saggion, J. Piskorski & R. Yangarber (Eds.), *Multi-source, Multilingual Information Extraction and Summarization* (pp. 23-49): Springer Berlin Heidelberg.

Pradhan, S., Elhadad, N., South, B., Martinez, D., Christensen, L., Vogel, A., Suominen, H., Chapman, W., & Savova, G. (2013). Task 1: ShARe/CLEF eHealth Evaluation Lab 2013. In *Online Working Notes of CLEF, CLEF 2013*.

Roberts, A. (2012). *Clinical information extraction: lowering the barrier* (Ph.D. dissertation). University of Sheffield, Sheffield, United Kingdom.

Scheffer, T., Decomain, C., & Wrobel, S. (2001). Active hidden markov models for information extraction. *Proceedings of the International Conference on Advances in Intelligent Data Analysis (CAIDA)* (pp. 309-318): Springer-Verlag.

Settles, B. (2012). *Active Learning* (Vol. 6): Morgan & Claypool Publishers.

Settles, B., & Craven, M. (2008). An analysis of active learning strategies for sequence labeling tasks. *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (pp. 1070-1079): Association for Computational Linguistics.

Seung, H. S., Opper, M., & Sompolinsky, H. (1992). Query by committee. *Proceedings of the fifth annual workshop on Computational learning theory* (pp. 287-294). 130417: ACM.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal, 27*, 379–423, 623–656.

Suominen, H., Salanterä, S., Velupillai, S., Chapman, W., Savova, G., Elhadad, N., Pradhan, S., South, B., Mowery, D., Jones, G. F., Leveling, J., Kelly, L., Goeuriot, L., Martinez, D., & Zuccon, G. (2013). Overview of the ShARe/CLEF eHealth Evaluation Lab 2013. In P. Forner, H. Müller, R. Paredes, P. Rosso & B. Stein (Eds.), *Information Access Evaluation. Multilinguality, Multimodality, and Visualization* (Vol. 8138, pp. 212-231): Springer Berlin Heidelberg.

Tomanek, K., & Hahn, U. (2009). Semi-supervised active learning for sequence labeling. *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2* (pp. 1039-1047): Association for Computational Linguistics.

Toutanova, K., Klein, D., Manning, C. D., & Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology* (Vol. 1, pp. 173-180): Association for Computational Linguistics.

Uzuner, Ö., Goldstein, I., Luo, Y., & Kohane, I. (2008). Identifying patient smoking status from medical discharge records. *Journal of the American Medical Informatics Association, 15*(1), 14-24.

Uzuner, Ö., Solti, I., & Cadag, E. (2010). Extracting medication information from clinical text. *Journal of the American Medical Informatics Association, 17*(5), 514-518.

Uzuner, Ö., South, B. R., Shen, S., & DuVall, S. L. (2011). 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association, 18*(5), 552-556.

Dynamic Class Prediction with Classifier Based Distance Measure

Şenay Yaşar Sağlam[1] **W. Nick Street**[2]

[1] University of Iowa,
108 John Pappajohn Business Building,
Iowa City, IA 52242, U.S.A.
Email: senay-yasarsaglam@uiowa.edu

[2] University of Iowa,
108 John Pappajohn Business Building,
Iowa City, IA 52242, U.S.A.
Email: nick-street@uiowa.edu

Abstract

Combining multiple classifiers (ensemble of classifiers) to make predictions for new instances has shown to outperform a single classifier. As opposed to using the same ensemble for all data instances, recent studies have focused on dynamic ensembles in which a new ensemble is chosen from a pool of classifiers specifically for every new data instance. We propose a system for dynamic class prediction based on a new distance measure to evaluate the distance among data instances. We first map data instances into a space defined by the class probability estimates from a pool of two-class classifiers. We dynamically pick classifiers (features) to be used and the k-nearest neighbors of a new instance by minimizing the distance between the neighbors and that instance in a two-step framework. Results of our experiments show that our measure is effective for finding similar instances and our framework helps making more accurate predictions.

Keywords: Classification, Dynamic Ensembles, Confidence, Probability Estimates, Distance Measures, k-NN.

1 Introduction

An ensemble of classifiers consists of a set of trained classifiers whose individual decisions are combined to classify new instances. Most existing methods construct static ensembles, in which only one ensemble is chosen from the pool and used for all new instances. Recently, there have been studies in which each new data instance is treated individually. Since different instances are often associated with different classification difficulties, it is hypothesized that using different classifiers for the classification task rather than a single static ensemble of classifiers can improve performance. In this study, we propose a method to make dynamic predictions to address the following questions:

- To find k-nearest neighbors of a new instance, should original feature space or classifier space be used?
- Is using class probability estimates better than just classifiers' predictions to find similar instances?
- Is classifiers' performance on the validation instances helpful when finding similar instances?
- Should all of the classifiers in the pool be used for computing similarity between instances? If not, which criteria should be considered to eliminate certain classifiers?
- After neighbors are found, should we use them to form an ensemble or to make a prediction?

In Section 2, we summarize the earlier studies regarding ensemble of classifiers. In Sections 3 and 4, the general structure of the system and its running time are explained. In Sections 5 and 6, we explain proposed and baseline distance measures. We carry out several experiments to evaluate performance of our framework. The results are presented in Section 7. Finally, Section 8 concludes the paper.

2 Literature Review

Ensembles have received great attention over the past two decades because averaging errors of multiple predictors increases overall accuracy. Ensemble generation methods can use sampling to create different training sets, use different training parameters for the learning algorithm, or employ different learning algorithms on the same training set to generate a diverse set of classifiers (as there is no point in having the same classifier multiple times). Some well-known algorithms for creating ensembles include *Bagging* (Breiman (1996)), *Boosting* (Freund & Schapire (1996)), *Random Subspace Method* (Ho (1998)), and *Stacking* (Wolpert (1992)). Recently, *Overproduce and Choose strategy*, in which a large initial pool of candidate classifiers is produced and then a classifier or a subset of classifiers is selected, has been adopted in several studies (Didaci & Giacinto (2004), Dos Santos et al. (2008), Giacinto & Roli (2001), Ko et al. (2008), Woods et al. (1997), Zhang et al. (2006)).

Several factors contribute to the generalization ability of an ensemble. The first is the accuracy of the base classifiers. It is easy to verify that constructing an ensemble with the most accurate classifiers will result in good generalization error. However, since the classifiers require uncorrelated errors for the ensemble to be effective, diversity among them is also important. In other words, combining accurate and diverse classifiers may boost the performance of an ensemble if they have uncorrelated errors (Tumer & Ghosh (1996)). Meanwhile, posterior class probability estimates returned by classifiers have also been considered. For example, confidence measures based on ensembles' vote margin is proposed by Dos Santos et al. (2008) to improve the quality of their solutions or to be used as an input to a meta classifier in Stacking (Menahem et al. (2009), Ting & Witten (1997)).

Earlier studies regarding this dynamic scheme focus on selecting a classifier based on different features or different regions of the instances, depending on the similarities among them (Didaci et al. (2005), Didaci & Giacinto (2004), Giacinto & Roli (2001), Woods et al. (1997)). In dynamic *classifier* selection, a single predictor is chosen based on its likelihood to correctly predict the test pattern (Didaci et al. (2005)). For example, dynamic classifier selection approaches based on estimating local competence in selection regions defined by k-nearest neighbors, also known as the *k-Nearest Neighbor (k-NN) method*, have been proposed by Didaci et al. (2005), Didaci & Giacinto (2004), Giacinto & Roli (2001), Woods et al. (1997). However, unlike the k-NN algorithm, which uses the prediction values from k instances that are "closest" or "most similar" to the new data instance, the prediction of the most competent classifier in the region is used to make the decision.

Similar to the static classifier selection methods, the drawback of dynamic classifier selection methods is that the choice of a single individual classifier over the rest depends on how much we trust in that classifier's performance. If that classifier makes an incorrect decision, we will not be able to correct that decision. Therefore, a dynamic ensemble selection approach has been proposed by later studies (Dos Santos et al. (2008), Ko et al. (2008), Cavalin et al. (2010)). These studies focus on dynamic ensemble selection rather than a single classifier selection to overcome this drawback. Dos Santos et al. populate a set of candidate ensembles from a large initial pool of candidate classifiers, then choose an ensemble from that set for each new data instance based on candidate ensembles' confidence on this instance. Dos Santos et al. define confidence as a measure of the ensemble, based on vote margin, and use it as a second-level objective after optimizing on accuracy and diversity.

Ko et al. propose a method which, for each new data instance, finds its nearest k neighbors in the validation set using the original feature set, and dynamically chooses an ensemble based on the estimated accuracy of the classifiers in this local region of the new data instance. Two different versions of this method are proposed by Ko et al. (2008) in terms of how classifiers are chosen to form an ensemble for the new instance: KNORA-Eliminate and KNORA-Union. KNORA-Eliminate uses the classifiers that correctly classify every instance in the local region of the new instance. However, in KNORA-Union method classifiers that correctly classify at least one of the neighbors are added to the ensemble and each classifier submits a vote for each neighbor it classifies correctly to predict the new instance. Following this approach, Cavalin et al. adopt a hybrid framework and employ the confidence measure defined by Dos Santos et al. (2008) to build the ensembles. However, when the confidence value is not enough, Cavalin et al. search for the "closest" or "most similar" instance to the new instance in the validation set and assign its label to the new instance. Similarity measures are defined based on assigned class labels. We agree that similarity should be based on base classifiers' outputs as we will not gain much by using the same information (same feature set) used to train the classifiers. However, using predictions of the classifiers alone will not provide enough information to accurately define similarity between points. The KNORA method (Ko et al. (2008)) has been improved by Vriesmann et al. (2012). To specify the local region of a new data point, Vriesmann et al. investigate the effects of using a different distance measure on accuracy and conclude that the choice of distance measure has no effect on the performance of KNORA. Furthermore, different strategies for combining the information obtained from k neighbors of the new instance and the output of KNORA are adopted. Based on the experimental results, Vriesmann et al. suggest that additional information provided by the k-NN improves the performance of KNORA.

With KNORA, Ko et al. indicate that using neighbor information of a new data point even for constructing an ensemble can prove useful. However, it has been shown that using neighbors of a new data instance not just for constructing the ensemble but in fusion with that ensemble's decision for that instance enhances the performance (Cavalin et al. (2010), Vriesmann et al. (2012)). In particular, Vriesmann et al. find the neighbors of a new data instance based on the original feature space that is also used to train classifiers in the pool. This indeed duplicates the information at hand and increases the performance of the system to a certain point. Cavalin et al. address this issue by defining the similarity between instances based on classifiers' prediction on the validation and new data instances. However, classifiers' probability estimates are more informative compared to using the predictions alone. This is because they provide information on not only whether the classifier assigns the same labels to these instances, but also how confident it is with its decisions.

3 Proposed Framework

The prediction probability returned by a classifier can be considered as a measure of proximity of a data instance to the decision boundary. This measure can be used to compare multiple instances and to decide whether that particular classifier considers those data instances similar. Our proposed framework consists of two main steps: a static step and a dynamic step. We illustrate both steps as a flow chart in Figure 1.

In the static step (on the left in Figure 1), we have a pool of classifiers, $\mathcal{C} = C_1, ..., C_N$, of size N, and we map data instances into a new space defined by the class probability estimates from each classifier for

a given class label ℓ_1. Since we consider two-class problems, the choice of class label does not change our results. Next, we find each new instance's k-nearest neighbors in the validation set $\mathcal{V}$, of size M, using this space. These k neighbors, denoted by the set $\mathcal{V}'$, can be considered as the data instances that, on average, the classifiers agree are similar to the new instance. This step is referred to as "static" because, for all new instances, we consider the output of all N classifiers in the pool.

In the dynamic step of our framework (on the right in Figure 1), we use the results from the static step to select a subset of classifiers $\mathcal{C}'$, of size $E < N$, from the original pool, $\mathcal{C}$, that are suitable for classifying a given new instance. Reducing the size of the space is important because the number of classifiers in our framework is large and (dis)similarity becomes less meaningful as the feature space dimensionality increases. Our selection method favors classifiers that have high confidence in their predictions for the neighbors identified in the static step (i.e. classifiers for which the predictions are away from the 0.5 decision boundary).[1] Confidence of a classifier C_n on k neighbors is calculated as follows:

$$Conf_{C_n} = \frac{\sum_{i=1}^{k} \Pr(o_{i,n}|i)}{k}$$

where $o_{i,n}$ represents the label assigned to instance i by classifier C_n and $\Pr(o_{i,n}|i)$ represents the probability estimate returned by the classifier for the assigned class label. In other words, the confidence of a classifier on k neighbors is the average of probability estimates returned for the decisions. Most confident E classifiers are chosen to form the reduced space. Reducing the size of the classifier set in this manner avoids the unstable behavior that may occur near the decision boundary of some classifiers.

Once the classifier subset, $\mathcal{C}'$, is generated, the original distance measure is reapplied to find a new set of neighbors $\mathcal{V}^*$ for the new instance. This neighborhood is then used for the final classification of the instance; the new instance is assigned to the class label most common among its k neighbors.

4 Running Time Analysis

To analyze the running time of the system, we should first evaluate the static and dynamic steps separately. The running time of the static step can be analyzed in two parts: (1) forming the probability space; and (2) finding k neighbors of a new instance. Forming the probability space consists of getting predictions for M validation instances from N classifiers which takes $O(M \times N)$. This is a preprocessing step as once this space is formed, it will remain same for all new data instances. Finding k neighbors requires getting prediction for each new instance from N classifiers ($O(N)$), calculating distance between the new instance and validation instances ($O(M \times N)$), and finally finding k instances that are closest to the new instance ($O(k \times M)$). Overall, the running time for this static step is $O(M \times N)$ which is for calculating the distance between a new instance and the validation instance.

The dynamic step is composed of two main parts: (1) reducing the classifier space; and (2) finding new k neighbors of a new instance. Since predictions for validation instances and k neighbors of a new instance are already found in the static step, the running time of reducing the space consists of calculating the confidence of classifiers on the neighbors ($O(k \times N)$) and finding the most confident E classifiers ($O(E \times N)$). The running time of the rest of the dynamic step can be analyzed in a similar manner to the static step ($O(k \times E)$). The overall running time of the whole system is dominated by the running time of the static step as $k < M$ and $E < N$.

5 Baseline Distance Measures

In this section, we consider three different distance measures for our baseline: Euclidean Distance, Template Matching, and Oracle-Based Template Matching. Two of these measures are proposed by Cavalin et al. (2010) in further detail.

5.1 Euclidean Distance

We first consider the Euclidean Distance (ED) between two data instances, i and j, which can be expressed as:

$$ED_{i,j} = \sqrt{\frac{\sum_{d=1}^{D} \left(\phi_{i,j,F_d}\right)^2}{D}}$$
$$\phi_{i,j,F_d} = F_d(i) - F_d(j)$$

where a data set consists of the feature set $\mathcal{F} = \{F_d | d = 1, \ldots, D\}$ and $F_d(i)$ represents the value of feature F_d for data instance i.

5.2 Template Matching

Given a set of classifiers $\mathcal{C} = \{C_n | n = 1, \ldots, N\}$, Template Matching (TM) considers the percentage of classifiers, denoted by $TM_{i,j}$, that agree on the label of test instance i and validation instance j:

$$TM_{i,j} = \frac{\sum_{n=1}^{N} \alpha_{i,j,n}}{N}$$
$$\alpha_{i,j,n} = \begin{cases} 1, & \text{if } o_{i,n} = o_{j,n} \\ 0, & \text{otherwise.} \end{cases}$$

where $o_{j,n}$ represents the label assigned to instance j by classifier C_n and $\alpha_{i,j,n}$ represents whether classifier C_n assigns the same label to instances i and j. It follows that the higher $TM_{i,j}$, the more similar the pair of instances (i, j).

5.3 Oracle-Based Template Matching

Oracle-based Template Matching (OTM) expands upon TM by taking into account the correctness of classifiers on data instances. Specifically, when evaluating the similarity of test instance i to validation instance j

[1] For completeness, other selection methods are considered in Section 7.7.

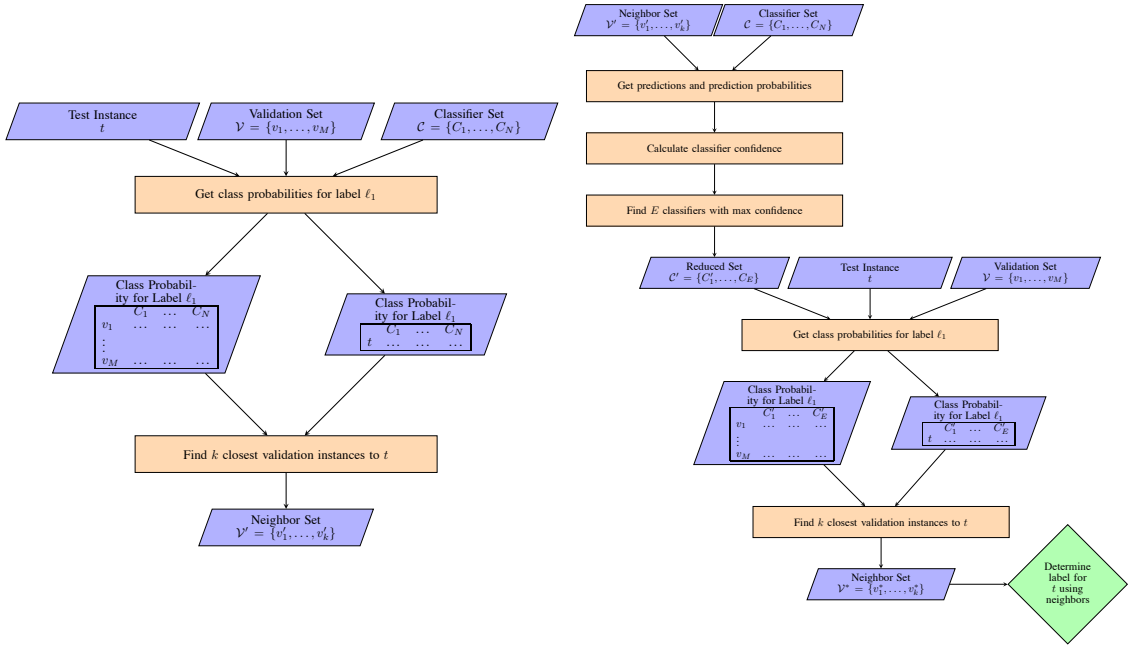


Figure 1: An overview of the proposed framework: Static Step (on the left) and Dynamic Step (on the right)

using OTM, only those classifiers that correctly classify j are considered. This measure is represented by the quantity $OTM_{i,j}$ as follows:

$$OTM_{i,j} = \frac{\sum_{n=1}^{N} \beta_{i,j,C_n}}{\sum_{n=1}^{N} \gamma_{i,j}}$$

$$\beta_{i,j,C_n} = \begin{cases} 1, & \text{if } o_{i,C_n} = o_{j,C_n} \ \& \ o_{j,C_n} = \text{label}_j \\ 0, & \text{otherwise.} \end{cases}$$

$$\gamma_{i,j} = \begin{cases} 1, & \text{if } o_{j,C_n} = \text{label}_j \\ 0, & \text{otherwise.} \end{cases}$$

Similar to TM, a higher value of $OTM_{i,j}$ indicates greater similarity between instances i and j.

6 Proposed Measures

For our dynamic class prediction framework, we propose two new distance measures: Probability-Based Template Matching and Probability-Based Template Matching with Accuracy. These measures are described in detail below.

6.1 Probability-Based Template Matching

Probability-Based Template Matching (PTM) maps each data instance into an alternate feature space constructed by using the probability estimates of each classifier in the pool as the values of the features. As previously mentioned, for the two-class problems considered in this paper, the probability estimates are taken with respect to a particular class label, and the choice of label is arbitrary. The similarity between instances i and j, denoted by $PTM_{i,j}$, is calculated as the Euclidean distance between them in this alternate feature space:

$$PTM_{i,j} = \sqrt{\frac{\sum_{n=1}^{N} \left(\phi_{i,j,C_n}\right)^2}{N}}$$

$$\phi_{i,j,C_n} = P_{C_n,i} - P_{C_n,j}$$

where $P_{C_n,i}$ represents the probability estimate in the alternative feature space for data instance i. Similar to the ED, the pair of instances (i, j) with smallest $PTM_{i,j}$ is considered to be most similar.

6.2 Probability-Based Template Matching with Accuracy

Probability-Based Template Matching with Accuracy (PTMA) integrates the correctness of classifiers on validation instance j. We focus on the probability estimates returned by the classifier for the correct class label of the new and validation instances. This gives us four different cases as specified in the equation. To avoid one case eliminating the effect of the other cases, we re-scale all values to be between $[0, 1]$.

$$PTMA_{i,j} = \sqrt{\frac{\sum_{n=1}^{N} (\phi_{i,j,C_n})^2}{N}}$$

$$\phi_{i,j,C_n} = \begin{cases} 2 \mid P_{C_n,i} - P_{C_n,j} \mid, \\ \quad \text{if } o_{i,C_n} = o_{j,C_n} = \text{label}_j \\ \mid 1 - P_{C_n,i} - P_{C_n,j} \mid, \\ \quad \text{if } o_{i,C_n} = o_{j,C_n} \neq \text{label}_j \\ \mid P_{C_n,i} - P_{C_n,j} \mid, \\ \quad \text{if } o_{i,C_n} \neq o_{j,C_n} = \text{label}_j \\ 2 \mid 1 - P_{C_n,i} - P_{C_n,j} \mid, \\ \quad \text{if } o_{i,C_n} \neq o_{j,C_n} \neq \text{label}_j \end{cases}$$

We illustrate how these measures differ with a simple example. Tables 1 and 2 show, respectively, the assigned labels and the corresponding probability estimates for class label l_1 returned by classifiers $C_1, ..., C_5$.

Table 1: Assigned class labels by classifiers

	C_1	C_2	C_3	C_4	C_5	Correct Label
v_1	l_1	l_1	l_2	l_1	l_1	l_1
v_2	l_1	l_1	l_2	l_1	l_2	l_2
v_3	l_2	l_2	l_1	l_1	l_1	l_1

Table 2: Class probabilities (for l_1) by classifiers

	C_1	C_2	C_3	C_4	C_5
v_1	0.53	0.57	0.44	0.90	0.85
v_2	0.92	0.89	0.24	0.52	0.35
v_3	0.46	0.43	0.53	0.88	0.90

We calculate the relevant measures for data instances v_1, v_2, v_3 below:

$$\begin{aligned} TM_{v_1,v_2} &= 0.80, & TM_{v_1,v_3} &= 0.40 \\ OTM_{v_1,v_2} &= 1.00, & OTM_{v_1,v_3} &= 0.66 \\ PTM_{v_1,v_2} &= 0.83, & PTM_{v_1,v_3} &= 0.19 \\ PTMA_{v_1,v_2} &= 0.94, & PTMA_{v_1,v_3} &= 0.11 \end{aligned}$$

We conclude that v_1 and v_2 are more similar than v_1 and v_3, according to both TM and OTM measures. However, v_1 and v_3 are considered more similar instances according to our PTM and PTMA measures. Therefore, considering only the assigned labels to determine similarity may lead to incorrect decisions. This is especially true for instances that are closer to the decision boundary of a classifier, as the classifier provides less confidence in its prediction for those instances.

7 Experimental Setup and Results

In our experiments, we use 13 data sets with varying numbers of features and data instances retrieved from the LIBSVM website (Chang & Lin (2011)). A summary of the data sets and the classifiers generated for each is presented in Table 3.

The programming code was written in MATLAB and LIBSVM (Chang & Lin (2011)) was used to construct RBF kernel SVM classifiers. The training parameters were chosen such that classifiers overfit the data. Each experiment was repeated 100 times, with each run registering a unique seed value for the random number generator.

Table 3: Summary of the data sets and classifiers

Data Set	#Data Points	#Features	%Good Classifiers
a1a	1605	119	100
australian	690	14	64.17
diabetes	768	8	99.74
german	1000	24	100
splice	1000	60	52.77
heart	269	13	59.44
liver disorder	345	6	79.46
sonar	208	60	54.04
breast cancer	683	10	97.14
ionosphere	351	34	88.75
mushrooms	8124	112	54.05
w1a	2477	300	100
rcv	20242	47236	56.65

For each run, the data sets are randomly divided into three subsets such that 60% of the instances is used to train classifiers, 20% is used for validation, and the remaining 20% is used for testing. A pool of 1000 classifiers is then constructed for each data set using a combination of bootstrap instance sampling (as in bagging) and random subspace selection on the training set.[2] In so doing, we ensure that the initial classifier pool is highly diverse. Finally, classifiers with an error rate above 50% are removed from the pool. The last column of Table 3 represents the percentage of the classifiers with less than 50% error rate in the generated pool of classifiers.

In the following sections, we first analyze the static step. In Section 7.2, we perform an experiment to set the value of k for baseline and proposed (dis)similarity measures. Then, we compare the effectiveness of these measures to find the closest k neighbors to make the predictions in Section 7.2. In Section 7.3, we investigate whether the classifier-based feature space can improve the *KNORA* method results. In addition, we choose the appropriate distance measure for the *KNORA* before comparing it with the static step.

Once our comparison for the static step is done, we turn our attention to the dynamic step. We first determine the optimal reduced space size in Section 7.4. In Section 7.5, we examine the contribution of the dynamic step in our framework in addition to the static step, while we compare the dynamic step against common benchmarks in Section 7.6. We then explore various strategies in evaluating the dynamic step in Section 7.7. Finally, in Section 7.8, we compare the performance of the dynamic step against that of the ensembles formed by the classifiers for the reduced space.

7.1 Evaluation of Size of Neighborhood

The number of neighbors considered for each similarity measure is crucial. As a preprocessing step, we perform an experiment in which k is varied over the odd integers from 1 to 25 to find the optimal value. In this step, for each run of a data set, we predict the validation instances using training instances and decide for which k value maximum accuracy is obtained. Table 4 shows the best k value for each similarity measure, averaged over all of the runs. We find that the value of

[2] Due to its size, only 100 classifiers are generated for the *rcv* data set.

k is smaller for the OTM and PTMA measures. A possible explanation is that these two measures take into account the correctness of the decision associated with the validation instances. Consequently, a small neighborhood of validation instances is sufficient to correctly classify the new instance.

Table 4: Best k value for each similarity measure

Data Set	TM	OTM	ED	PTM	PTMA
a1a	9.6	3.3	14.24	9.8	1.42
australian	9.34	1.5	12.14	11.24	1.46
breast_cancer	9.06	1.66	5.14	7.74	1.08
diabetes	11.14	3.22	13.34	12.3	2.66
german	14.52	3.52	14.08	13.68	1.28
heart	7.48	1.56	13.32	7.06	1.86
ionosphere	9.64	2.16	2.84	9.34	1.12
liver_disorder	10.02	2.2	9.02	12.12	3.82
mushrooms	9.56	1	1	10.48	1
rcv	16.5	4.9	1.94	16	3.62
sonar	5.94	1.12	1.86	8.26	1.22
splice	3.46	1.16	10.02	5.48	1
w1a	1.34	1	2.2	1.34	1

7.2 Evaluation of Different Distance Measures

After the best k value is determined for each of the similarity measures presented in Section 5, we perform experiments to assess the accuracy of these measures in classifying new data instances. These experiments use the k-NN algorithm to define the neighborhood for and classify each new instance based on the validation instances deemed to be similar. Table 5 summarizes the results of these experiments: we report mean and standard deviation of accuracy from 100 runs in parentheses, respectively. Entries highlighted in bold in the table indicate the best accuracy achieved for that data set. From these results, it is clear that, on average, PTMA achieves better performance than the other measures.

We also perform pairwise t-tests (at 5% significance level) to compare the proposed distance measures with the baseline measures, with the results tabulated in Table 6. Entries are specified as (x_1, x_2, x_3) which represents (wins,ties,losses) of the proposed methods against the baseline measures over all 13 data sets. For example, when compared to the TM measure, the PTM measure performs statistically significantly better for 6 data sets and worse for 1 data set. For the rest of the data sets, the differences between two measures is not statistically significant. PTM, TM and ED use the same distance measure but in different spaces. As demonstrated by the pairwise t-test results for the PTM measure against the TM and ED measures, the use of the class probability space improves accuracy over either the class prediction space or the original feature space, even without consideration of the classifiers' accuracies for the validation instances. Based on the results for the PTMA measure against the PTM measure, we can conclude that integration of the classifier accuracy into the distance measure further improves accuracy.

Table 6: Pairwise t-test results for dis/similarity measures

	TM	OTM	ED	PTM
PTM	(6,6,1)	(8,2,3)	(10,3,0)	-
PTMA	(7,3,3)	(5,5,3)	(8,3,2)	(6,3,4)

7.3 Comparisons against *KNORA* method

Our proposed framework in Section 3 in both static and dynamic steps finds k nearest neighbors of a new instance. However, unlike *KNORA* instead of using them to form an ensemble of classifiers it uses neighbors' labels to make predictions. In this section, we perform experiments to compare our static step against *KNORA* methods. In addition, *KNORA* uses Euclidean distance measure to find the neighbors of a new instance and the choice of distance measure had no effect on the performance of *KNORA* (Vriesmann et al. (2012)). However, the measures considered by Vriesmann et al. (2012) are based on finding distance in the original feature space. We extend Vriesmann et al. (2012) to analyze the effectiveness of *KNORA* in classifier-based space and find best distance measure for *KNORA* methods before we perform the comparison against static step of our framework.

7.3.1 Evaluating Similarity Measures for KNORA

KNORA-Eliminate and *KNORA-Union* are implemented using the measures discussed in Sections 5 and 6: TM, OTM, ED, PTM, and PTMA. A pairwise t-test at $\alpha = 0.05$ was performed to compare PTM and PTMA against other measures. Entries in tables 7 and 8 represent the number of wins, ties, and losses, respectively, of the proposed measures against the baseline measures (as well as PTMA against PTM) for *KNORA* methods for 13 datasets considered here. For *KNORA-Eliminate* PTMA is always superior to the other measures. However, PTM seems to be more suitable for smaller k values. PTM performs better than other measures for *KNORA-Union*. Interestingly, PTMA performs considerably worse compared to others. As a result of this experiment, we can conclude that $KNORA$ performs well in a classifier based space especially with our proposed measures.

Table 7: Pairwise t-test: comparison of (dis)similarity measures for KNORA-Eliminate

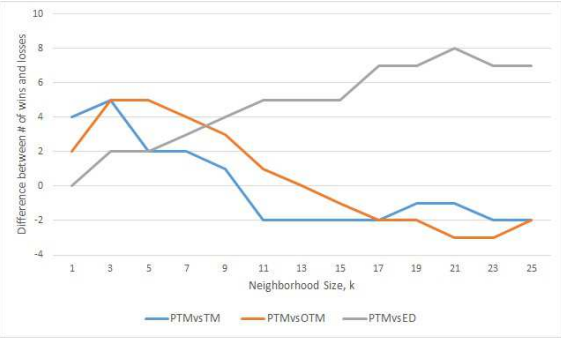
	PTM vs.			*PTMA vs.*			
k	TM	OTM	ED	TM	OTM	ED	PTM
1	(5,7,1)	(6,3,4)	(3,7,3)	(7,1,5)	(5,5,3)	(6,2,5)	(6,3,4)
3	(6,6,1)	(7,4,2)	(5,5,3)	(7,2,4)	(6,4,3)	(8,1,4)	(8,1,4)
5	(5,5,3)	(8,2,3)	(3,9,1)	(7,2,4)	(8,2,3)	(8,2,3)	(7,2,4)
7	(5,5,3)	(7,3,3)	(5,6,2)	(7,3,3)	(7,3,3)	(8,3,2)	(7,3,3)
9	(4,6,3)	(6,4,3)	(6,5,2)	(5,5,3)	(8,2,3)	(9,1,3)	(7,5,1)
11	(4,3,6)	(6,2,5)	(7,4,2)	(5,6,2)	(7,3,3)	(10,1,2)	(9,3,1)
13	(4,3,6)	(5,3,5)	(7,4,2)	(5,5,3)	(6,4,3)	(11,1,1)	(10,2,1)
15	(4,3,6)	(5,2,6)	(7,4,2)	(5,6,2)	(6,3,4)	(11,1,1)	(11,1,1)
17	(4,3,6)	(5,1,7)	(8,4,1)	(6,5,2)	(5,3,5)	(11,1,1)	(11,1,1)
19	(4,4,5)	(5,1,7)	(8,4,1)	(7,5,1)	(5,3,5)	(11,1,1)	(10,2,1)
21	(4,4,5)	(4,2,7)	(9,3,1)	(10,3,0)	(5,3,5)	(11,1,1)	(11,1,1)
23	(2,7,4)	(4,2,7)	(8,4,1)	(10,3,0)	(6,2,5)	(11,1,1)	(11,1,1)
25	(2,7,4)	(4,3,6)	(8,4,1)	(10,3,0)	(7,2,4)	(11,1,1)	(11,2,0)

7.3.2 Static Step vs. *KNORA*

Since the size of an ensemble used to make predictions for each new instance changes for *KNORA*-Eliminate and -Union methods, we analyze the performance of *KNORA* Eliminate and Union against the *static* step of our framework. We explore whether k-NN (Static Step) is better than *KNORA* after neighbors are found by using the proposed measures since the experiments

Table 5: Average accuracy and its standard deviation for dis/similarity measures

Data Set	TM	OTM	ED	PTM	PTMA
a1a	(79.65, 1.34)	(76.18, 10.63)	(78.92, 1.28)	**(81.11, 2.01)**	(77.54, 1.04)
australian	(85.21, 2.84)	(84.2, 5.22)	(85.76, 3.03)	(85.54, 2.93)	**(86.06, 2.70)**
breast_cancer	(96.26, 2.43)	(96.03, 4.64)	(96.13, 1.67)	(96.54, 1.49)	**(96.82, 1.38)**
diabetes	(71.03, 11.74)	(52.8, 20.31)	(72.04, 3.26)	(74.15, 3.87)	**(75.36, 2.50)**
german	(71.35, 1.21)	(71.9, 1.12)	(71.16, 2.07)	**(72.72, 1.79)**	(71.46, 0.92)
heart	(81.96, 5.16)	(82.97, 4.76)	(80.68, 5.36)	(80.83, 5.56)	**(83.08, 5.12)**
ionosphere	(90.93, 12.01)	(82.48, 22.58)	(80.71, 5.93)	(93.74, 2.79)	**(94.04, 2.49)**
liver_disorder	(69.14, 6.15)	**(71.06, 5.71)**	(58.09, 5.71)	(69.14, 5.11)	(70.45, 4.09)
mushrooms	(99.75, 0.09)	(99.67, 0.08)	(98.56, 0.05)	**(100, 0)**	**(100, 0)**
rcv	(86.77, 16.49)	(55.51, 17.08)	(90.54, 0.51)	(96.93, 0.38)	**(97.04, 0.26)**
sonar	(83.16, 5.84)	**(85.83, 5.63)**	(72.15, 7.73)	(82.87, 6.65)	(77.42, 6.54)
splice	(69.05, 7.92)	(66.25, 9.03)	(65.76, 3.82)	**(77.55, 4.84)**	(53.5, 0.96)
w1a	(97.18, 0.27)	(92.91, 0.14)	(97.19, 0.43)	(97.17, 0.41)	(97.13, 0.15)



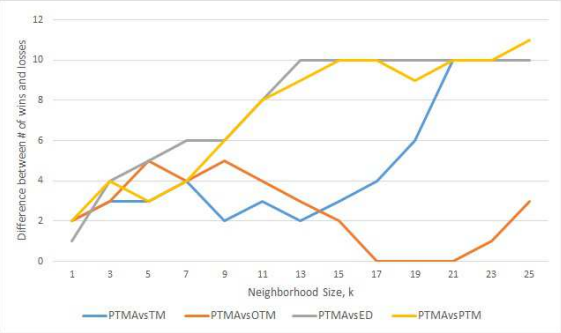


Figure 2: Comparisons of proposed measures against other dis/similarity measures for KNORA Eliminate

in Section 7.3.1 indicate that *KNORA* performs better with PTM and PTMA measures compared to ED, TM, and OTM. Table 9 show the t-test performed to compare *KNORA* and k-NN in classifier-based probability space. Figure 4 illustrates the difference between the number of times *KNORA* outperforms and underperforms against static step of our framework. For instance, for $k = 1$ and distance measure PTM, static step of our framework outperforms *KNORA*-Eliminate for 10 datasets and under-performs for 1 dataset. For 2 datasets, there is a tie. So, for $k = 1$ the Figure 4 shows $1 - 10 = -9$. The results in Table 9 and in Figure 4 show that static step —k-NN with PTM or PTMA— outperforms *KNORA*-Eliminate and -Union with PTM or PTMA.

Table 8: Pairwise t-test: comparison of (dis)similarity measures for KNORA-Union

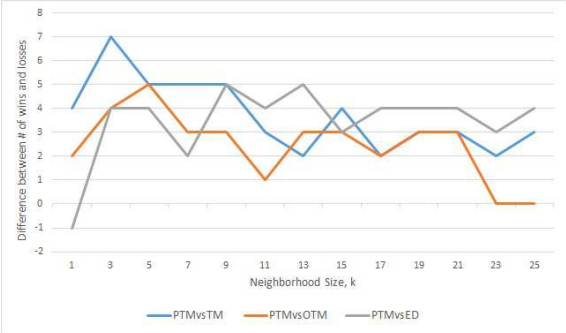
	PTM vs.			*PTMA vs.*			
k	TM	OTM	ED	TM	OTM	ED	PTM
1	(5,7,1)	(6,3,4)	(3,6,4)	(7,1,5)	(5,5,3)	(7,1,5)	(6,3,4)
3	(7,6,0)	(6,5,2)	(5,7,1)	(2,6,5)	(4,5,4)	(2,5,6)	(1,6,6)
5	(5,8,0)	(6,6,1)	(5,7,1)	(2,4,7)	(3,5,5)	(2,6,5)	(2,3,8)
7	(5,8,0)	(4,8,1)	(3,9,1)	(2,4,7)	(2,7,4)	(2,6,5)	(1,5,7)
9	(5,8,0)	(4,8,1)	(6,6,1)	(1,5,7)	(1,8,4)	(2,7,4)	(1,5,7)
11	(3,10,0)	(4,6,3)	(4,9,0)	(2,4,7)	(2,6,5)	(2,5,6)	(1,5,7)
13	(3,9,1)	(5,6,2)	(5,8,0)	(1,5,7)	(1,7,5)	(2,5,6)	(1,5,7)
15	(5,7,1)	(4,8,1)	(4,8,1)	(1,5,7)	(1,7,5)	(2,7,4)	(1,5,7)
17	(3,9,1)	(3,9,1)	(5,7,1)	(1,6,6)	(1,7,5)	(2,8,3)	(1,5,7)
19	(3,10,0)	(4,8,1)	(5,7,1)	(1,7,5)	(1,8,4)	(2,8,3)	(1,6,6)
21	(4,8,1)	(4,8,1)	(5,7,1)	(1,8,4)	(1,8,4)	(2,5,6)	(1,6,6)
23	(3,9,1)	(2,9,2)	(5,6,2)	(1,7,5)	(1,7,5)	(2,5,6)	(1,7,5)
25	(3,10,0)	(1,11,1)	(5,7,1)	(1,9,3)	(1,9,3)	(2,7,4)	(1,7,5)

Table 9: Pairwise t-test: KNORA methods vs. static step

	KNORA-E vs.		*KNORA-U vs.*	
k	PTM-S	PTMA-S	PTM-S	PTMA-S
1	(1,2,10)	(3,3,7)	(1,2,10)	(3,3,7)
3	(1,2,10)	(3,3,7)	(5,4,4)	(3,4,6)
5	(1,1,11)	(3,1,9)	(5,4,4)	(3,3,7)
7	(0,2,11)	(3,2,8)	(5,4,4)	(3,4,6)
9	(0,2,11)	(3,1,9)	(5,3,5)	(3,4,6)
11	(0,1,12)	(3,1,9)	(4,4,5)	(3,5,5)
13	(0,1,12)	(3,1,9)	(5,3,5)	(3,3,7)
15	(0,1,12)	(2,2,9)	(4,4,5)	(3,4,6)
17	(0,1,12)	(2,1,10)	(4,2,7)	(3,4,6)
19	(0,1,12)	(2,1,10)	(4,2,7)	(3,5,5)
21	(0,1,12)	(2,1,10)	(4,2,7)	(3,3,7)
23	(0,1,12)	(2,1,10)	(4,2,7)	(3,2,8)
25	(0,1,12)	(2,1,10)	(4,3,6)	(3,3,7)

7.4 Evaluating Reduced Space Size in Dynamic Step

It was shown that the marginal improvement in the performance of an ensemble diminishes for sizes beyond 25 (Breiman (1996)). In this section, we validate and refine this assumption by investigating the effects of the reduced space size on the performance of our framework. Experiments are run with for reduced space sizes, E, in increments of 5 over the range of 10 to 50, and the results are tabulated in Table 10. Overall, as shown in the table, the choice of the reduce space size has minimal effect on the performance of the framework. Therefore, for the experiments in Sections 7.5–7.7, the size of the reduced space is kept at 25.



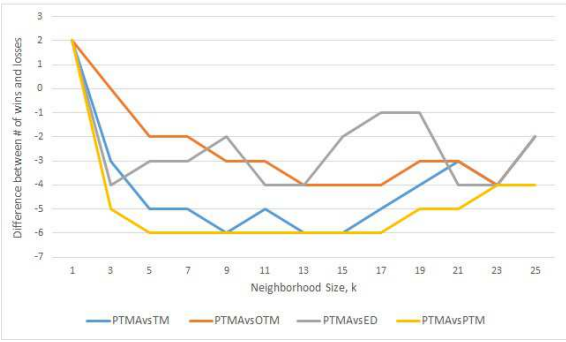


Figure 3: Comparisons of proposed measures against other dis/similarity measures for KNORA Union

Table 10: Pairwise t-test: dynamic vs. static steps across reduced space size (E)

	PTM-D vs.		*PTMA-D vs.*	
E	PTM-S	PTMA-S	PTM-S	PTMA-S
10	(0,7,6)	(3,3,7)	(5,4,4)	(4,6,3)
15	(0,8,5)	(3,3,7)	(5,4,4)	(5,7,1)
20	(0,9,4)	(3,4,6)	(6,3,4)	(5,8,0)
25	(1,8,4)	(3,4,6)	(6,3,4)	(5,8,0)
30	(1,8,4)	(3,3,7)	(7,2,4)	(5,8,0)
35	(2,7,4)	(4,3,6)	(6,3,4)	(5,8,0)
40	(0,9,4)	(4,4,5)	(7,2,4)	(5,8,0)
45	(0,10,3)	(4,3,6)	(6,3,4)	(5,8,0)
50	(1,9,3)	(4,5,4)	(5,4,4)	(4,9,0)

7.5 Comparison of Dynamic and Static Steps

This experiment is performed to evaluate the gain achieved by having the dynamic step as part of our framework. Even though the running time of our system is dominated by the static step, for the dynamic step the neighborhood of a new instance is refined using the information retrieved from the static step to form the "Reduced Space" and a new set of neighbors is found. These calculations require some additional time cost to the system.

Tables 11 and 12 compare the results from the static and dynamic steps for both the PTM and PTMA distance measures. In Table 11, we report mean and standard deviation of accuracy from 100 runs in parentheses, respectively. As observed in these two tables, the dynamic step for the PTM and PTMA measures (hereafter referred to as PTM-D and PTMA-D, respectively) consistently outperforms the static step for the ED measure, even for data sets where it underperforms the static step for the PTM and PTMA measures

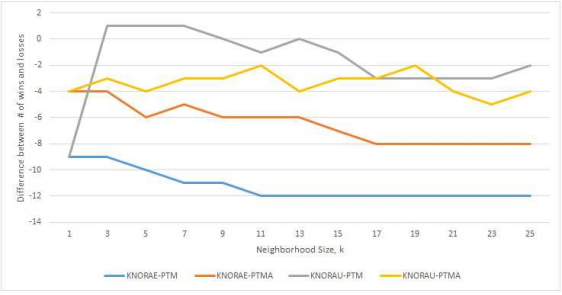


Figure 4: Comparison of Static Step with KNORA methods

(termed PTM-S and PTMA-S). Further, an important result is that PTMA-D performs at least as well as PTM-S and PTMA-S. However, PTM-D performs significantly worse than PTM-S and PTMA-S. When the classifiers for the reduced space are chosen, it is aimed to find expert classifiers on the neighbor instances and the results from this experiment clearly shows that expert classifiers can be identified by considering not just the confidence but also the accuracy of the classifiers on the neighbors which explains the change in the performance of PTMA-D and PTM-D.

Table 12: Pairwise t-test: dynamic vs. static steps

	ED	PTM-S	PTMA-S
PTM-D	(8,4,1)	(1,8,4)	(3,4,6)
PTMA-D	(10,2,1)	(6,3,4)	(5,8,0)

7.6 Dynamic Step vs. Common Benchmarks

We also compared the performance obtained with our framework against the common benchmarks. The baseline benchmarks considered here are:

- Use of the best classifier from set $\mathcal{C}$;
- Use of the 25 best-performing classifiers from set $\mathcal{C}$ with majority voting
- Use of a single SVM classifier trained on all of the training data.

In addition, we used "Modified Best Improvement" algorithm to find the best ensemble of size E over the validation set. Since it can be impractical to evaluate all $\binom{N}{E}$ possible ensembles, as required for the traditional "Best Improvement" algorithm, a modified form of the algorithm is employed: given one of the E classifier slots in the ensemble, each of the $N - E$ unused classifiers is, in turn, substituted into that slot, and the accuracy of the ensemble is re-evaluated. The classifier that provides the best ensemble performance is permanently assigned to the slot, and the displaced classifier is returned to the pool of unused classifiers. This process is then repeated for the next classifier slot in the ensemble until all slots have been processed. We perform an experiment in which E is set to 25 to find the best ensemble on the validation set for a given data set. Tables 13 and 14 compare the performance of our proposed measures against these benchmarks in terms

Table 11: Average accuracy and its standard deviation in the static and dynamic steps

Data Set	ED	PTM-S	PTMA-S	PTM-D	PTMA-D
a1a	(78.92, 1.28)	**(81.11, 2.01)**	(77.54, 1.04)	(79.37, 2.08)	(79.69, 1.63)
australian	(85.76, 3.03)	(85.54, 2.93)	**(86.06, 2.7)**	(85.43, 2.79)	(85.79, 2.81)
breast_cancer	(96.13, 1.67)	(96.54, 1.49)	**(96.82, 1.38)**	(96.63, 1.49)	**(96.82, 1.37)**
diabetes	(72.04, 3.26)	(74.15, 3.87)	(75.36, 2.5)	(74.39, 3.57)	**(75.51, 2.58)**
german	(71.16, 2.07)	**(72.72, 1.79)**	(71.46, 0.92)	(71.65, 1.87)	(71.89, 1.26)
heart	(80.68, 5.36)	(80.83, 5.56)	**(83.08, 5.12)**	(80.33, 5.45)	(82.95, 4.92)
ionosphere	(80.71, 5.93)	(93.74, 2.79)	(94.04, 2.49)	**(94.13, 2.64)**	**(94.13, 2.47)**
liver_disorder	(58.09, 5.71)	(69.14, 5.11)	**(70.45, 4.09)**	(68.62, 5.71)	**(70.49, 4.52)**
mushrooms	(98.56, 0.05)	**(100, 0)**	**(100,0)**	**(100, 0)**	**(100, 0)**
rcv	(90.54, 0.51)	(96.93, 0.38)	**(97.04, 0.26)**	(96.93, 0.37)	**(97.03, 0.26)**
sonar	(72.15, 7.7)	**(82.87, 6.65)**	(77.42, 6.5)	(82.68, 5.82)	(80.77, 6.15)
splice	(65.76, 3.82)	**(77.55, 4.84)**	(53.5, 0.96)	(75.85, 4.06)	(55.22, 1.2)
w1a	(97.19, 0.43)	(97.17, 0.41)	(97.13, 0.15)	(96.79, 0.58)	**(97.19, 0.24)**

of average accuracy and pairwise t-test results, respectively. From these tables, we conclude that PTMA outperforms all of the benchmarks. Even PTM, a basic distance measure in the probability space, performs better than any of these benchmarks.

Table 14: Pairwise t-test: dynamic step vs. benchmark methods

	Best Classifier	Best 25	Single SVM	Best Ens
PTM-D	(12,0,1)	(7,4,2)	(8,2,3)	(4,7,2)
PTMA-D	(12,1,0)	(9,3,1)	(8,3,2)	(9,3,1)

7.7 Evaluating Different Strategies for Dynamic Step

As described in the previous section, our proposed framework takes into account the class prediction probabilities assigned to the k-nearest neighbors found in the static step to choose the classifiers whose outputs will be used for generating the reduced probability space for the dynamic step. In this section, we investigate different strategies for choosing a subset of classifiers from the original pool to be used in the dynamic step. In addition to the selection strategy presented earlier, five alternate strategies are also implemented:

- **Local Accuracy (LA):** Classifiers which correctly classify at least one of the k neighbors are added to $\mathcal{C}'$. If $|\mathcal{C}'| > 25$, then we consider only the 25 top-performing classifiers.
- **LA $+\mathbf{Acc_{Val}}$:** This strategy is similar to LA; however, if $|\mathcal{C}'| < 25$, additional classifiers are selected to reach a size of 25. These classifiers are chosen from the best-performing classifiers on the validation instances that are not already in $\mathcal{C}'$.
- **Conditional LA (C + LA):** For this strategy, we only consider the classifiers with $\geq 50\%$ accuracy in the neighborhood. If there is no such classifier found, then the label for a new instance is assigned randomly. If there are more than 25 classifiers, then we consider more accurate 25 classifiers in the region.
- **LA + Closeness (LA + Cl):** If the total number of classifiers which correctly classify at least one of the k neighbors is less than 25, we also add the classifiers which are close to making the correct decision on the neighbors. Closeness is defined as the absolute difference between the probability estimate returned by the classifier for the correct class label and the 0.5 decision boundary.
- **Minimum Distance (MinDist):** This strategy chooses classifiers that minimize the average distance between a test instance and its neighbors.

Table 15: Pairwise t-test: dynamic vs. static steps under different strategies

	PTM-D vs.		*PTMA-D vs.*	
Version	PTM-S	PTMA-S	PTM-S	PTMA-S
LA	(1,7,5)	(4,4,5)	(4,5,4)	(2,6,5)
LA+Acc_{Val}	(0,6,7)	(4,2,7)	(2,6,5)	(1,7,5)
C+LA	(0,6,7)	(4,2,7)	(3,6,4)	(1,7,5)
LA+Cl	(0,6,7)	(4,2,7)	(2,6,5)	(1,7,5)
MinDist	(0,6,7)	(4,2,7)	(3,5,5)	(1,7,5)

Table 15 lists the pairwise t-test results for each of these classifier selection strategies against the static step of our framework. As observed in the table, none of these strategies yield improved accuracy over the static step. In contrast, the selection strategy discussed in Section 7.5 does improve performance over the static step when the PTMA measure is used. Intuitively, this can be understood because of the complementary natures of the classifier selection method, which favors *high* levels of confidence, and the similarity measure, which favors *similar* levels of confidence. Further, while the PTM and PTMA measures achieve improved performance by considering instance classification as a continuous range of probabilities rather than as discrete labels–thereby being able to recognize the similarity of two instances close to but on opposite sides of the decision boundary–the area near the decision boundary still reflects uncertainty regarding the nature of the new instance. Therefore, especially refining the static step with PTMA measure to favor high classification confidence on the neighbors while still considering the full range of class label probabilities, as is done in the dynamic step, further improves the accuracy of our framework. However, none of the strategies proposed in this section removes the uncertainty associated with the decision boundary, and therefore they do not offer improved performance.

7.8 Dynamic Step vs. Classifiers in the Reduced Space

The reduced space technique of our framework can be considered a dynamic ensemble selection method. In-

Table 13: Average accuracy and its standard deviation for benchmark methods and dynamic step

Data Set	Best Classifier	Best 25	Single SVM	Best Ens	PTM-D	PTMA-D
a1a	(75.50, 2.33)	(78.06, 1.05)	(75.40, 0.78)	(77.11, 1.10)	(79.37, 2.08)	**(79.69, 1.63)**
australian	(75.93, 6.64)	(84.95, 3.03)	(82.07, 2.94)	(85.33, 2.87)	(85.43, 2.79)	**(85.79, 2.81)**
breast_cancer	(94.80, 2.19)	(96.45, 1.40)	(95.80, 1.40)	(96.50, 1.47)	(96.63, 1.49)	**(96.82, 1.37)**
diabetes	(65.94, 2.05)	(65.12, 0.38)	(73.95, 2.66)	(75.01, 2.90)	(74.39, 3.57)	**(75.51, 2.58)**
german	(69.89, 0.6)	(70.01, 0.01)	(70.37, 1.22)	(71.31, 1.15)	(71.65, 1.87)	**(71.89, 1.26)**
heart	(72.96, 6.76)	(80.85, 5.30)	(73.28, 5.71)	(80.89, 5.21)	(80.33, 5.45)	**(82.95, 4.92)**
ionosphere	(91.80, 3.63)	(93.27, 3.03)	(92.19, 3.09)	(93.21, 2.84)	**(94.13, 2.64)**	**(94.13, 2.47)**
liver_disorder	(63.03, 6.25)	(64.22, 3.86)	**(71.13, 4.71)**	(68.57, 4.28)	(68.62, 5.71)	(70.49, 4.52)
mushrooms	(99.76, 0.62)	(99.97, 0.51)	(99.99, 0.01)	(99.99,0.01)	**(100, 0.0)**	**(100, 0.0)**
rcv	(95.17, 1.61)	(97.03, 0.26)	**(97.17, 0.25)**	(97.05, 0.24)	(96.93, 0.37)	(97.03, 0.26)
sonar	(72.28, 7.73)	(79.54, 6.81)	(73.85, 6.26)	(80.71, 6.50)	**(82.68, 5.82)**	(80.77, 6.15)
splice	(55.37, 3.61)	(56.68, 1.44)	(56.185, 1.22)	(56.83, 1.98)	**(75.85, 4.06)**	(55.22, 1.22)
w1a	(97.10, 0.18)	(97.11, 0.12)	(97.16, 0.25)	(97.13, 0.16)	(96.79, 0.58)	**(97.19, 0.24)**

stead of performing k-NN in this reduced space, those classifiers can be used to form an ensemble to make predictions on the new instance. We performed this experiment with several ensemble and neighborhood size values, (E and k) respectively. As mentioned in Section 7.7, we employed different strategies to choose the classifiers to form the reduced space (i.e. the *dynamic* ensemble). Figures 5 - 8 are plotted based on the difference between the number of wins and losses of dynamic step of our framework against dynamic ensemble selection strategy for a given (k, E) pair.

7.8.1 Dynamic Step vs. *Max-Confidence* Ensemble

Figures 5 and 6 summarize the comparison results between dynamic framework (with PTM/PTMA) against classifiers chosen to form the reduced space per their confidence on the k neighbors chosen in the static step. Figure 5 show that for PTM measure if $k > 7$ and $E > 11$ our dynamic framework performs at least as well as using the E classifiers that form the reduced space. According to Figure 6, Dynamic Framework is still better than the ensemble built by using the classifiers that are chosen to form the reduced space based on their high confidence on the neighbors of a test instance. Dynamic framework with PTMA does not dominate the ensemble as strongly as PTM does but it almost always performs at least as well as the ensemble.

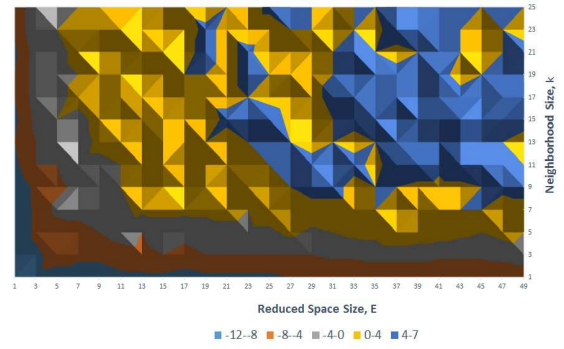


Figure 5: Comparison of Dynamic Step (PTM-D) against the ensemble formed using the classifiers in the reduced space per MaxConf strategy

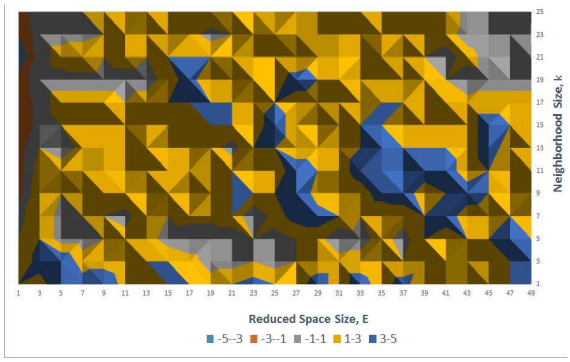


Figure 6: Comparison of Dynamic Step (PTMA-D) against the ensemble formed using the classifiers in the reduced space per MaxConf strategy

7.8.2 Dynamic Step vs Local Accuracy Based Selection Methods

The local accuracy based classifier selection methods that are used for reducing the space in the dynamic step of our framework performed similarly. We present results for the LA method in Figures 7 and 8. PTM-D performs worse or similar to dynamic ensemble generated based on local accuracy. However, PTMA-D performs particularly better for smaller k values. Change in ensemble size seems to have no effect in the performance of PTMA-D.

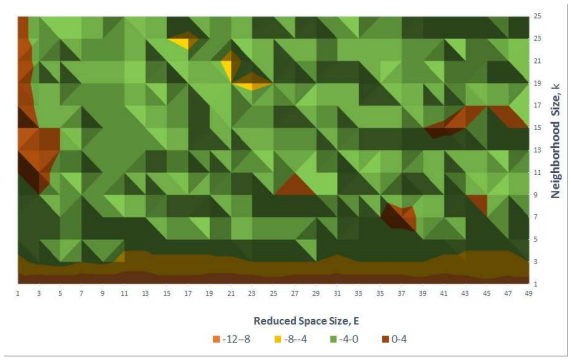


Figure 7: Comparison of Dynamic Step (PTM-D) against the ensemble formed using the classifiers in the reduced space per LA strategy

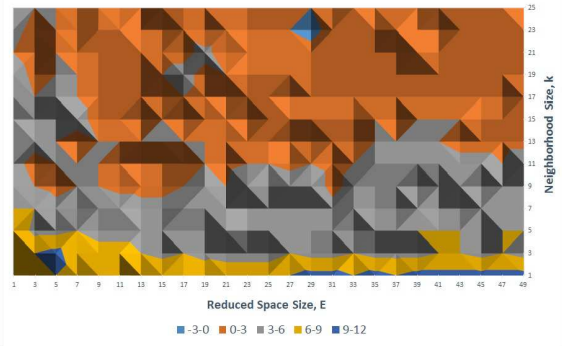


Figure 8: Comparison of Dynamic Step (PTMA-D) with the ensemble formed using the classifiers in the reduced space per LA strategy

8 Conclusion and Future Work

This work proposes a framework for dynamic class prediction using two distance measures (PTM and PTMA) defined based on the probability estimates returned for a particular class label by classifiers for data instances. These measures are compared to three baseline (dis)similarity measures. From our experiments, we conclude that the two proposed measures outperform the baseline measures. Additionally, our experiments reveal that using classifiers' outputs is better than using original feature space to find similar instances. We then compare our results from the static and dynamic steps to some of the traditional classifier and ensemble selection methods. We conclude that k-NN in the probability-estimate-based classifier space outperforms the best classifier, the top performing 25 classifiers, an SVM classifier trained using the entire training data set, and the best ensemble (of size 25) on the validation set.

We considered stationary data for the experiments. This framework can easily be generalized to streaming data as new coming data can be buffered. The size of the pool of classifiers can be controlled by replacing the classifier that performs worst on the buffered instances with a classifier trained on these data instances. When a new classifier is trained, validation set can also be updated. The closest validation instances to the buffered instances can be replaced by these new instances if they also have the same labels. Otherwise, the new instances add new information to the system and should be added to the system without removing any other data instances. As a future work, we would like to modify our framework to account for data streams.

In Section 6, PTM and PTMA measures are defined for two class problems. We would also like to consider multi-class problems as an extension. Multi-class problems present several challenges. Most importantly, unlike two-class problems, there is no clear decision boundary in the probability space. Nonetheless, we believe that the measures presented in this paper could easily be modified to investigate multi-class problems. For example, a simple way to overcome this obstacle would be to transform these problems into a set of two-class problems and training base classifiers accordingly. Alternatively, we could instead consider the probability for each class label of each classifier as an orthogonal dimension of the probability space and calculate the Euclidean distance in this space. However, this approach increases the size of the space by a factor equal to the number of class labels, which makes the distance between two instances less meaningful.

References

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

Cavalin, P., Sabourin, R. & Suen, C. (2010), Dynamic selection of ensembles of classifiers using contextual information, *in* N. Gayar, J. Kittler & F. Roli, eds, 'Multiple Classifier Systems', Vol. 5997 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 145–154.

Chang, C.-C. & Lin, C.-J. (2011), 'LIBSVM: A library for support vector machines', *ACM Transactions on Intelligent Systems and Technology (TIST)* **2**(3), 27.

Didaci, L. & Giacinto, G. (2004), Dynamic classifier selection by adaptive K-nearest-neighbourhood rule, *in* 'Multiple Classifier Systems', Springer, pp. 174–183.

Didaci, L., Giacinto, G., Roli, F. & Marcialis, G. L. (2005), 'A study on the performances of dynamic classifier selection based on local accuracy estimation', *Pattern Recognition* **38**(11), 2188–2191.

Dos Santos, E. M., Sabourin, R. & Maupin, P. (2008), 'A dynamic overproduce-and-choose strategy for the selection of classifier ensembles', *Pattern Recognition* **41**(10), 2993–3009.

Freund, Y. & Schapire, R. E. (1996), Experiments with a new boosting algorithm, *in* 'International Workshop on Machine Learning', Vol. 96, Morgan Kaufmann, pp. 148–156.

Giacinto, G. & Roli, F. (2001), 'Dynamic classifier selection based on multiple classifier behaviour', *Pattern Recognition* **34**(9), 1879–1882.

Ho, T. K. (1998), 'The random subspace method for constructing decision forests', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(8), 832–844.

Ko, A. H., Sabourin, R. & Britto Jr, A. S. (2008), 'From dynamic classifier selection to dynamic ensemble selection', *Pattern Recognition* **41**(5), 1718–1731.

Menahem, E., Rokach, L. & Elovici, Y. (2009), 'Troika–an improved stacking schema for classification tasks', *Information Sciences* **179**(24), 4097–4122.

Ting, K. M. & Witten, I. H. (1997), Stacking bagged and dagged models, *in* 'ICML', pp. 367–375.

Tumer, K. & Ghosh, J. (1996), 'Error correlation and error reduction in ensemble classifiers', *Connection Science* **8**, 385–404.

Vriesmann, L. M., Jr, A. D. S. B., Oliveira, L. E. S. D., Sabourin, R. & Ko, A. H.-R. (2012), 'Improving a dynamic ensemble selection method based on oracle information', *International Journal of Innovative Computing and Applications* **4**(3), 184–200.

Wolpert, D. H. (1992), 'Stacked generalization', *Neural networks* **5**(2), 241–259.

Woods, K., Kegelmeyer Jr, W. P. & Bowyer, K. (1997), 'Combination of multiple classifiers using local accuracy estimates', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**(4), 405–410.

Zhang, Y., Burer, S. & Street, W. N. (2006), 'Ensemble pruning via semi-definite programming', *The Journal of Machine Learning Research* **7**, 1315–1338.

Improving Scalability and Performance of Random Forest Based Learning-to-Rank Algorithms by Aggressive Subsampling

Muhammad Ibrahim **Mark Carman**

Faculty of Information Technology
Monash University, Australia,
Wellington Road, Clayton VIC 3800.
Email: muhammad.ibrahim@monash.edu, mark.carman@monash.edu

Abstract

Random forest based Learning-to-rank (LtR) algorithms exhibit competitive performance to other state-of-the-art algorithms. Traditionally, each tree of the forest is learnt from a bootstrapped copy (sampled with replacement) of the training set, where approximately 63% examples are unique, although some studies show that sampling without replacement also works well. The goal of using a bootstrapped copy instead of the original training set is to reduce correlation among individual trees, thereby making the prediction of the ensemble more accurate. In this study, we investigate whether we can decrease the correlation of the trees even more without compromising accuracy. Among several potential options, we work with the sub-sample used for learning individual trees. We investigate the performance of a random forest based LtR algorithm as we reduce the size of the sub-samples used for learning individual trees. Experiments on Letor data sets reveal that substantial reduction of training time can be achieved using only small amount of data training data. Not only that, the accuracy is likely to increase while maintaining the same level of performance stability as the baseline. Thus in addition to the existing benefit of being completely parallelizable, this study empirically discovers yet another ingredient of random forest based LtR algorithms for making them one of the top contenders for large scale LtR.

Keywords: Random forest, Learning-to-rank, Scalability, Sub-sampling, Correlation, Bootstrapping.

1 Introduction

When a user submits a query, the task of an information retrieval system is to return a list of documents ordered by the predicted relevance to that query. Traditionally different scoring methods, ranging from simple heuristic models (eg. tf-idf score) to probabilistic models (eg. BM25, Language Models), have been used for this task[1]. Recently researchers have investigated supervised machine learning techniques for solving this problem. In this setting a training example is a query-document pair, the corresponding is the relevance judgement for the document with respect to the query is considered to be the ground truth label, and the features are measurements of various base rankers (eg. tf-idf score) – this is then called the learning-to-rank (LtR) problem. Many supervised learning methods have been used so far with empirical success over conventional scoring functions (Li 2011), (Liu 2011).

Since it has been proven that the ranking error is bounded by both the classification error (Li et al. 2007) and regression error (Cossock & Zhang 2006), people often address the LtR problem using classification or regression framework. In these settings a classification or regression algorithm learns to predict the relevance label (which can also be thought as relevance score) of an individual query-document pair, and then during evaluation time, the documents associated to a query are ranked in decreasing order of these scores. This approach is called *pointwise* in the literature (Li 2011) because it treats the instances (i.e. feature vectors corresponding to query-document pairs) independently from one another even if two documents are associated with the same query. The benefits of this approach include lower computational and resource requirements as well as simplicity.

1.1 Random Forest

A random forest (Breiman 2001) is a simple, effective, non-parametric learning algorithm which aggregates the outputs of a large number of independent and variant base learners (decision trees in the default setting). Its major benefits over other state-of-the-art methods include inherent parallelizability, ease of tuning and competitive performance. These benefits attract researchers of various disciplines such as Bioinformatics (Qi 2012), Computer Vision (Criminisi & Shotton 2013), and Data Analysis (Ham et al. 2005), where random forest is a very popular choice. For the LtR task, however, its use has not been studied thoroughly. A few LtR algorithms use random forest in a pointwise manner with regression settings such as (Geurts & Louppe 2011) and (Mohan 2010). The authors show that these algorithms, in spite of their relative simplicity as compared to other state-of-the-art techniques, show competitive performance.

1.2 Bootstrapping in Random Forest

Bootstrapping is a well known effective technique in statistical learning, and is an inherent component of random forest. The method builds a *bootstrapped* sample by randomly choosing an instance *with* replacement from the original training set of size the same as the original sample. The benefits of bootstrapping include: (i) sound mathematical properties (Hastie et al. 2009, Ch. 8), (ii) for ensemble learning, proper use of it reduces variance and possibly also bias (Friedman & Hall 2007), (iii) for ensemble learning, during training phase, the out-of-bag data can be used to estimate test error (Breiman 2001).

In the existing random forest based pointwise algorithms, simple bootstrapping is used where sampling is performed without any concept of query-document structure of the data. That is, sampling is performed not on

[1] (Manning et al. 2008) is a useful work to know about these models.

a per query basis, but rather on a par example (query-document pair) basis. We believe that this may not be the most appropriate approach because it results in fewer documents (and thus less information) per query in the sample. So bootstrapping in our methods ensures that once a query is chosen for inclusion in a training sample (also known as *bag*), all of its associated documents (examples) are chosen, i.e., we sample the training data on a per query-basis.

The usual practice among the researchers is to perform bootstrapping with replacement (Breiman 2001), but (Friedman 2002), (Friedman & Hall 2007) show that bootstrapping without replacement also works for supervised learning, and moreover, (Ganjisaffar et al. 2011) show that it works well for the LtR task. Bootstrapping without replacement has an additional potential benefit from the perspective of scalability because it reduces the learning time.

Suppose Q is the set of queries pertaining to a training set $\mathcal{T}$, and a query q has n_q number of associated documents where its ith document is d_i^q. The bootstrapping (on a per query basis) procedure is given in Algorithm 1.

Data: Training set $\mathcal{T}$ containing $|Q|$ queries (and associated documents).
Result: Bootstrap/Sub-Sample $\mathcal{T}_{sample}$
$\mathcal{T}_{sample} \leftarrow \emptyset$;
while *size*($\mathcal{T}_{sample}$) $<$ *threshold* **do**
  $q \leftarrow chooseRandomQuery(Q)$;
  $\mathcal{T}_{sample}.add(\{d_i^q\}_{i=1}^{n_q})$;
  if *sampleWithoutReplacement* **then**
    $Q = Q \setminus q$;
  end
end

Algorithm 1: Bootstrap Sampling / Sub-sampling on a Per Query Basis. The *threshold* is the required size of the sample.

2 Motivation

In this paper, our goal is to increase scalability while maintaining or improving the level of accuracy.

A random forest has two components which affect its error rate (Breiman 2001): (1) the correlation between the trees: the lower, the better, and (2) the strength of individual trees: the higher, the better. Usually there is a trade-off between the two.

If each tree learns from the entire training set, then there will be high correlation between them, and so their outputs will tend to be comparatively similar, thereby reducing the advantage of aggregation of those outputs. The goal of bootstrapping in random forest is to decrease correlation (i.e., increase variability) between individual trees so that the ensemble is less sensitive to peculiarities of (or perturbations in) the training set. This is achieved by learning individual trees using perturbed training data. At the same time, each tree should be strong enough to have good prediction on unseen data, so intuitively, using a very small training sample for each tree is not likely to work. This raises the question: is there any specific benefit of using 63% of the (distinct) training data examples (as is the case for bootstrapping) in the sample? That is, can we decrease correlation by sub-sampling less than 63% of the data? Put differently, what is the optimal sub-sample size in the context of LtR? To the best of our knowledge, no study has tried to address this question before. If we find that we can use much smaller training set than traditional bootstrapping while maintaining the same level of accuracy, then it will help in reducing training time which is an important factor in large scale LtR as mentioned in (Li 2011, Ch. 7), (Liu 2011, Ch. 20), (Chapelle et al. 2011). While doing so, if we find better performance (because of making individual trees even less correlated), that would add to this benefit. This motivates us to work here.

Another fact increases our motivation which is as follows. Random forests are known to perform well compared to other algorithms with small datasets ((Qi 2012), for example.). For example, some disciplines such as Bioinformtics must work with comparatively smaller samples (Kosorok et al. 2007). Still very good performance is achieved for these domains using random forest based algorithms (Qi 2012). LtR training sets are generally larger than some other disciplines and some other tasks[2]. Hence we investigate the use of smaller sub-samples – smaller than bootstrapped sample – for learning each tree.

The main contribution of this paper is to demonstrate that much smaller sub-samples (for some datasets, as little as 1.6% of the original training set) not only works as well, but indeed increases the accuracy while maintaining similar level of stability of the model in terms of variance across multiple runs.

The rest of the paper is organized as follows. Section 3 discusses some related works. Sections 4 and 5 describe our approach. Section 6 analyzes the experimental results. Section 7 summarizes some important insights from this study. Section 8 concludes the paper.

3 Related Work

Our work is mainly related to two domains. They are described below.

On reducing the correlation among trees. Some options to reduce correlation have been suggested in some works such as the following. (Robnik-Šikonja 2004) use five different gain functions instead of the Gini index[3] to decrease correlation among the trees for classification and regression tasks. With each of the five functions, their method learns one-fifth of the trees of an ensemble, and yields minor improvement. (Geurts et al. 2006) inject more randomness in a standard random forest in selection of the best *split-point*[4]. They consider only one randomly chosen value for an attribute, but the entire training set is used to learn each tree thereby mitigating the possible benefit of scalability. Later (Geurts & Louppe 2011) apply this algorithm to LtR problem.

Another supervised learning framework which can use bootstrapping is gradient boosting (Friedman 2001). (Ganjisaffar et al. 2011) show that bootstrapping without replacement works well in their gradient boosting based LtR algorithm. As for other supervised learning tasks such as classification and regression, (Friedman 2002) shows in the context of regression that the individual trees of a gradient boosted ensemble can learn from a bootstrapped sample (without replacement) instead of the usual practice of using the original sample. (Friedman & Hall 2007) demonstrate both theoretically and empirically that sampling around 50% of the original training set is approximately equivalent to bagging (in terms of bias/variance considerations) for regression problem.

On reducing the training set size. Another line of research which is somewhat related to the topic of our paper is on preparing a comparatively smaller, labelled training sample from a larger unlabelled corpus. (Aslam et al. 2009) investigate different methods for sampling from a large corpus. That is, they studied techniques for generating a good training set from a large document collection. Some other works such as (Macdonald et al. 2012),

[2]For example, as compared to most of the data sets in well-known UCI Repository for Classification (`https://archive.ics.uci.edu/ml/datasets.html`).

[3]Gini index is typically used in a random forest, please see (Breiman 2001) for details.

[4]A split-point is a pair of an attribute and a value for that attribute.

(Long et al. 2010), etc. are concerned with the size of the training set. Some other works such as (Donmez & Carbonell 2008), (Yu 2005) etc. also try to find the examples which, if added to a training set, increase the quality of the learned ranking function. All these works try to improve the quality of the training set while limiting its size so that only *informative* examples are included. Although this line of work is not directly related to our topic, findings from our experiments are applicable *after* these methods are applied.

Thus to the best of our knowledge, there is no study on the optimal size of sub-sample to be used for leaning each tree of a random forest based LtR algorithm. This study tries to address this question.

4 Approach

In our approach, each tree of an ensemble is learnt from a training set which contains q randomly chosen queries (and associated documents) from the original training set. We start with $q = 1$, then gradually increase it (as explained in Experiments section) until the sub-sample contains all the queries of the original training set (i.e., with no bootstrapping at all). In order to get reliable values of evaluation metrics, we run 10 independent runs of each experiment, and report the average value of the metrics.

Although random forest is known to perform well with smaller training sets, if q is too small, then there may be a concern about the stability of the model. Hence for the experiments we, besides measuring average performance, analyze the variance of the metrics across 10 runs. We note however that the variance (instability) is not a major concern for random forest because the larger the number of trees an ensemble has, the lower the variance will be (Breiman 2001). In spite of this, in some cases the lack of sufficient computational resources may limit maximum allowable size of an ensemble, making it important to also analyze the variance.

Another approach would be to make use of all queries for building each tree, but to sample the documents per query, starting initially with a small number of documents, and then gradually increasing the number of documents chosen. In this setting, each tree in the ensemble would have the same queries (all the queries) but different subsets of documents – the size of the subset is the same across all trees. We think, however, that there is a subtle issue in this setting which is as follows. As explained earlier, in order to achieve good accuracy, the trees of a random forest, given they are "strong" enough, are supposed to be as uncorrelated as possible with one another. Now (Yilmaz & Robertson 2009) show that for the LtR task, given a large pool of queries and documents, if we are to select a training set from it, then sampling more queries with less documents is better than sampling less queries with more documents – better for a rank-learner to learn a ranking function. This means that in the former case, i.e., sampling more queries and less documents, the characteristics of the original training set (i.e., the collection of documents mentioned above) are largely retained in the sampled set. Now back to our setting: for every tree, if we sample all queries and a subset of documents per query, then this is not likely to reduce correlation (which is one of our objectives) because the sampled training set for individual trees, according to the above-mentioned work, is likely to contain *enough information* to be representative of the original training set. Hence we avoid this setting.

As our work may be seen as an experiment to tune a parameter where the parameter is the size of sub-sample to be used for learning a tree, we find the best parameter using a validation set (that is, not used for training), and then we apply our model with the learnt parameter to a separate test set to validate our findings.

5 Model

The LtR algorithm we use can be thought of as a hybrid of the classification and regression settings. For splitting a node, the classification settings (entropy-based gain) are used. For assigning a label to a test instance, regression settings are used, i.e., when a test instance lands on a leaf of a tree, then instead of predicting the majority class label, the algorithm assigns a real score to the test instance which is the average of the labels of all the documents of that leaf. Finally, in order to calculate an IR metric for evaluation, the documents are sorted in decreasing order using these scores. Thus the training and the testing phases adopt a classification and a regression setting respectively. The procedure for building a tree is given in Algorithm 2. In our experiments, each configuration uses a sample of fixed size b (the number of queries), for learning each tree, and varying b constitutes different configurations (cf. the 1st line of Algorithm 2). We omit the pseudo-code of some routines when the meaning is obvious (*chooseRandomFeature(.), Split(.) and entropy(.)*).

Data: Entire (non-bagged) data $\mathcal{T}$ having N features, number of queries to sample b
Result: A tree of the ensemble
$\mathcal{T}_{subsampled} \leftarrow getSubsampledData(\mathcal{T}, b)$; //(cf. Alg. 1)
$BuildNode(\mathcal{T}_{subsampled})$;

Algorithm 2: RandomTree

Data: (Sub-sampled) Data $\mathcal{T}$
if $|\mathcal{T}| < gainThreshold$ **then**
  Store the class distribution of $\mathcal{T}$ in this leaf;
  return;
else
  $K \leftarrow 0$;
  while $K < \sqrt{(N)}$ **do**
    $f \leftarrow chooseRandomFeature(N)$;
    $\langle gain, bestSplitVal \rangle \leftarrow$
    $getMaxGain(\mathcal{T}, f)$;
    if $gain > bestGain$ **then**
      $gain \leftarrow bestGain$;
      $bestSplitPoint \leftarrow \langle f, bestSplitVal \rangle$;
    end
    $K = K + 1$;
  end
  if $bestGain > gainThreshold$ **then**
    $\langle \mathcal{T}_{left}, \mathcal{T}_{right} \rangle \leftarrow$
    $Split(\mathcal{T}, bestSplitPoint)$;
    $BuildNode(\mathcal{T}_{left})$;
    $BuildNode(\mathcal{T}_{right})$;
  else
    Store the class distribution of $\mathcal{T}$ in this leaf;
    return;
  end
end

Algorithm 3: BuildNode

6 Experiments

We now discuss the experiments we have performed to understand the effect of sub-sample size on performance of random forest based LtR algorithms.

Data: Data $\mathcal{T}$, feature f
Result: Best split value for this feature
for $splitVal \in possibleSplits(\mathcal{T}, f)$ **do**
  $\mathcal{T}_{Left}, \mathcal{T}_{Right} \leftarrow Split(\mathcal{T}, splitVal)$;
  $gain \leftarrow entropyGain(\mathcal{T}, \mathcal{T}_{Left}, \mathcal{T}_{Right})$;
  if $gain > bestGain$ **then**
    $bestSplitVal \leftarrow splitVal$;
    $bestGain \leftarrow gain$;
  end
end
return $\langle bestGain, bestSplitVal \rangle$;

Algorithm 4: MaxGain

Data: parentData $\mathcal{T}$, leftChildData $\mathcal{T}_L$, rightChildData $\mathcal{T}_R$
Result: Gain for this split.
return $entropy(\mathcal{T}) - (\frac{|\mathcal{T}_L|}{|\mathcal{T}|}.entropy(\mathcal{T}_L) + \frac{|\mathcal{T}_R|}{|\mathcal{T}|}.entropy(\mathcal{T}_R))$

Algorithm 5: EntropyGain

6.1 Setup

We start with $q = 1$, then gradually increase q such that for a data set we have nearly 25 experiments, ending with the experiment with the original training set (i.e., with no bootstrapping at all).

The *Weka* machine learning toolkit[5] is modified and extended to run the experiments.

Table 1 shows statistics of some publicly available datasets from Letor repository[6] which have been heavily used for the LtR task. In the Letor repository, each dataset is divided into 5 predefined folds for fairness in comparison between different algorithms, and each fold is further divided into a training, a validation and a test set having 3/5ths, 1/5th and 1/5th of the entire data respectively. We run the experiments using these 5 folds, and report the average results. Two widely used rank-based metrics are measured, namely, NDCG@10 and MAP. There are 1000 trees in each ensemble. The terminating conditions to stop further building a tree to check whether there is no gain in terms of entropy (cf. Algorithm 5) for any potential split of that node. We do not impose any restriction on maximum height of a tree or minimum required instances of a node.

6.2 Result Analysis

Figures 1 and 2 show the performance on validation sets in terms of NDCG@10 and MAP (respectively) of the 10 independent runs of each experiment. In both figures the standard deviation is also shown in the right. In the left-hand side plots of these figures, each point is the NDCG@10 (MAP) of a single run (out of 10) of an experiment where a tree of the ensemble at hand is learnt from a $p\%$ of random queries of the original training set – p is indicated by the x-axis. The procedure for computing p is as follows: we start with $b = 1$ for all the three datasets[7], then we increase it by 3, 20 and 50 for Ohsumed, MQ2008, and MQ2007 respectively until $b = |Q|$ (i.e. unless all the queries are used), thereby making the total number of distinct configurations for Ohsumed, MQ2008 and MQ2007 as 22, 25 and 22 respectively. There are 10 points for each p value which show the NDCG@10 (MAP) value for 10 independent runs. For example, for the experiment where each tree of the ensemble uses 60% random queries as its training set, the 10 NDCG@10 (MAP) values are indicated along the vertical line corresponding to p = 60%. A smooth-spline is fitted (with parameter 0.01) in order to capture the trend of a plot (as shown by the black curve). The bar plots in the right-hand side indicate the standard deviations of the NDCG@10 (MAPs) across the 10 runs of each experiment. The values of x-axes of the right-hand side plots indicate the same things as that of the left-hand side plots.

The plots show that the performance of the initial configuration is low (except for Ohsumed) which is anticipated because using only one query is not likely to yield good accuracy. Then performance increases, but after using a certain percentage of the queries, it starts to decrease, and this trend of reduction continues until the end. The reduction of accuracy for the configurations beyond the point $p = 63\%$ is already known to the research community as it causes the trees to be more correlated[8], but our experiments suggest that using standard bootstrapping is not the best choice for the LtR problem.

For Ohsumed dataset, using only one query is one of the best configurations in terms of performance. We conjecture that the reason is that Ohsumed has fewer queries and more documents per query than MQ2008 and MQ2007 (cf. Table 1), so one query is sometimes enough to learn a good ranking function, and because of the large number of base learners used in an ensemble, the overall accuracy is high. Another conjecture is, there are only 63 ($\approx 106.(3/5)$) queries per fold in the training set of Ohsumed dataset which is comparatively smaller than the ensemble size (i.e. 1000). So each query is used for learning approximately 16 ($\approx 1000/63$) trees, whereas this number stands to only 2 ($\approx 1000/((784).(3/5))$) and 1 ($\approx 1000/((1692).(3/5))$) for MQ2008 and MQ2007 respectively. Hence we conjecture that as we grow the ensemble size, similar trend (of witnessing better performance using fewer queries per tree) can be found in MQ2008 and MQ2007 as well. We leave this to future work.

As for variance analysis, we focus more on the MAP since it is a comparatively more stable metric than NDCG@10, but we do not omit the NDCG@10 altogether from the discussion. Accordingly, for MQ2008, the variances of the models (represented by standard deviations) after the first configuration seem to be random in the sense that there is no clear trend in the bar plots after the 1st configuration. For MQ2007, there seems to be a consistent trend of small decrease as the sub-sample size increases – this may be due to the fact that the MQ2007 has comparatively large number of queries, so as the number of queries increases, so does the stability of the models. As for Ohsumed, for both MAP and NDCG@10, no clear trend is seen, and the differences seem to be due to relatively small number of independent experiments (10 only). We conjecture that a reason for insignificant variation of variances across different models[9] is that the large number of base learners used in an ensemble mitigates the possible instability in output.

Figure 3 shows that the training time steadily increases with the increasing percentage of the queries.

Now we need to select the *best* setting (i.e., the number/percentage of queries) from these experiments, and then to apply this setting to a separate test set in order to corroborate the findings. To select the best setting, we consider only absolute performance since the variances of different sub-sample sizes don't vary greatly. For Ohsumed, MQ2008 and MQ2007, the best configurations

[5] http://www.cs.waikato.ac.nz/ml/weka/

[6] http://research.microsoft.com/en-us/um/beijing/projects/letor/

[7] Recall that b is the number of queries to sample.

[8] In fact, the motivation of injecting randomization in the trees came from this insight (Breiman 2001).

[9] Please note that the y-axis in the right-hand side bar plots are quite enlarged, so the differences of variances are actually not very big.

Table 1: Statistics of the Three Datasets Used in Our Experiments.

Data Set	Task	# Queries	# Features	# Relevance Labels	# Query-doc Pairs	Avg. # Docs per Query	# Relevant Docs per Query
Ohsumed	Medical Docs	106	45	3	16000	152	46
MQ2008	Web Search	784	46	3	15000	19	5
MQ2007	Web Search	1692	46	3	69000	41	12

are: 1st (with 1 queries (1.6%)), 2nd (with 21 queries (9%)), and 2nd and 3rd (with 51 and 101 queries (5% and 10%)) respectively – for MQ2007 since only this dataset has a slight trend of decreasing variance as sub-sample size increases, we select two best settings here. Table 2 summarizes the improvements on validation sets of the selected best settings. Across different datasets, only 1.6-10% of the queries are needed to get even better accuracy with similar model variance, and the learning time is reduced by 10-29 times.

Now using the best settings found with the validation sets, we run the experiments on separate test sets. Suggested by Table 2, the number of queries we chose for Ohsumed, MQ2008 and MQ2007 are 1, 41 and 51/101 respectively. Table 3 shows the improvement of accuracy and reduction of learning time of these settings over the baselines (i.e., over 63% sub-sampling).

From the experiments on the test sets, we see that in addition to the increased scalability, on average the accuracy of the models run with our selected best settings is higher than the baselines. This improvement of accuracy is not likely to be a random gain because we have conducted 10 distinct runs of each experiment, and the plots of the Figures 1 and 2 clearly show the average trend of improvement of our configurations over the baselines. We note that in information retrieval, improvement of accuracy by as small amount as 0.01-0.02 (1-2%) in the commonly used metrics such as NDCG and MAP is considered to be considerably better performance (Chapelle et al. 2007), (Xu & Croft 2000).

So far we have not compared performance of random forest based pointwise LtR algorithm with other algorithms. Table 4 shows the comparison between other algorithms whose performance are listed in the Letor website; in addition we run experiments for another state-of-the-art algorithm called LambdaMart[10] (Wu et al. 2010) because it is also a tree ensemble method using the gradient boosting framework[11]. The top performances of a metric of a dataset (i.e. of a row) is shown in **bold face**. The results of the two random forest based algorithms (RF-bt stands for 63% sub-sampling and RF-sb stands for reduced sub-sampling) are copied from Table 3. We see that although very simple as compared to most of the baseline algorithms, in most of the cases the performance of RF-sb (which is the contribution of this paper) is very close to the best value. Let us not forget one inherent benefit of a random forest based algorithm over all other ones listed in Table 4: it is embarrassingly parallelizable as each tree can be leant in parallel. Table 4 also unveils one important aspect: no single algorithm performs consistently better than others across all data sets. This shows that LtR task is per se a complex task.

7 Discussion

We now summarize the observations from the experiments discussed above.

- Our experiments have empirically discovered that for the datasets we have used, the training time can be reduced by 10-29 times thereby increasing scalability of a random forest based pointwise LtR algorithm while achieving better prediction accuracy and similar level of stability (in terms of standard deviation of results across multiple runs). This is because the plots of Figures 1 and 2 show that for an experiment with the best settings, there is a high probability that the performance will be higher than the baseline.
- The findings depend on the datasets, but the optimal % of queries to be used in each tree definitely lies far below the 63% level as currently being used in the LtR literature and even the 50% which is reported by (Friedman & Hall 2007) for classification and regression tasks. is the optimal value to be used in each tree. Here lies the main contribution of our work. Further research can be done in order to find the aspects, if exist, of a training set which influence the optimum percentage. Some aspects to investigate are: number of queries, average number of documents per query, and average ratio of relevant to irrelevant documents per query.
- By reducing the sub-sample size per tree, we decrease the correlation among the trees which is supposed to cause the error rate of the ensemble to decrease. However, as mentioned in the original paper of random forest (Breiman 2001), there is another aspect which controls the accuracy of the ensemble – the strength of the individual trees. By reducing subsample size, the strength is likely to decrease. In spite of this, we still have achieved slightly better performance – this means that the positive effect of decreasing correlation *compensates* for any negative effect of reducing strength. Inspired by this conjecture, we are currently working on as to how to retain/increase strength of individual trees in spite of reducing subsample size. One idea is to increase the parameter K (randomly chosen features at each split). However, since this will increase the training time linearly, we shall probably loose the benefit of scalability which is currently present in our experiments. So there may be some optimal choice of sub-sample size and K which will both increase accuracy of ensemble and decrease the training time as compared to the baseline (where the baseline uses a bootstrapped sample and $K = \sqrt{\#features}$).
- The findings discovered in this work are in fact somewhat pessimistic. For example, for MQ2008 and MQ2007 datasets, we have showed that the 2nd configurations (having 21 and 51 queries respectively) have the best accuracy. So here the granularity of our experiments could be improved in order to identify the exact number of queries that results in the best performance. Work along this line is in progress.
- There may be a relation between the ensemble size, T and number of queries used to learn each tree, b. For example, if $b = 1$, then each query appears in approximately $\frac{T}{b}$ number of trees. We are working to elicit some sort of relationship between these two parameters.

[10] The source code has been taken from `https://code.google.com/p/jforests/`).

[11] The parameters of LambdaMart are set as follows: number of trees = 1000, number of leaves per tree = 7, learning rate = 0.5.

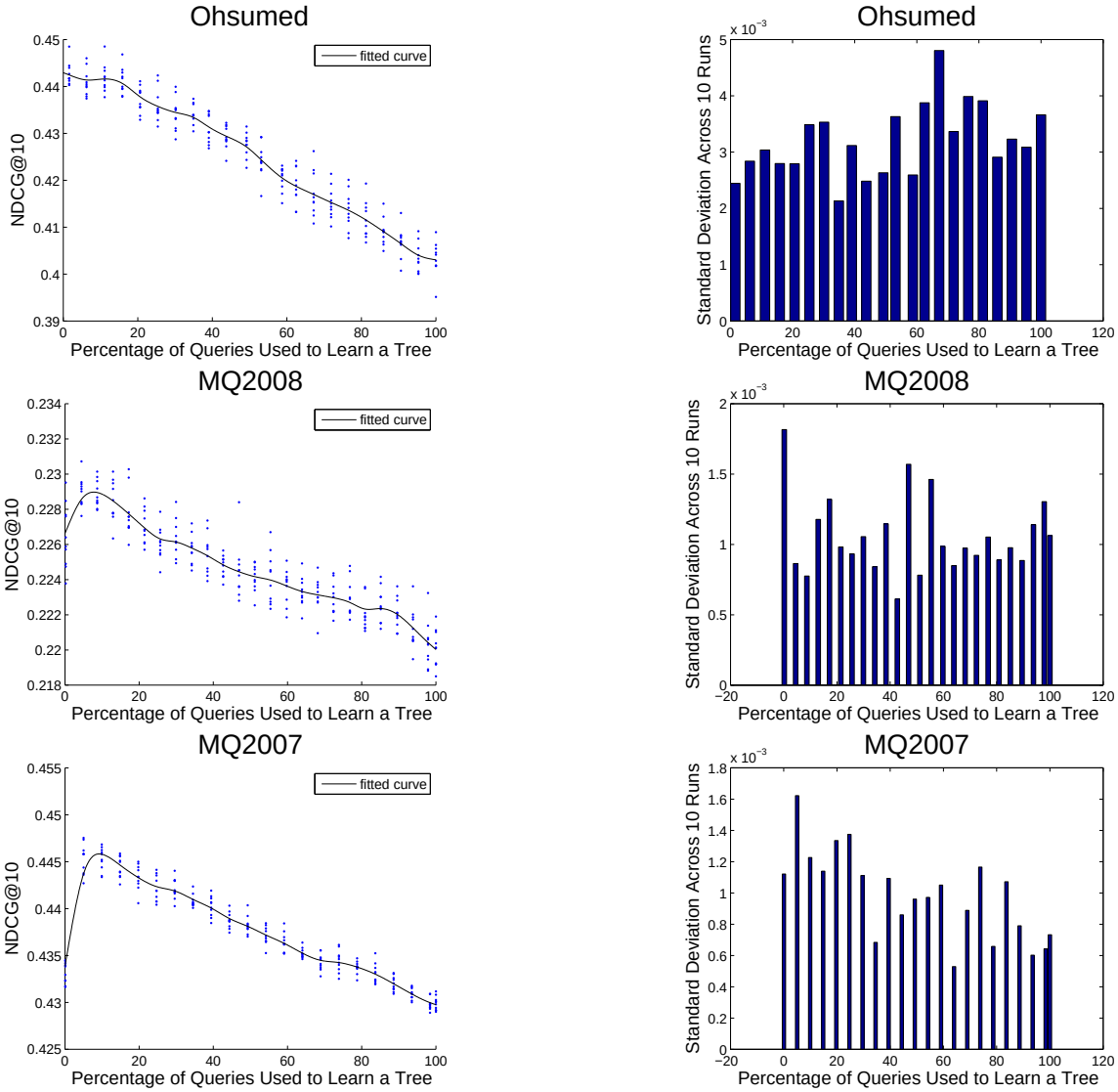


Figure 1: Performance (NDCG@10) and Standard Deviation of Various Datasets for 1000 trees in an ensemble (With Validation Sets): 1st, 2nd and 3rd rows are for Ohsumed, MQ2008 and MQ2007 respectively. In the left plots, each point is the NDCG@10 of a single run (out of 10) of an experiment (using the % of queries per tree indicated by the x-axis). There are 10 points for a single value in the x-axis implying the NDCG@10 for 10 independent runs. A smooth-spline is fitted (with parameter 0.01). The bar plots in the right-hand side indicate the standard deviations of the NDCG@10s across the 10 runs.

8 Conclusions and Future Work

Random forests has two components which control its accuracy, namely, (1) the correlation between the trees, and (2) the strength of the individual trees. In this work, we have investigated into the former of these two in the context of LtR task. We have used reduced sub-sample per tree to decrease correlation. Experiments show that we can achieve better performance while achieving substantial reduction in the training time – training time is an important factor for large scale LtR task. The stability of the models is also roughly of similar level of the baseline.

We are now working on several directions as mentioned in the Discussion section such as investigating other options for reducing correlation (as well as the training time) and on the better understanding the relationship between the ensemble size and number of queries.

References

Aslam, J. A., Kanoulas, E., Pavlu, V., Savev, S. & Yilmaz, E. (2009), Document selection methodologies for efficient and effective learning-to-rank, *in* 'Proc. of the 32nd international ACM SIGIR conf. on Research and development in information retrieval', ACM, pp. 468–475.

Breiman, L. (2001), 'Random forests', *Machine learning* **45**(1), 5–32.

Chapelle, O., Chang, Y. & Liu, T.-Y. (2011), Future directions in learning to rank, *in* 'JMLR Workshop and Conference Proceedings', Vol. 14, pp. 91–100.

Chapelle, O., Le, Q. & Smola, A. (2007), Large margin optimization of ranking measures, *in* 'NIPS Workshop: Machine Learning for Web Search'.

Cossock, D. & Zhang, T. (2006), 'Subset ranking using regression', *Learning theory* pp. 605–619.

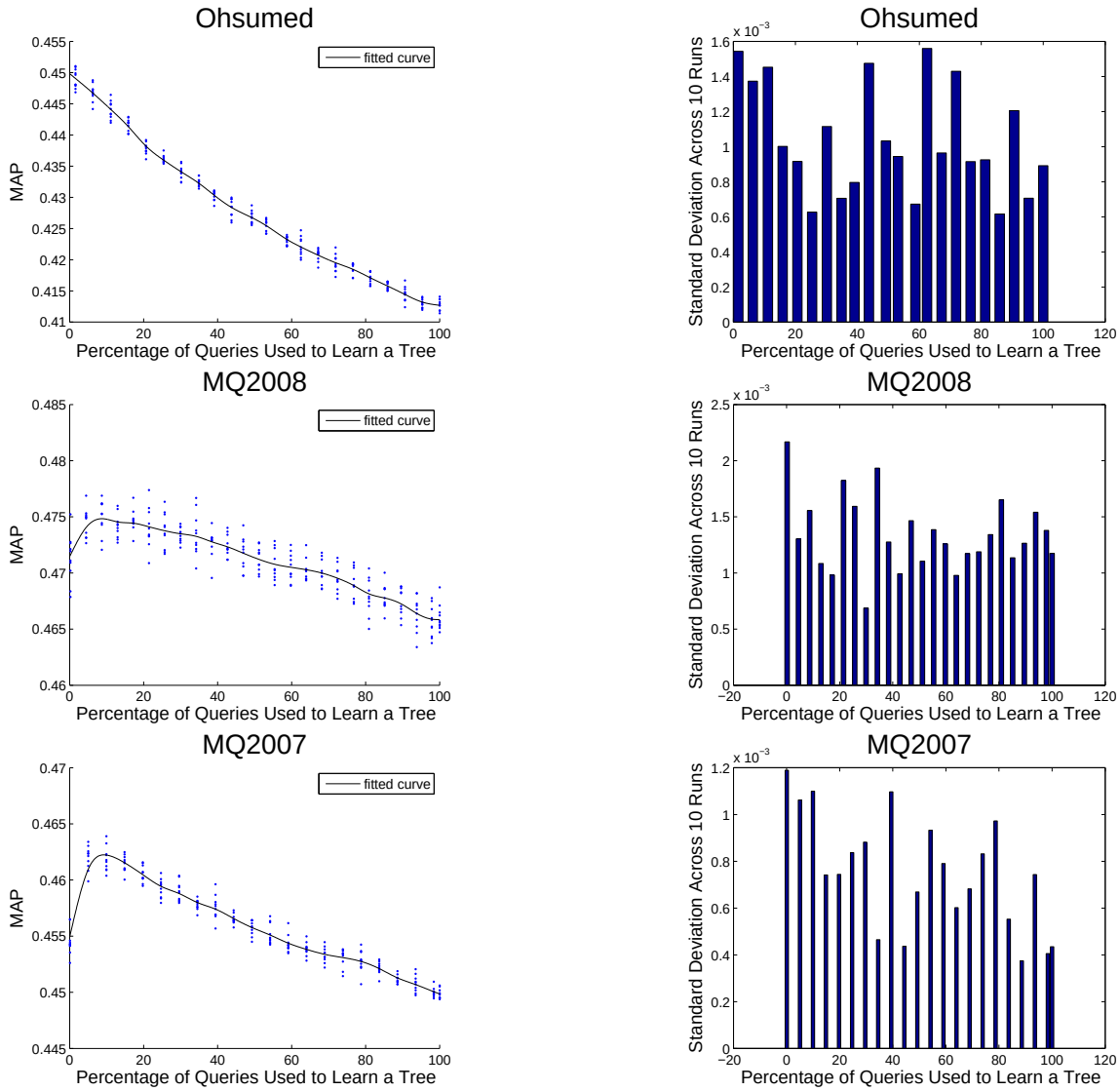


Figure 2: Performance (MAP) and Standard Deviation of Various Datasets for 1000 trees in an ensemble (With Validation Sets): Please See the Caption of Figure 1 for Description.

Criminisi, A. & Shotton, J. (2013), *Decision forests for computer vision and medical image analysis*, Springer.

Donmez, P. & Carbonell, J. G. (2008), Optimizing estimated loss reduction for active sampling in rank learning, *in* 'Proc. of 25th international conf. on Machine learning', ACM, pp. 248–255.

Friedman, J. H. (2001), 'Greedy function approximation: a gradient boosting machine.(english summary)', *Ann. Statist* **29**(5), 1189–1232.

Friedman, J. H. (2002), 'Stochastic gradient boosting', *Computational Statistics & Data Analysis* **38**(4), 367–378.

Friedman, J. H. & Hall, P. (2007), 'On bagging and nonlinear estimation', *Journal of statistical planning and inference* **137**(3), 669–683.

Ganjisaffar, Y., Caruana, R. & Lopes, C. V. (2011), Bagging gradient-boosted trees for high precision, low variance ranking models, *in* 'Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval', ACM, pp. 85–94.

Geurts, P., Ernst, D. & Wehenkel, L. (2006), 'Extremely randomized trees', *Machine learning* **63**(1), 3–42.

Geurts, P. & Louppe, G. (2011), Learning to rank with extremely randomized trees, *in* 'JMLR: Workshop and Conference Proceedings', Vol. 14.

Ham, J., Chen, Y., Crawford, M. M. & Ghosh, J. (2005), 'Investigation of the random forest framework for classification of hyperspectral data', *Geoscience and Remote Sensing, IEEE Transactions on* **43**(3), 492–501.

Hastie, T., Tibshirani, R. & Friedman, J. (2009), 'The elements of statistical learning'.

Kosorok, M. R., Ma, S. et al. (2007), 'Marginal asymptotics for the large p, small n paradigm: with applications to microarray data', *The Annals of Statistics* **35**(4), 1456–1486.

Li, H. (2011), 'Learning to rank for information retrieval and natural language processing', *Synthesis Lectures on Human Language Technologies* **4**(1), 1–113.

Li, P., Burges, C. & Wu, Q. (2007), 'Learning to rank using classification and gradient boosting', *Advances in neural information processing systems* **19**.

Liu, T.-Y. (2011), *Learning to rank for information retrieval*, Springerverlag Berlin Heidelberg.

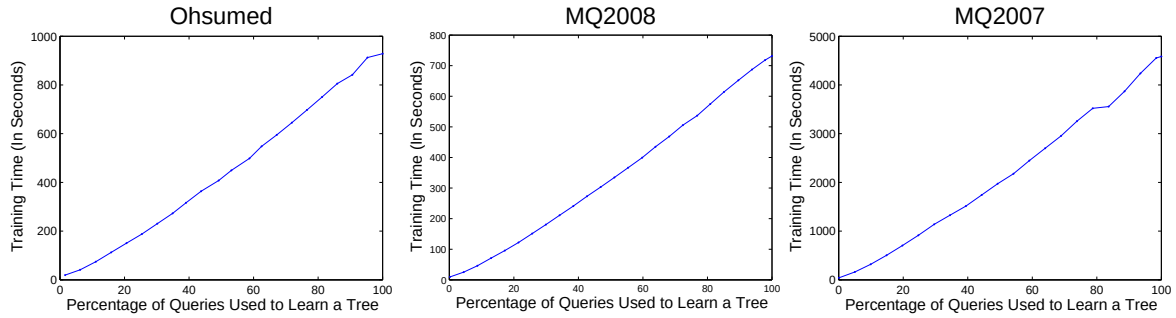


Figure 3: Training Time (to Learn an Ensemble of 1000 Trees) Vs % of Queries Used to Learn Each Tree (With Validation Sets): (a) Ohsumed (b) MQ2008 (c) MQ2007.

Table 2: Summary of Reduction in Training Time and Improvement of Performance for Validation Set.

Data Set	# (and %) of Queries Used in Reduced Subsample Per Tree	# (and %) of Queries Used in Subsample Per Tree in Baseline	Relative Training Time (Speed-up over Baseline)	NDCG@10 of Reduced Sample (Avg of 10 Runs and Std. Dev.)	NDCG@10 of Baseline (Avg of 10 Runs and Std. Dev.)	MAP of Reduced Sample (Avg of 10 Runs and Std. Dev.)	MAP of Baseline (Avg of 10 Runs and Std. Dev.)
Ohsumed	1 (1.6%)	40 (63%)	29 times	0.4427 (0.0024)	0.4187 (0.0039)	0.4490 (0.0015)	0.4221 (0.0016)
MQ2008	21 (4.5%)	301 (64%)	17 times	0.2290 (0.0009)	0.2233 (0.0085)	0.4745 (0.0013)	0.4703 (0.0098)
MQ2007	51 (5%)	651 (64%)	17 times	0.4453 (0.0016)	0.4353 (0.0005)	0.4619 (0.0011)	0.4539 (0.0006)
	101 (10%)	651 (64%)	9 times	0.4453 (0.0012)	0.4353 (0.0005)	0.4619 (0.0011)	0.4539 (0.0006)

Long, B., Chapelle, O., Zhang, Y., Chang, Y., Zheng, Z. & Tseng, B. (2010), Active learning for ranking through expected loss optimization, *in* 'Proceedings of the 33rd international ACM SIGIR conf. on Research and development in information retrieval', ACM, pp. 267–274.

Macdonald, C., Santos, R. L. & Ounis, I. (2012), 'The whens and hows of learning to rank for web search', *Information Retrieval* pp. 1–45.

Manning, C. D., Raghavan, P. & Schütze, H. (2008), *Introduction to information retrieval*, Vol. 1, Cambridge University Press Cambridge.

Mohan, A. (2010), 'An empirical analysis on point-wise machine learning techniques using regression trees for web-search ranking'.

Qi, Y. (2012), Random forest for bioinformatics, *in* 'Ensemble Machine Learning', Springer, pp. 307–323.

Robnik-Šikonja, M. (2004), Improving random forests, *in* 'Machine Learning: ECML 2004', Springer, pp. 359–370.

Wu, Q., Burges, C. J., Svore, K. M. & Gao, J. (2010), 'Adapting boosting for information retrieval measures', *Information Retrieval* **13**(3), 254–270.

Xu, J. & Croft, W. B. (2000), 'Improving the effectiveness of information retrieval with local context analysis', *ACM Transactions on Information Systems (TOIS)* **18**(1), 79–112.

Yilmaz, E. & Robertson, S. (2009), Deep versus shallow judgments in learning to rank, *in* 'Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval', ACM, pp. 662–663.

Yu, H. (2005), Svm selective sampling for ranking with application to data retrieval, *in* 'Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining', ACM, pp. 354–363.

Table 3: Summary of Reduction in Training Time and Improvement of Performance for **Test Set** (Using Best Settings from Validation Sets).

Data Set	**# (and %) of Queries Used in Reduced Subsample Per Tree**	**# (and %) of Queries Used in Subsample Per Tree in Baseline**	**Relative Training Time (Speed-up over Baseline)**	**NDCG@10 of Reduced Sample (Avg of 10 Runs and Std. Dev.)**	**NDCG@10 of Baseline (Avg of 10 Runs and Std. Dev.)**	**MAP of Reduced Sample (Avg of 10 Runs and Std. Dev.)**	**MAP of Baseline (Avg of 10 Runs and Std. Dev.)**
Ohsumed	1 (1.6%)	40 (63%)	29 times	0.4443 (0.0022)	0.4317 (0.0033)	0.4473 (0.0009)	0.4221 (0.0013)
MQ2008	21 (4.5%)	301 (64%)	17 times	0.2292 (0.0008)	0.2238 (0.0011)	0.4736 (0.0016)	0.4701 (0.0012)
MQ2007	51 (5%)	651 (64%)	17 times	0.4447 (0.0013)	0.4363 (0.0010)	0.4619 (0.0012)	0.4533 (0.0004)
	101 (10%)	651 (64%)	9 times	0.4453 (0.0016)	0.4363 (0.0010)	0.4624 (0.0004)	0.4533 (0.0004)

Table 4: Comparison Between Random Forest Based Pointwise Algorithms and Various Algorithms. The top performance is shown in **boldface**. The Algorithms are: ListNet (LN), AdaRank-MAP (AM), AdaRank-NDCG (AN), RankBoost (RB), RankSVM-Primal (RSP), RankSVM-Struct (RSS), LambdaMart (LM), RF-bootstrapped (RF-bt), and RF-reduced-subsampled (RF-sb). The last two algorithms' (RF-bt and RF-sb) results are copied from Table 3.

Data	**Metrics**	LN	AM	AN	RB	RSP	RSS	LM	RF-bt	RF-sb
Ohsumed	N@10	0.441	0.4429	0.4496	0.4302	0.4504	**0.4523**	0.4367	0.4317	0.4443
	MAP	0.4457	0.4487	**0.4498**	0.4411	0.4446	0.4478	0.4173	0.4221	0.4473
MQ2007	N@10	0.4440	0.4335	0.4369	0.4464	0.4436	0.4439	**0.4484**	0.4363	0.4453
	MAP	0.4652	0.4577	0.4602	0.4662	0.4659	0.4644	**0.4675**	0.4533	0.4624
MQ2008	N@10	0.2303	0.2288	**0.2307**	0.2255	0.2279	0.2309	0.2302	0.2238	0.2292
	MAP	0.4775	0.4764	**0.4824**	0.4775	0.4744	0.4696	0.4751	0.4701	0.4736

Optimized Pruned Annular Extreme Learning Machines

Lavneet Singh and **Girija Chetty**
Faculty of ESTEM, University of Canberra, Australia
Lavneet.singh@canberra.edu.au, girija.chetty@canberra.edu.au

Abstract:-

Data mining with big datasets and large samples can be problematic, due to increase in complexity and computational times, and bad generalization due to outliers. Using the motivation from extreme learning machines (ELM), in this paper, we propose a novel approach based on annular ELM, involving RANSAC multi model response regularization. Experimental results on different benchmark datasets showed that proposed algorithm based on annular ELM can optimally prune the hidden nodes, and allow better generalization and higher classification accuracy to be achieved as compared to other algorithms, including SVM and OP-ELM for binary and multi-class classification and regression problems.

Keywords:- Extreme Learning Machine, RANSAC, Regularization, Classification, Regression

1. INTRODUCTION

Neural Networks have been extensively used in many fields due to their ability to approximate complex nonlinear mappings directly from the input sample; and to provide models for a large class of natural and artificial phenomena that are difficult to handle using classical parametric techniques. There are many algorithms for training neural Networks like back propagation, Support Vector Machine (SVM), Hidden Markov Model (HMM) etc. One of the disadvantages of the Neural Network is the learning time involved.
In earlier single feed forward networks, the parameter needs to be tuned based on existing dependency between different layers of parameters (weights and biases). For training and tuning the parameters, gradient descent based methods are traditionally used for learning of feed forward neural networks. Gradient descent based learning methods are slow due to slower learning process and rapid convergence to local minima. Thus, in order to avoid these learning issues, iterative learning steps can be very promising for obtaining better learning performance. Some of the earlier work has shown the benefits of single layer feedforward networks – aka SLFNs (with N hidden nodes) [1]with random chosen input weights and hidden layer biases (as random hidden nodes) to learn exactly N distinct observations. Unlike, the previously proposed manually iterative tuning of parameters of SLFNs, it would be worthwhile to try a better learning strategy for feed forward neural networks, by randomly putting the input weights and hidden layer biases in real time problems.
(Huang, Zhu, & Siew, 2006)[1] proposed a new novel algorithm known as Extreme Machine Learning (ELM) for single hidden layer feed forward neural network which has less computational time and faster speed even on large datasets. The main working core of ELM is random initialization of weights rather than learning through slow process via iterative gradient based learning, such as back-propagation learning proposed in (Abid, Fnaiech, & Najim, 2001)[2]. In Extreme machine learning, the number of hidden nodes and their weights are randomly assigned, which distinguishes the linear differentiable between the output of hidden layer and output layer. The output weights can be determined by linear least square solution of hidden layer output through activation function and the data sample targets.

For N arbitrary distinct samples (x_i, t_i), where $x_i = [x_{i1}\ x_{i2},\ldots,\ x_{in}]^T \in R^n$ and $t_i = [t_{i1},\ t_{i2},\ldots,t_{im}]^T \in R^m$, standard SLFNs with N hidden nodes and activation function *g(x) can be* mathematically modelled as

$$\sum_{i=1}^{\tilde{N}} \beta_i g_i\left(x_j\right) = \sum_{i=1}^{\tilde{N}} \beta_i g\left(w_i \cdot x_j + b_i\right) = 0, \quad (1),$$
$$j = 1, \ldots, N,$$

where $w_i = \left[w_{i1}, w_{i2}, \ldots, w_{in}\right]^T$ is the weight vector connecting the f[th] hidden node and the input nodes, $\beta_i = \left[\beta_{i1}, \beta_{i2}, \ldots, \beta_{in}\right]^T$ is the weight vector connecting the f[th] hidden node and the output nodes, and b_i is the threshold of the i[th] hidden node. $w_i \cdot x_j$ denotes the inner product of w_i and x_j. The output nodes are chosen as linear in this paper.

That standard SLFNs with N hidden nodes with activation function $g(x)$ can approximate these N samples with zero error means that $\sum_{j=1}^{\tilde{N}} \left\| o_j - t_j \right\| = 0,$ i.e., there exist β_i, w_i and b_i such that

[1] Copyright (c) 2014, Australian Computer Society, Inc. This paper appeared at Australasian Data Mining Conference (AusDM 2014), Brisbane, 27-28 November 2014. Conferences in Research and Practice in Information Technology, Vol. 158. Richi Nayak, Xue Li, Lin Liu, Kok-Leong Ong, Yanchang Zhao, Paul Kennedy Eds. Reproduction for academic, not-for profit purposes permitted provided this text is included.

$$\sum_{i=1}^{\tilde{N}} \beta_i g_i\left(w_i \cdot x_j + b_i\right) = t_j, \quad j = 1,...,N. \quad (2)$$

The above *N* equations can be written compactly as

$$H\beta = T,$$

Where

$$H\left(w_1,...,w_{\tilde{N}}, b_1...,b_{\tilde{N}}, x_1,...,x_N\right)$$

$$= \begin{bmatrix} g\left(w_1 \cdot x_1 + b_1\right) & \cdots & g\left(w_{\tilde{N}} \cdot x_1 + b_{\tilde{N}}\right) \\ \vdots & \cdots & \vdots \\ g\left(w_1 \cdot x_N + b_1\right) & \cdots & g\left(w_{\tilde{N}} \cdot x_N + b_{\tilde{N}}\right) \end{bmatrix}_{N \times \tilde{N}} \quad (3)$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{bmatrix}_{\tilde{N} \times m} \text{ and } T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N \times m} \quad (4)$$

H is called the hidden layer output matrix of the neural network; the i^{th} column of H is the i^{th} hidden node output with respect to inputs $x_1, x_2..., x_N$.

As named in (Huang et al., 2006)[1], H is called the hidden layer output matrix of the neural network; the i^{th} column of H is the i^{th} hidden node output with respect to inputs. The authors (Huang, Wang, & Lan, 2011)[6] presented a comprehensive survey on extreme learning machines and its applications. Optimally pruned extreme leaning machine (OP-ELM) algorithm which is an extension of original ELM algorithm with pruning of neurons using ranking multi-response sparse regression (MRSR) method to design optimal neural architecture removing irrelevant variables was proposed by (Yoan et al., 2010)[13]. (Martínez-Martínez et al., 2011)[8] proposed a new strategy to prune the ELM networks using regularized regression methods to acquire optimal tuned parameters. The algorithm can acquire optimal tuned parameters by identifying the degree of relevance of the weights that connects the *k-th* hidden element with the output layer using lasso and ridge regression. Regularized version of least squares regression with several penalties on coefficient vector are used to remove the irrelevant or low relevance hidden nodes to achieve compact neural networks. (Singh & Chetty, 2012)[12] proposed LDA-ELM for classification of brain abnormalities in magnetic resonance images using pattern recognition and machine learning. (Lavneet Singh, 2012)[7] proposed a Novel Approach for protein Structure prediction using Pattern Recognition and Extreme Machine Learning.

The proposed variation of ELM strongly validates that the input weights and hidden layer biases of SLFN can be randomly assigned with infinitely differentiable activation function of hidden layers. By randomly choosing input weights and hidden layer biases, SLFN can act as a linear system on the output weights with linkage of hidden layer to the output layer. These output weights of SLFNs can be determined through simple generalised inverse operation of the hidden layer output matrices. Although the network training based on utilizing ELM is faster than other algorithms and can have improved generalization performance, there are still two major unresolved problems: (1) The only parameter that needs to be determined for ELM is the number of hidden nodes in the hidden layer. In former studies, this parameter is usually obtained by trial and error method that may not be optimal. How to choose the most suitable network structures for different applications is still unknown. (2) ELM sometimes requires a large network structure (large number of hidden nodes in the hidden layer) due to the random process in the initial stage. The issue is whether the network complexity can be further reduced without affecting the generalization performance.

Pruning method is one of the useful heuristic approaches to address the problem of network architecture design. Some researchers have tried to obtain compact ELM networks based on pruning methods. Rong et al. [16] has proposed a pruned ELM (P-ELM) for pattern classification applications, which starts with a large network and then eliminates the hidden nodes with low relevance to the class label. To overcome the drawbacks of regularization or penalty methods, motivated by the idea of using sparse modesl and removing redundant variables for better generalization and prediction accuracy, in this paper we propose a novel approach based on RANSAC multi model response regularization, which implements a L1 penalty for the output weights by performing RANSAC multi model response regression between the hidden and output layer.

Rest of the paper is organised as follows. Next Section describes background for the proposed scheme and provides a description of the Annular ELM and Regularized ELM. The details of RANSAC Multi-model response regularization is presented in Section 3, and details of the experimental validation of the proposed scheme on benchmark datasets is presented in Sections 4. The paper concludes in Section 5 with

some key conclusions of this work and plans for further study.

2. Background

2.1 Annular ELM

Circular Back Propagation (CBP) networks [5] improve over the basic formulation of MLP; the CBP model augments the input vector by one additional dimension, which is computed as the norm of the input vector itself. In a classic set-up involving a single layer network, the estimation process supported by the enhanced, CBP network is expressed as

$$y_{CBP}(x) = \xi\left(b + \sum_{j=1}^{N_h}\left[wj\xi\left(\hat{b}_{j,0} + \sum_{k=1}^{M} \hat{w}_{j,k} x_k + \hat{w}_{j,M} + 1\|x\|^2 \right)\right]\right) \quad (5)$$

Based on circular back-propagation network, we propose a new architecture networ for extreme learning machines known as Annular ELM. The first formation of the annular ELM augments by adding one more dimension in both training and testing data which is computed as

Training $data_{ij}$ = Training $data_{ij+1}$

Where i = {1……m} as number of observation and j = (1…….n) as number of features. Similarly

Testing $data_{pj}$ = Testing $data_{pj+1}$

As the annular topology is implemented to the input layer and feeded into the hidden layer in ELM network, the random weights of the hidden layer and the bias is redefined as

$$Aj(x, \hat{w}_j, \hat{b}_j) = \left(\xi(\hat{w}_j . \bar{x} + \hat{b}_j)\right) \quad (6)$$

which could be rewritten as

$$Aj(x, \hat{w}_j, \hat{bj}) = \xi(z_j . \|x - c_j\|^2 + \bar{b}_j) \quad (7)$$

Where

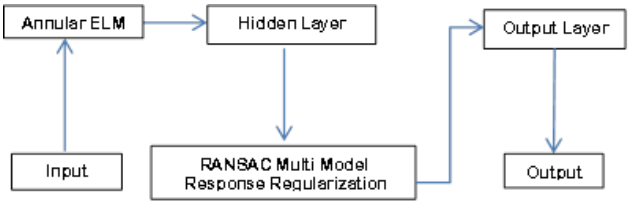
$$Z_j = \hat{w}_{j,M+1} \quad (8)$$

$$cj = \left[\frac{\hat{w}_{j,1}}{2\hat{w}_{j,M}+1}, ..., -\frac{\hat{w}_{j,M}}{2\hat{w}_{j,M}+1} \right] \quad (9)$$

$$\bar{b}_j = \frac{1}{\hat{w}_{j,M+1}}\left(\sum_{k=1}^{M} \frac{\hat{w}^2_{j,k}}{4\hat{w}_{j,M+1}} - \hat{b}_j \right) \quad (10)$$

Input data (X_{ij}) = [X_{ij},……X_{ij+1}]

Figure 1:- Architectural Design of Annular ELM based on RANSAC Multi Model Response Regularization



The annular based ELM is able to map both linear and circular separation boundaries by boosting the ability of ELM network. In annular based ELM, the initial hidden weights are chosen randomly only along with the bias too similar to ELM. But, later new random weights and bias are approximated by applying proposed annular functions to estimate the output of hidden layer, known as hidden matrix.

Later, proposed RANSAC multi response regularization is applied to the output of hidden layer using annular ELM topology to prune the hidden units for better generalization and higher accuracy.

2.2 Regularized Extreme Learning Machine

To resolve these limitations of ELM, constructive and heuristic approaches have been proposed in the literature. In most recent years, regularization or penalty approach seems to be significant in resolving the ELM limitations. As in extreme machine learning, there is linear behaviour between hidden layer and output layer, thus as a problem of linear regression, regularization helps to reduce the number of predictors in hidden layer by using sparse model.

Significant work have been done in past for better generalization, faster learning and rate of convergence. But, unfortunately, ELM also suffers with some limitations as outliers, irrelevant variables and number of hidden nodes. To resolve these limitations of ELM, constructive and heuristic approaches have proposed in the literature.

In most recent years, regularization or penalty approach seems to be significant in resolving the ELM

limitations. As in extreme machine learning, there is linear behaviour between hidden layer and output layer, thus as a problem of linear regression, regularization helps to reduce the number of predictors in hidden layer by using sparse model. Least square solution with regularization is fitted to the model to find the nonzero coefficients as output weights for output layer. Using regularized sparse model, most of the predictors are moved to zero with increase in lambda. Thus, it creates a sparse model of output of hidden layer of finding the beta coefficients with respect to lambda with minimum deviationor minimum convergence with respect to mean square error. Regularization is applied to regression problems to select the relevant hidden units, by negotiating a trade-off between over fitting with respect to network size. The big architectures are selected for the network because regularization approach prunes the network with optimal hidden neurons. In regularizationbased ELM, the weights of the input layer connected to the hidden layer are chosen randomly. The output weights of the output layer are determined through regularized regression removing hidden units with an optimized size of the network. The approach used for regularization for ELM can be stated as:Lasso regularization as L_1 penalty

- Ridge Regression or L_2 penaltyElastic net combining both L_1 and L_2 penalty

To define the general case of regularization, as a single output regression represented as –

$$Y = Xw + \mathcal{E} \tag{11}$$

with $X = (X_1, X_2, \ldots\ldots X_n)^T$ are the inputs of a dataset and $Y = (Y_1, Y_2, \ldots\ldots Y_n)^T$ are the output and $W = (w_1, w_2, \ldots\ldots w_p)^T$ are the regression weights of the hidden layer. As discussed, the model possess a linear regression between input layer and output layer, thus the simple least square solution (OLS) is a heuristic approach to solve single output regression formulated as

$$\min_{\hat{w}} (y - X\hat{w})^T (y - X\hat{w}) \tag{12}$$

Or in least square form

$$\min_{\hat{w}} \sum_{i=1}^{n} (y_i - x_i\hat{w})^2 \tag{13}$$

with $\hat{w} = (\hat{w}_1, \ldots\ldots \hat{w}_n)^T$ the estimated regression weights.

The solution of equation (7) is then obtained by a pseudo inverse (Moore- Penrose) as

$$\hat{w}_{OLS} = (X^T X)^{-1} X^T y \tag{14}$$

Moore Penrose is not useful in every numerical problem if X in (Equation 8) is not full rank.

To improve better generalization and prediction accuracy, OLS technique doesn't provide a complete solution to remove irrelevant variables. Further, OLS doesn't use sparse models either. to get related variables with respect to output. To resolve these issues with simple OLS approach, regularization factors or penalty approach added to minimization cost function in (Equation 8) to get sparse model of OLS (acquire sparse model which try to shift most of the irrelevant variables to zero). Regularization or penalty term lambda with its weights added to minimization problem with its nonzero coefficients can result in beta coefficients of particular model to be obtained.

3. RANSAC Multi Model Response Regularization

The RANSAC (Random Sample Consensus) algorithm was proposed by (Fischler & Bolles, 1981) to estimate the parameters of a certain model from a set of data with large number of outliers. RANSAC approach takes out the outliers from a data if it doesn't fit with a set of parameters within the error threshold with respect to maximum deviation. RANSAC can handle outliers greater than 50% of the entire dataset known as breakdown point.
RANSAC first hypothesizes minimum sample sets MSSs) randomly selected from the input dataset and the parameters of the model are estimated using MSS. To test the estimated parameters of the model using MSS, RANSAC checks which element of the entire dataset is contained within the defined model known as consensus set (CS). RANSAC rank the consensus with a set of iterations with respect to estimated probability at certain threshold. RANSAC is extensively used in computer vision, motion detection and features matching of images and is optimized using different parameters (Raguram, Frahm, & Pollefeys, 2008),(Nistér, 2005),(Chum, Matas, & Kittler, 2003)[10,11,3].

3.1 RANSAC Multi Model Response Regularization for Regression problems

To implement the RANSAC on regression problems, we propose RANSAC multi model response

regularization approach, which implements the sequential RANSAC on multiple models. To implement RANSAC, which in our case, are the irrelevant hidden nodes as predictor variables, and H is the hidden matrix as input from Equation 8. In our case, the output weights follow a linear regression between hidden and output layer defined as:

$$Y = mx + \epsilon$$

Where Y is the output of instances of data, m is the predictor's weights or slope and x is the input data and c is the constant. The output weights follow a linear regression between hidden and output layer defined as

$$Y = \text{Output weights} * H + \epsilon \quad (15)$$

Where

$$H = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \cdots & g(w_{\tilde{N}} \cdot x_1 + b_{\tilde{N}}) \\ \vdots & \cdots & \vdots \\ g(w_1 \cdot x_N + b_1) & \cdots & g(w_{\tilde{N}} \cdot x_N + b_{\tilde{N}}) \end{bmatrix}_{N \times \tilde{N}} \quad (16)$$

$$OutputWeights = Multi\_RANSAC \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \cdots & g(w_{\tilde{N}} \cdot x_1 + b_{\tilde{N}}) \\ \vdots & \cdots & \vdots \\ g(w_1 \cdot x_N + b_1) & \cdots & g(w_{\tilde{N}} \cdot x_N + b_{\tilde{N}}) \end{bmatrix}_{N \times \tilde{N}} * T \quad (17)$$

$$CS = RANSAC\left(\sum_{W=1}^{m}\left(\sum_{j=1}^{n} D_{W\,j}\right)\right) \quad (18)$$

$Dx_{wj} = \{x_{11}, x_{12} \ldots\ldots x_{wj}\}$ be the sets of data H with w^{th} observations in rows and j^{th} as hidden nodes predictors in columns of D matrix. For regression problems, sequential RANSAC is implemented on the set of all inliners, D that are generated by W different models where $W_m = rand(H_M)$. The numbers of models are randomly generated using 20 % of the input data.

Multi Models are selected from hidden matrix and RANSAC is implemented by bootstrapping so as to get maximum sparse coefficients of consensus set (CS) for hidden nodes. Multi models from hidden matrix are randomly selected as 20 % of data so as to get maximum sparse CS coefficients to prune the hidden nodes. CS ranking helps to determine zero and non-zero coefficients of hidden nodes which is efficiently determine by bootstrapping of RANSAC on selected multi models.

To estimate the parameters of W models, each one is represented by *k* dimensional parameter vector θ_w at each iteration, *iter*. CS is estimated using (minimum sample set) MSS of each W model. The iteration is run M times which is calculated before and after removing the inliners from data D. The total number of inliners at iteration *iter* is less than total number of inliners at iteration *iter-1*. The whole formulation of multiple RANSAC response is defined in (Equation 17).

The set of all inliners D generated by W different models has cardinality CS and can be defined as: $N_I = (N_{I,1} + N_{I,2} + \ldots\ldots\ldots N_{I,W})$

As RANSAC is a parametric model, the set of parameters need to be defined before implementing RANSAC. These set of parameters can be defined as:

- Epsilon = False alarming rate as the probability of the algorithm throughout all the iterations will never sample a MSS containing only inliers
- Probability of inlier = the probability that a point whose fitting error is less or equal than is actually an inlier
- Sigma = Gaussian noise
- Estimate function = function that returns the estimate of the parameter vector starting from a set of data.
- Mean square function = function that returns the fitting error of the data.
- CS ranking Algorithm = Algorithm to rank the CS of data
- Minimum number of iterations
- Maximum number of iterations
- W = 20% of the training data

Let $M_w(\theta_w)$ defines the manifold of dimension k_w of all points with respect to parameter $\theta_w \in R^{kw}$ for the specified model for $1 \le w \le W$ with a subset S_w from D of k_w elements at iteration i called minimal sample set (MSS). To estimate the parameters of W models, each one represented parameter vector θ_w. At each i iterations, MSS for each W model is defined and CS is estimated removing all outliners.
To combine and fuse the estimated CS computed from i(W) iterations, the whole RANSAC multi model response algorithm can be summarize as follows

3.2 RANSAC Multi Model Response Regularization Algorithm

$S_w^{(i)} = S_{(i)}(\theta_w)$ be the CS of w^{th} model at i^{th} iteration.
The all combined updated CS of $S_{(i)}(\theta_w)$ is updated as
$M_{(iter)} = 100$; $i = 0$
For $i \le M_{iter}$(maximum number of iterations)
Do $i = i + 1$
$\{S_w^{(i-1)}\},\{S_w^{(i)}\}$
While $1 \le w \le W$
$$S^i(\theta) = S^{(i)}\theta \cup S^{(i-1)}\theta$$
W=w+1
Return $\{S_w^{(i)}\}$

To reduce the number of hidden units with respect to ranked CS estimated using RANSAC multi-response algorithm is calculated as

$$\text{Hidden Layer Output } (H_1) = S^{(i)}\theta * H \quad (19)$$

which reduces the hidden matrix as hidden layer output H_1 into zero and nonzero coefficients of ranked elements with respect to estimated CS. Nonzero coefficients are extracted from sparse matrix which reduces the no of hidden units to which are ranked less giving the hidden units coefficients highly correlated. Thus

$$\text{Output weights} = (H_1^T H_1)^{-1} H_1^T T \quad (20)$$

3.3 RANSAC Multi Model Response Regularization for binary and multiclass problems

The proposed RANSAC multi model response regularization for binary and multiclass problems for ELM is implemented using one against all (OAA) method. As in OAA method, j binary classifiers will be constructed in which all the training examples will be used at teach time of training. The training data having the original class label j_n = (1……n) have each j_n elements of positive one class and the remaining training data will be of zero class, creating j_n models implementing proposed RANSAC multi model response regularization on (j_n) binary classes. Finally, CS defined as $S^i(\theta)$ of j(n) classes is computed as

$$S^i(\theta)_j = \sum_{j=1}^{n} S^{(i)}\theta_j \cup S^{(i-1)}\theta_j \quad (21)$$

$$S^i(\theta) = \sum_{j=1}^{n} S^{(i)}\theta_j \cup S^{(i)}\theta_n \quad (22)$$

For this, consider the ELM for multi-class classification problem, formulated as k binary ELM classification problem with the following form:

$$Hw_1 = y_1 \ldots Hw_j = y_j;$$

Where for each j, w_j is the output weights from the hidden layer to the output layer with output vector $y_j = (y_{1j}, \ldots\ldots y_{mj})^t \in R_m$. Thus the output of the hidden layer as H hidden matrix defines with respect to multiclass binary classifiers as

$$H_j = \sum_{j=1}^{n} H * Y \begin{pmatrix} Y_j = 1 \\ o \end{pmatrix} \quad (23)$$

where H is the hidden layer output matrix and Y is the j binary classes with m^{th} observations of training data and n binary classes as columns vectors. Thus, we get H_j hidden matrix where each H_j belong to each binary class and RANSAC multi response regularization is implemented to acquire CS for each binary class as $S^i(\theta)_j$. It can be concluded that RANSAC multi response regularization for binary and multiclass problems work in similar fashion as OAA-ELM with j binary classes with a difference of j^{th} label with positive class and rest other classes with -1 class.

To improve better generalization and prediction accuracy, OLS doesn't provide a complete solution to remove irrelevant variables. OLS doesn't use sparse models to get related variables with respect to output.to resolve the mentioned issues with simple OLS, regularization factors or penalty approach added to minimization cost function in (Equation 20) to get sparse model of OLS to acquire sparse model which tries to shift most of the irrelevant variables to zero.

X = LSQR (A, B) attempts to solve the system of linear equations A*X=B for X if A is consistent, otherwise it attempts to solve the least squares solution X that minimizes norm (B-A*X). LSQR is an iterative method to find Ordinary Least Squares solution for large sparse matrix which is analytically equivalent to the standard equation of conjugate gradients, but possess more favourable numerical properties.

The experimental validation of the proposed algorithm on the benchmark datasets, and the extension to other datasets is discussed in next two Sections.

4. Experimental Results

For experimental validation of the proposed algorithm, the performance will be compared with the original ELM approach, the SVM approach, and other machine learning methods using public available benchmarks datasets for regression, binary and multiclass classification.

In the proposed algorithm, different activation functions were used as *G(a,b,x)* along with existing algorithms like ELM, MLP and SVM. The different activation functions of hidden layer is defined as:

1. Sigmoid Function

$$G(a,b,x) = \frac{1}{1 + \exp(-(a^t x+b))}$$

2. Radial Basis Function

$$G(a,b,x) = (\|x-a\|_2^2 + b^2)^{1/2}$$

For simulations of experimental results, the input weights and the hidden nodes are chosen randomly at the beginning of the iterations and were fixed in rest of the iterations. All experiments were conducted in MATLAB R2010 platform running on windows 7-64 bit operating system with 3.0 GHz Intel® core 2 i5 processor having 8 GB of RAM. LIBSVM is used for the implementation of SVM in matlab and Weka platform. OP-ELM toolbox is used for implementation of OP-ELM(Miche, Sorjamaa, & Lendasse, 2008). RANSAC toolbox is used for implementation and making proposed changes in sequential RANSAC(Zuliani, 2008)..

4.1 Datasets

For experimental validation of proposed algorithm, several benchmark datasets from UCI machine learning repository [2] were chosen, and are as shown in Table 1 and 2 here. The data sets have all been processed using 10 different random permutations taken with- out replacement; for each permutation, two thirds are taken for the training set, and the remaining third for the test set (see Table 1) by using cross validation function in MATLAB (crossvalind). Training sets are then normalized (zero-mean and unit variance) and test sets are also normalized using the very same normalization factors than for the corresponding training set. The 10 fold cross validation also enables to obtain an estimate of the standard deviation of the results presented (see Table 2). We use the software LIBSVM library for experiments. LIBSVM is a general library for support vector classification and regression, which is available at http://www.csie.ntu.edu.tw/~cjlin/libsvm . As mentioned above, that there are different functions to map data to higher dimensional spaces, practically we need to select the kernel function $K(x^i; x^j) = \emptyset(xi)^T \emptyset(x^j)$. There are several types of kernels used for solving different kinds of problems. Each kernel uses different parameters for solving these different problems. For example, some well-known problems with large amount of features, such as text classification, protein folding and image processing problems are reported to be classified more correctly with the linear kernel. In our study, we use the RBF kernel. A learner with the RBF kernel usually performs no worse than others do, in terms of the generalization ability. In this study, we did some simple comparisons and observed that using the RBF kernel the performance is a little better than the linear kernel $K(x^i; x^j) = \emptyset(xi)^T \emptyset(x^j)$ for all the problems we studied. Therefore, for the three data sets instead of staying in the original space a non-linear mapping to a higher dimensional space seems useful. Another important issue is the selection of parameters. For SVM training, few parameters such as the penalty parameter C and the kernel parameter of the RBF function must be determined in advance. Choosing optimal parameters for support vector machines is an important step in SVM design. We use the cross validation on different parameters for the model selection.

Table 1:- Classification datasets attributes and classes

Dataset	Attributes	Classes	Training data size	Testing data size
WDBC	30	2	427	142
Wincosin_BC	10	2	525	174
Cleveland	13	2	228	75
Australian Credit	14	2	518	172
Ionosphere	34	2	264	87
diabetes	8	2	576	192
Liver Disorders	6	2	259	86
Iris	4	3	113	37
Wine	13	3	134	44
Glass	9	6	161	53
Auto Vehicle	18	4	635	211
Page Blocks	10	5	4105	1368
Image Seg	19	7	1733	577
Satellite	36	6	4827	1608

Table 2. – Regression datasets attributes and classes

Dataset	Attributes	Training data size	Testing data size
Auto-MPG	8	294	98
Machine-CPU	7	157	52
Servo	5	126	41
Forest-Fires	13	388	129
Boston	14	380	126
Concrete-CS	9	773	257
Abalone	8	3133	1044
Wine-(white)	12	3674	1224
Wine-(Red)	12	1200	399
Parkinson	22	4407	1468
Kin-8	9	6144	2048
Demo	5	1536	512
Ailerons	40	5366	1788

4.2 Classification Results.

Table 3 depicts the comparative analysis of proposed RANSAC multi model response regularized ELM with support Vector Machine (SVM) and other ELM variants. For each dataset, training data is trained with higher number of hidden nodes so as to prune the network with optimal hidden nodes with better classification accuracy. From Table 3, it can be seen that for binary classification, using sigmoid function; most of the binary datasets shows the higher testing accuracy compared to SVM, ELM and OP-ELM. Using the RBF function (shown in Table 4, the proposed model does not result in better testing accuracy results compared to other algorithms. But in both cases of activation function, RANSAC multi model response regularized ELM prune the number of hidden nodes improving the optimality of the ELM network. The bold numbers in the Table show the high classification and regression accuracy compared to other algorithms considered for the study.

Table 5 depicts the comparative analysis of proposed algorithm with SVM and other variants for multi-class classification. As can be seen from Table 5 using sigmoid and RBF kernel,, RANSAC multi model response ELM shows the significant higher testing accuracy results compared to other algorithms for wine, glass, auto and segmentation datasets. Table 6 defines the number of hidden nodes pruned using RANSAC multi model response ELM with optimal higher testing accuracy. As can be seen from Table 6, for binary and multi-class datasets, the RANSAC multi model response ELM significantly prune the number of hidden nodes from higher number of hidden nodes maintaining the higher testing accuracy, faster implementation and better generalization performance for most of the binary and multiclass classification.

Table 3. Experimental results in terms of testing accuracy for binary classification using Sigmoid kernel

Datasets	HN	Testing(sigmoid)			
		SVM	OP-ELM	ELM	RANSAC-ELM`
WDBC	200	95.77	90.85	95.39	**95.81**
Win-BC	200	94.82	89.66	96.53	**96.97**
Cleveland	200	76.00	78.67	**90.40**	83.31
Aus-credit	200	83.72	84.88	81.40	**86.03**
Ionosphere	200	78.16	79.31	79.45	**88.92**
Diabetes	200	69.79	77.60	71.26	**76.21**
Liver Disorders	200	58.13	54.65	57.81	**65.88**

Table 4. Experimental results in terms of testing accuracy for binary classification using RBF kernel

Datasets	HN	Testing(RBF)			
		SVM	OP-ELM	ELM	RANSAC-ELM`
WDBC	200	**97.18**	90.14	83.20	93.20
Win-BC	200	96.55	**97.13**	90.74	95.78
Cleveland	200	76.00	78.61	**90.55**	81.25
Aus-credit	200	83.72	**86.63**	73.84	84.72
Ionosphere	200	89.65	**95.40**	78.41	92.90
Diabetes	200	75.00	**79.69**	69.20	77.30
Liver Disorders	200	**74.41**	68.60	57.47	67.30

Table 5. Experimental results in terms of testing accuracy for multiclass classification

Datasets	HN	Testing(sigmoid)			Testing(RBF)		
		SVM	ELM	RANSAC-ELM`	SVM	ELM	RANSAC-ELM
Iris	200	**91.89**	78.32	83.89	**97.29**	85.24	89.46
Wine	200	93.18	**92.77**	**95.45**	**97.72**	84.85	94.36
Glass	500	39.62	41.85	**51.81**	41.50	**50.60**	48.79
Auto	500	43.60	67.39	**79.68**	56.39	58.23	**79.94**
Page	500	91.30	**94.85**	92.96	**92.17**	89.47	91.43
Segment	500	83.88	94.55	**95.33**	87.17	94.67	**94.75**
Satellite	500	83.64	89.60	88.66	83.70	90.25	88.20

Table 6. Experimental results in terms of number of hidden nodes pruned by proposed method using sigmoid and radial basis functions for binary and multi-class classification

Dataset	ELM Hidden Nodes	RANSAC-ELM hidden nodes(sig)	RANSAC-ELM hidden nodes(RBF)
WDBC	200	40	48
Wincosin_Breast_cancer	200	34	30
Cleveland	200	52	53
Australian Credit	200	33	36
Ionosphere	200	90	92
diabetes	200	29	30
Liver Disorders	200	61	70
Iris	200	94	71
Wine	200	60	58
Glass	200	65	123
Auto Vehicle	500	90	118
Page Blocks	500	97	230
Image Segmentation	500	232	245
Satellite	500	250	252

4.3 Regression Results.

Table 7 depicts the training accracy for differnt regression problem datasets using ELM and proposed algorithm with respect to different kernel function. With both activation functions, for regression problems, RANSAC multi model response regularized ELM prune the number of hidden nodes improving the optimality of the ELM network and least RMSE compared to other algorithms.For regression problems, Table 8 and 9 depict the comparative analysis of proposed RANSAC multi model response regularized ELM

with ELM and other ELM variants using various kernels.. As can be seen from Table 8 and 9, RANSAC multi model response annular ELM shows the significant higher testing accuracy results compared to other algorithms on different datasets. Table 10 defines the number of pruned hidden nodes using RANSAC multi model response annular ELM with optimal higher testing accuracy.Figure 3 depicts the confusion matrix using ELM of auto dataset for multiclass classification. As we can see from the Figure 3, the true positive rate is higher for output class with respect to target class.

Table 7:- Experimental results in terms of training root mean square accuracy for regression using sigmoid kernel

		ELM	RANSAC-ELM	ELM	RANSAC-ELM
AutoMPG	200	**1.4316**	1.4339	**1.5571**	1.6019
CPU	200	72.1778	**48.2834**	75.2482	**34.5921**
Servo	200	1.7221e-23	4.5645e-05	7.0477e-28	**0.0172**
Forest	200	0.9071	1.2545	0.9449	1.2358
Boston	200	**2.7443**	3.0085	**8.6816**	3.6345
Concrete	200	20.3213	**14.9237**	40.3428	**16.3896**
Abalone	200	**3.8052**	3.8399	3.9294	**3.8636**
Wine-W	200	0.4576	**0.4440**	1.3107	**0.4724**
Wine-R	200	**0.2943**	0.3011	0.9455	**0.2865**
Parkinson	200	0.0012	**0.0011**	0.0061	**0.0011**
Kin-8	200	**0.0176**	0.0211	0.0337	**0.0225**
Demo	200	**1.1882**	1.1902	**1.1867**	1.2408
Ailerons	200	3.4750e-08	**2.3821e-08**	**1.8721e-07**	2.4917e-08

Table 8. - Experimental results in terms of testing root mean square accuracy for regression using Sigmoid kernel

Datasets	HN	Testing(sigmoid)		
		OP-ELM	ELM	RANSAC-ELM`
Auto-MPG	200	**11.7282**	91.2388	**11.9512**
Machine-CPU	200	2.0269e+04	6.2133e+08	**1.4293e+03**
Servo	200	0.5501	11.4059	**0.2861**
Forest-Fires	200	4.6637	5.5861	**1.9943**
Boston	200	29.8807	36.5606	**8.7404**
Concrete-CS	200	223.4046	58.5347	**28.3328**
Abalone	200	6.1798	10.6831	**4.1517**
Wine-(white)	200	0.5494	0.5251	**0.4637**
Wine-(Red)	200	0.4698	0.4738	**0.3627**
Parkinson	200	0.0025	0.0014	**0.0012**
Kin-8	200	0.0481	**0.0189**	0.0219
Demo	200	1.7995	54.3682	**1.4459**
Ailerons	200	0.7569	3.9234e-08	**2.5698e-08**

Table 9. - Experimental results in terms of testing root mean square accuracy for regression using RBF kernel

Datasets	HN	Testing(RBF)		
		OP-ELM	ELM	RANSAC-ELM`
Auto-MPG	200	41.4800	49.6352	**8.0027**
Machine-CPU	200	9.5348e+03	6.8199e+07	**1.4612e+03**
Servo	200	0.5550	2.1920	**0.3258**
Forest-Fires	200	2.4981	4.0079	**1.8157**
Boston	200	33.7665	60.3066	**10.3812**
Concrete-CS	200	227.5232	95.1963	**31.114**
Abalone	200	5.9829	4.9659	**3.9806**
Wine-(white)	200	0.6074	1.4766	**0.4808**
Wine-(Red)	200	0.5203	1.4369	**0.3837**
Parkinson	200	0.0031	0.0065	**0.0012**
Kin-8	200	0.0426	0.0361	**0.0224**
Demo	200	1.7991	2.5114	**1.3197**
Ailerons	200	0.8324	2.0346e-07	**2.6605e-08**

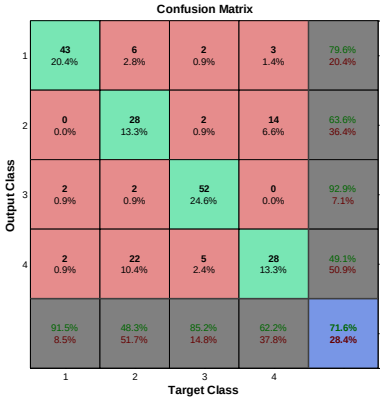


Figure 3:- Confusion Matrix using Extreme Learning Machine Algorithm for Auto Dataset [2] for Multiclass Classification. Figure 5:- Confusion Matrix depicting true positive and false positive rate using Extreme Learning Machine Algorithm of Auto Dataset for Multiclass Classification.

Table 10. - Experimental results in terms of number of hidden nodes pruned by proposed method using sigmoid and radial basis functions for regression

Dataset	ELM Hidden Nodes	RANSAC-ELM hidden nodes(sig)	RANSAC-ELM hidden nodes(RBF)
Auto-MPG	200	164	132
Machine-CPU	200	96	102
Servo	200	110	87
Forest-Fires	200	93	81
Boston	200	78	113
Concrete-CS	200	139	138
Abalone	200	118	97
Wine-(white)	200	96	79
Wine-(Red)	200	113	97
Parkinson	200	148	127
Kin-8	200	62	147
Demo	200	133	100
Ailerons	200	87	125

Thus, from above experiments, it can be stated that proposed method based on RANSAC regularization performs very well in terms of training and testing accuracy compared to traditional ELM and its variants. The above experiment also reduces the size of network

to optimal value, thus decreasing computation time and with faster performance of the network.

4.4 Results for the Hybrid Experiments

In this set of experiments, we have use SVM and the proposed hybrid classifier for increasing the performance and accuracy. Further, accuracy is measured in terms of percentage recognition ratio for this set of experiments. Suppose there is $N = n_1 + n_2 + n_3 \ldots\ldots\ldots\ldots\ldots + n_p$ test data, where in is the number of samples which belongs to the class i. suppose that c_i of sample from n_i are correctly recognized (as belonging to class i). So the total number of $C = c_1 + c_2 + c_3 \ldots\ldots + c_p$ classes is correctly recognized. Therefore the total accuracy is $Q = C/N$.

Table 11 describes the classification results using proposed ELM with SVM. In Table 11, we compared the recognition ratio using various kernels. As we can see in the Table both learning algorithms are processed with many hidden layers and their evaluations is done in terms of various factors. As depicted in Table 11, it clearly shows that proposed optimized ELM plays a major role in reducing the classification error. Table 12 defines the comparative study of multi kernels with SVM and proposed method respect to their learning classification accuracy rate.

Table 11:- Comparison among different methods

Method	**Recognition Ratio (%)**
SVM	91.32
H-KNN	84.38
Bayesian Naives	85.63
SVM	93.43
Optimized RANSAC-ELM	96.24

Table 12:- Classification through SVM using Various Kernels

Kernels	**SVM (%)**		**Proposed ELM (%)**	
	HAR	DAUB4	HAR	DAUB4
Polynominal	91.32	86.78	96.24	92.95
RBF	76.33	71.45	86.41	91.68
Linear	71.56	69.44	73.63	70.81

5. Conclusions

In this paper, we proposed an annular ELM based on RANSAC multi model response regularization to optimally prune the hidden nodes in a network and improve better generalization and classification accuracy. Experimental results were conducted using comparative analysis of proposed RANSAC multi model response regularization based annular ELM network on different benchmark datasets for binary and multiclass classification and regression problems.

It can be concluded that from experimental results that proposed RANSAC multi model response regularized based annular ELM works significantly better with higher classification accuracy with optimally pruned hidden units. The proposed algorithm implements faster compared to other algorithms in the study as it implements the ELM with less pruned hidden units without scarifying the higher generalization capability of ELM network. Further work will be conducted by testing the proposed algorithm in problems and datasets with images datasets such as bio-medical images, and images from videos.

6. References

1. Guang-Bin Huang, Qin-Yu Zhu, Chee-Kheong Siew, Extreme learning machine: Theory and applications, Neurocomputing, Volume 70, Issues 1–3, December 2006, Pages 489-501, ISSN 0925-2312, http://dx.doi.org/10.1016/j.neucom.2005.12.126.
2. Abid, S., Fnaiech, F., & Najim, M. (2001). A fast feedforward training algorithm using a modified form of the standard backpropagation algorithm. *Neural Networks, IEEE Transactions on*, *12*(2), 424-430. doi: 10.1109/72.914537
3. Blake, C., & Merz, C. J. (1998). UCI Repository of machine learning databases [http://www. ics. uci. edu/~ mlearn/MLRepository. html]. Irvine, CA: University of California. *Department of Information and Computer Science*, *55*.
4. Chum, O., Matas, J., & Kittler, J. (2003). Locally Optimized RANSAC. In B. Michaelis & G. Krell (Eds.), *Pattern Recognition* (Vol. 2781, pp. 236-243): Springer Berlin Heidelberg.
5. Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, *24*(6), 381-395. doi: 10.1145/358669.358692

6. Huang, G.-B., Wang, D., & Lan, Y. (2011). Extreme learning machines: a survey. *International Journal of Machine Learning and Cybernetics, 2*(2), 107-122. doi: 10.1007/s13042-011-0019-y
7. Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2006). Extreme learning machine: Theory and applications. *Neurocomputing, 70*(1–3), 489-501. doi: http://dx.doi.org/10.1016/j.neucom.2005.12.126
8. Lavneet Singh, G. (2012). A Novel Approach for protein Structure prediction Using Pattern Recognition and Extreme Machine Learning. *Proceedings of International Conference of Neuro Computing and Evolving Intelligence, NCEI*.
9. Martínez-Martínez, J. M., Escandell-Montero, P., Soria-Olivas, E., Martín-Guerrero, J. D., Magdalena-Benedito, R., & Gómez-Sanchis, J. (2011). Regularized extreme learning machine for regression problems. *Neurocomputing, 74*(17), 3716-3721. doi: http://dx.doi.org/10.1016/j.neucom.2011.06.013
10. Miche, Y., Sorjamaa, A., & Lendasse, A. (2008, 2008/01/01). OP-ELM: Theory, Experiments and a Toolbox. *Artificial Neural Networks - ICANN 2008*. 5163, from http://dx.doi.org/10.1007/978-3-540-87536-9_16
11. Nistér, D. (2005). Preemptive RANSAC for live structure and motion estimation. *Machine Vision and Applications, 16*(5), 321-329. doi: 10.1007/s00138-005-0006-y
12. Raguram, R., Frahm, J.-M., & Pollefeys, M. (2008). A Comparative Analysis of RANSAC Techniques Leading to Adaptive Real-Time Random Sample Consensus. In D. Forsyth, P. Torr & A. Zisserman (Eds.), *Computer Vision – ECCV 2008* (Vol. 5303, pp. 500-513): Springer Berlin Heidelberg.
13. Singh, L., & Chetty, G. (2012). Review of classification of brain abnormalities in magnetic resonance images using pattern recognition and machine learning. *Proceedings of International Conference of Neuro Computing and Evolving Intelligence, NCEI*.
14. Yoan, M., Sorjamaa, A., Bas, P., Simula, O., Jutten, C., & Lendasse, A. (2010). OP-ELM: Optimally Pruned Extreme Learning Machine. *Neural Networks, IEEE Transactions on, 21*(1), 158-162. doi: 10.1109/TNN.2009.2036259
15. Zuliani, M. (2008). RANSAC toolbox for Matlab.
16. H. J. Rong, Y. S. Ong, A. H. Tan, and Z. X. Zhu, "A fast pruned-extreme learning machine for classification problem," *Neurocomputing*, vol. 72, pp. 359–366, 2008.

Identifying Product Families Using Data Mining Techniques in Manufacturing Paradigm

Israt J. Chowdhury and Richi Nayak
School of Electrical Engineering and Computer Science
Queensland University of Technology
Brisbane, Australia
{israt.chowdhury, r.nayak}@qut.edu.au

Abstract

Identifying product families has been considered as an effective way to accommodate the increasing product varieties across the diverse market niches. In this paper, we propose a novel framework to identifying product families by using a similarity measure for a common product design data BOM (Bill of Materials) based on data mining techniques such as frequent mining and clustering. For calculating the similarity between BOMs, a novel Extended Augmented Adjacency Matrix (EAAM) representation is introduced that consists of information not only of the content and topology but also of the frequent structural dependency among the various parts of a product design. These EAAM representations of BOMs are compared to calculate the similarity between products and used as a clustering input to group the product families. When applied on a real-life manufacturing data, the proposed framework outperforms a current baseline that uses orthogonal Procrustes for grouping product families.

Keywords: Product families BOM, frequent mining, matrix representation, and clustering.

1 Introduction

Agile manufacturing has resulted in mass customization and product proliferation, which consequently increases the number of products and part variations extensively. Simultaneously the current business climate demands for moving a product quickly from concept-to-market by reducing the product development lead time. A key element of shortening this lead time is the ability to use existing knowledge and designs to generate new variations of existing products, which ensure a reduction in time-to-market (Utterback & Meyer, 1993). Therefore, the concept of grouping product families has been introduced. Besides leveraging product development cost, this grouping can offer multiple benefits including reduction in new product launching risks, improved ability to upgrade products, and enhanced flexibility and responsiveness of manufacturing processes (Sawhney, 1998). For example, if two products have 45 out of 50 parts common, design of the similar parts can be reused and positioned for assembly early so that the remaining five parts can be added to the assembly when an order for a specific assembly has arrived. Exploring similarity among products may lead to the redesign of some parts.

Nowadays, with the advent of cheap storage and fast computer, a huge amount of data is generated during product design and development in a manufacturing system. The ability beyond human is required to process this huge amount of complex data into useful knowledge such as common product family information. The identification of product families is a non-trivial task due to the volume and complexity of the available data. A well-known historical approach of grouping product families is Group Technology (GT) (Harhalakis, Kinsey & Minis, 1992; Marion, Rubinovich & Ham, 1986). However, the practical acceptance of GT has been limited in modern manufacturing (Romanowski & Nagi, 2002; Romanowski & Nagi, 2005), as it requires enormous effort to do groupings due to the involvement of manual intermittent steps for developing a "coding system" to summarize the key design and other attributes. Some efforts have been made towards automation (Iyer & Nagi, 1997), but acceptable performance is not reached yet, especially for situations where the sheer volume of data becomes overwhelming for both human and systems.

Data mining techniques have been specifically designed to deal with massive amount of data automatically (i.e. without human intervention) and to identify meaningful patterns and dependencies hidden behind the data. However, due to the complex nature of the data generated in product design domain, existing data mining algorithms require modifications. Although data mining algorithms have been specifically written to effectively analyse large datasets, the product design data often cannot be simply "plugged in" to these programs (Romanowski, Nagi & Sudit, 2006).

Bill of Materials (BOM) is a common product design data used in various domains like mechanical, electrical, electronic and civil/infrastructure. BOM is a hierarchical, structured representation of the products that details information such as parts descriptions, raw materials, quantities, manufacturing details, production times, etc. (Romanowski & Nagi, 2005). Researchers and practitioners have started using BOM specifications more commonly to represent their data (Matías, Garcia, Garcia & Idoipe, 2008). It has become essential to propose similarity measures for BOM data to determine similarity between product designs, which will eventually lead to find effective groupings of product families.

For BOM data, the critical information lays in the recursive parent-child relationships between the end item, its components or subassemblies, and the raw (or

purchased) materials they contain. This information can naturally be depicted in rooted labelled unordered tree format. In this paper we represent BOM data as unordered trees and introduce a novel matrix form called Extended Augmented Adjacency Matrix (EAAM) for equivalent tree representation. This representation facilitates search for similar designs and thus reduces the time consumption between concept and product launch. Our approach is to utilize the data mining techniques like frequent mining and clustering for ensuring efficient similarity calculation and reducing the search space for finding similar groups. Using frequent mining allows finding frequent structural dependencies like parent-child in a particular database, which gives the list of most occurred BOM parts or components relations. This information is then used with other content and topological information such as optimal part encoding, hierarchical position or level, and part quantity, in clustering. Using EAAM representations of BOM data, cosine similarity measure is used to generate a similarity matrix that becomes input to a clustering algorithm for identifying the product families.

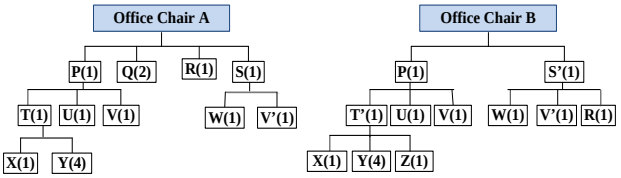
When applied on a real-life manufacturing data, the proposed framework including the BOMs similarity measure method has proven to excel in solving the problem of grouping product families automatically. The results are also compared with a current baseline that uses orthogonal Procrustes (Shih, 2011) for finding the product families and the proposed framework clearly outperforms.

Road map: In the following section, the related work is discussed. In section 3, the background knowledge is presented. In section 4, the proposed method for BOM similarity measure and the framework for identifying product families are given. The results are discussed in section 5. In section 6, the conclusion is drawn.

2 Related Works

Many efforts have been made for grouping the product families based on similarity schemes with emphasis on the different design areas and manufacturing. Most of them have focused on the historical approach of grouping individual parts into families, called as Group Technology (GT) (Harhalakis, Kinsey & Minis, 1992; Marion, Rubinovich & Ham, 1986)). The practical acceptance of GT has remained limited due to the expensive coding system development for summarizing the key product design and manufacturing attributes for doing the grouping. The main limitation of GT is the manual coding system. Though some efforts have been made towards automation, still more improvements are needed. Later, Authors in (Kao & Moon, 1991) used a back-propagation neural network based method for the product family grouping, but kept the existing GT classification and coding system. Another automated retrieval and ranking process for finding similar parts was proposed by authors in (Iyer & Nagi, 1997), but again based on GT coding. Authors in (Lee-Post, 2000) employed genetic algorithm to form the families, however, this approach also required to use the existing classification and coding scheme.

Instead of using information derived from a fixed GT code; some methods proposed similarity based on product functional features. Authors in (Chen, Chen, Wang & Chen, 2004) used the adaptive resonance theory (ART) neural network to develop a functional feature-based similarity method for grouping product families. Authors in (Liu, Yang, Bai & Tan, 2008) introduced another functional similarity-based combinatorial design method to produce a variety of products that satisfy various customer requirements in time. However, these functional feature-based schemes did not consider the hierarchical product design features. Authors in (Shih, 2011) attempted to calculate the similarity between BOMs considering the shape or geometrical structure, where a matrix representation and orthogonal Procrustes method were used to calculate the similarity score for grouping the product families. But BOMs are very flexible in shape, since there is no common rule or template to follow for generating them, therefore looking for geometrical or exact shape difference may give false similarity score. Emphasis should be put on the significant structural dependencies, hierarchical positions and other important contents during similarity calculation. The proposed framework in this paper focuses on the above for identifying the product families. To our best of knowledge, this is the first work on BOM data to determine product families using data mining.



Nodes meanings: P=Seat, Q=Elbow rest, R=Lumbar support, S and S'=Back variation, T and T'=Under frame variation, U=Seat frame, V and V'=Upholstery variation, W=Back frame, X =Standard, Y=Wheel, Z=Footrest

Figure 1: Variants of office chair

3 Background Knowledge

3.1 Bill of Materials (BOMs)

BOM represents hierarchical relations between various product parts with necessary details of manufacturing a particular product. It is a structural representation of a product including its required subassemblies, components and parts at various levels of production (Clement, Coldrick & Sari, 1992). To understand the proposed framework, following definitions need to be considered.

Definition 1 (*End Items*): The entities that are sold directly to the customer without any further value added manufacturing step. End items usually contain several subassembly parts and raw materials and appear at the top of the BOM hierarchal position.

Definition 2 (*Subassemblies*): These are the entities that cannot be sold to the customer. Subassemblies may contain manufactured or purchased part or other subassemblies, and therefore, are appeared at a level of BOM hierarchy which is positioned neither at the top nor at the bottom.

Definition 3 (*Purchased Parts*): The raw materials which are the initial entities for finishing a final product. Purchased parts are positioned at the bottom level of the BOM structure.

Definition 4 (*Quantity Representation*): In BOM, repeated subassemblies or parts are represented by a quantity per value. This value is the number of the part required per one unit of the part's parent.

Definition 5 (*Part Number*): This is an alphanumeric string that uniquely identifies an end item, subassembly and a purchased part. Each number corresponds to a specific item with specific characteristics.

Properties of BOM: BOM structures can be different for the identical end items, as each end item may be designed by a different company. Moreover, the product design is the result of human made input and developed completely based on individuals' understandings of how the product is manufactured or assembled. Similar BOMs may have different structures with same parts appearing at different level. However they will share similar components or parts and, most importantly, the structural dependencies among them will be usually kept same (Figure 1). BOMs substructures are unordered which means that the order of components is not significant. For instance, it does not matter if we say a chair has a seat, elbow rest and wheel, or a chair has a wheel, seat and elbow rest. In this paper we depicted BOM as rooted labelled unordered tree.

Definition 6 (*Unordered Tree*): A rooted labelled unordered tree has an identical root node and preserves only ancestor-descendant or parent-child relationships among nodes. There is no left-to-right order among the sibling nodes.

3.2 Data Mining Techniques Used

To satisfy the need of mass customization and agile manufacturing, we need to apply techniques that will extract implicit, previously unknown, potentially useful and understandable pattern from a large database (Fayyad, Piatetsky-Shapiro & Smyth, 1996), thus the product design and manufacturing system will have substantial improvement. Using data mining techniques in advance manufacturing is becoming popular (Choudhary, Harding & Tiwari, 2009). In the proposed framework, we have used frequent mining and clustering, two well-known data mining techniques for finding similarities between products and grouping them into families.

Frequent mining is used to extract interesting patterns from a database using a specified support (Chowdhury & Nayak, 2014a, 2014b). Support determines how often a pattern is applicable to or appears to a given data set. It represents the probability that a database instance contains that pattern.

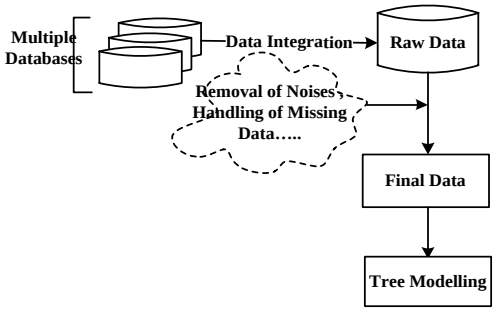


Figure 2: Data Pre-Processing Steps

BOM consists of structural dependencies like parent-child and ancestor-descendant relations between the end item, its components or subassemblies, and the raw (or purchased) materials they contain. The main challenge in BOM data analysis is dealing with the flexibility in its representation. It is very hard to put BOM data into a common format, thus the accurate analysis like similarity comparison can be carried out. Apparently, in BOM no other information keeps constant except the structural dependencies. So, instead of considering geometrical structure and shape, understanding structural dependencies is crucial for BOM similarity comparison. We utilise frequent mining to extract common structural dependencies in a database, which can be used as important representational component of the BOM data. These common structures can be input to clustering along with other information about the BOM data.

Clustering is an unsupervised data mining technique that can group objects based on their common characteristics, without the presence of any prior information about classification (Algergawy, Mesiti, Nayak & Saake, 2011; Kutty, Tran, Nayak & Li, 2008). Without using domain knowledge and GT coding based classification, the identification of product families can be possible using clustering. Clustering is now commonly used in manufacturing domain for doing unsupervised grouping (Ye & Gershenson, 2008). To apply clustering, a similarity measure value needs to be calculated based on commonality of the features. In this work, we utilise cosine similarity (Cha, 2007) to determine a similarity matrix based on the equivalent Extended Augmented Adjacency Matrix (EAAM) of a BOM dataset.

4 Proposed BOM Similarity Framework for Identifying Product Families

In this section a method of similarity measure between two BOM data instances is presented. A framework is then proposed integrating the similarity measure for identifying product families.

4.1 Data Pre-processing

To begin with our approach it is necessary to pre-process BOM data in order to make it useful for knowledge discovery. Figure 2 shows the tasks, which are used in this process.

4.1.1 Final Data

A company's database generally consists of a lot of data records. Only those records that correlate closely with the mining purpose are taken into account. Mostly BOM records are found in a tabular form, which typically contains the part name, part no, part revisions, part manufacturing description and the quantities required building a product assembly (as shown in Figure 3). Usually, the BOM input is given by human in spreadsheet, that can be formatted however one likes, but as anyone can format them, it often results in inconsistencies across a company's BOMs. Hence for mining BOM data, these inconsistencies need to be removed. Moreover not all of the information comprised by BOMs is necessarily mined for knowledge discovery. Therefore, once received the raw data through integration

of multiple databases, the final data sets should be identified involving such data cleaning and filtering tasks as removal of noises, handling of missing data files, etc.

Unordered Tree	BOM data
Root node	End item
Parent or ancestor node	End item and subassemblies
Child or descendant node	Subassemblies and purchased parts
Leaf node	Purchased parts
Parent-child or, ancestor-descendant relationships	End item-subassembly or, end item-purchased part or, subassembly-purchased part relationships
Node label	Part number

Table 1: Considered Mapping for BOM to Unordered Tree Representation

4.1.2 Unordered Tree Representation

After identifying final BOM data, tree modelling is done to support the EAAM construction. This modelling is carried out by using unordered tree structure scheme as template, where only parent-child and ancestor-descendent relationships are important. The BOM data can naturally be represented as unordered tree. By considering the parent-child and ancestor-descendant relationships between end item, subassemblies and purchased parts, a mapping can be derived.

Table 1 shows a general mapping that can be used to represent the BOM data as unordered tree. The end item, or finished product, can be considered a root of the tree; manufactured or assembled components can become the nodes; purchased parts or raw materials can be the leaf nodes. For example, in figure 3 the tabular or indented BOM of an ABC Lamps Product-LA01 (Fogarty, Blackstone & Hoffmann, 1991) is given, where the lamp is the end product, and the parts given under first column are different subassemblies and purchased parts. For constructing a tree from this BOM only the relationships among various parts are important, such as B100, S100 and A100 are the children of the end item; 1100, 1200, 1300, 1400 are the children of B100, representing descendants of the end item.

For node labelling, part numbers are used. If we compared two BOMs of product Lamp, using part numbers as labels, two BOMs would only match where the part numbers were exactly the same. For instance, suppose part S-14 is a shade with I.D. = 14” (inch). Part S-18 is a shade with I.D. = 18” (inch). These two shades would not be matched because of the unique part numbers. However, we are interested in finding BOMs of similar nature even if they do not share exact content and topology. For this reason, we replace the part numbers with general node labels derived from the part characteristics and types. In the case of these two parts, we would replace the unique part labels with a single label S for the class of shades.

4.2 Finding Frequent Structural Relationship

The objective of the proposed framework is to form the product families based on the existing product models (BOMs). Due to the vast flexibility in BOM data, characterizing structural relationships based on frequent occurrence is essential to include in the global similarity calculation as in some cases, frequent-infrequent decision are used as a scale to measure the importance of the structural relations (Chi, Muntz, Nijssen & Kok, 2004). We consider these relationships as a representational component for the BOM dataset. We explain next how these relationships are derived.

ABC Lamp Company, Indented Bill of Materials, Lamp LA01.

Part number	Description	Quantity for each assembly	Unit of measure
B100	Base assembly	1	Each
1100	Finished shaft	1	Each
2100	3/8" Steel tubing	26	Each
1200	7"-Diameter steel plate	1	Inches
1300	Hub	1	Each
1400	1/4–20 Screws	4	Each
S100	14" Black shade	1	Each
A100	Socket assembly	1	Each
1500	Steel holder	1	Each
1600	One-way socket	1	Each
1700	Wiring assembly	1	Each
2200	16-Gauge lamp cord	10	Feet
2300	Standard plug terminal	1	Each

Figure 3: ABC Lamps Product-LA01 (Fogarty, Blackstone, & Hoffmann, 1991)

4.2.1 Tree traversal

Prior to implement frequent subtree mining algorithm, an optimal traversal (Chowdhury & Nayak, 2013) algorithm is used to ensure unique identity or canonical form (Valiente, 2002) of each product model, which is in unordered tree form. Optimal traversal is included as it ensures optimality by providing unique encoding within minimum computation time (Chowdhury & Nayak, 2013).

4.2.2 Frequent Mining Algorithm

Once the canonical form is built, the frequent mining can now be applied that permits not only to explore the relationships and dependencies but also to handle a huge amount of data in an optimal way (Chowdhury & Nayak, 2014a, 2014b). However, such algorithms are sometimes limited to the memory because of its size and calculations that they perform. The candidate frequent subtrees generation can be exponential in large databases (Chi et al., 2004).

We propose to apply the BOSTER algorithm (Chowdhury & Nayak, 2014b) which allows setting the subtree length equal to 1 and retrieves only single relationships exhibiting between parent-parts. This algorithm has proved to be memory efficient and exhibits limited computational complexity (Chowdhury & Nayak, 2014b). A support threshold is needed for frequent subtree mining process. A minimum support is set by trial and error, as it is a data specific parameter that prunes the infrequent subtree.

4.2.3 Characterizing Structural Relationships

Based on the result of the frequent subtree mining algorithm the structural relationships are characterized. If a subtree is frequent then the inherent parent child relation is considered as mandatory. Once all mandatory parent-child relationships are identified, the remaining parent-child relationships are classified as optional. During the EAAM representation a weighted value of 1

and 0 are used to represent the mandatory and optional relationship respectively.

4.3 Extended Augmented Adjacency Matrix (EAAM) Representation

In this paper a new matrix representation called EAAM is introduced. Although, EAAM is an extension of Augmented Adjacency Matrix (AAM) representation (Chowdhury & Nayak, 2013), but to our best knowledge, this is the first matrix, where the frequent structural relationship is included as one of the representational components. The rest of the components are:

- Optimal part sequence of BOM using optimal traversal.
- Part level information from BOM interface.
- Quantity representation (q) representing the number of the part required per unit of the part's parent.

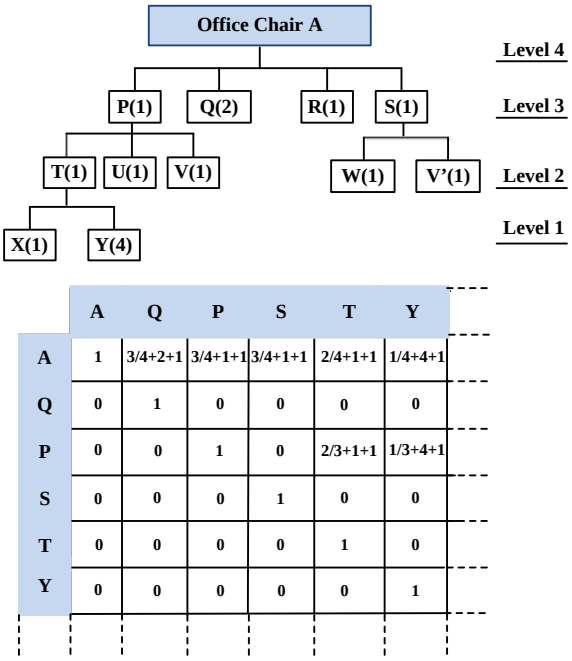
An adjacency matrix of a tree is based on the ordering chosen for the nodes (Rosen, 2011). For EAAM the ordering is achieved using optimal traversal (Chowdhury & Nayak, 2013) which ensures unique encoding of BOM represented in unordered tree form. For populating the cell of EAAM mainly structural relationship importance weight, level information and quantity representation are used.

Let a BOM, B is depicted as a rooted labelled unordered tree $B = (I, R)$, where $I = \{i_0, i_1, i_2, \ldots, i_n\}$ denotes the set of items with i_0 as end item, and other set elements as subassembly and purchased items, $R = \{(i_1, i_2)|i_1, i_2 \in I\} = \{r_1, r_2, \ldots, r_{n-1}\}$. The number of each item is given as $\{q_0, q_1, q_2, \ldots, q_n\}$. For B, the EAAM representation can be formulated in which a cell, a_{cd} is populated as follows:

$$a_{cd} = \begin{cases} 1 & \text{if } i_c \text{ is a node of } B \\ \dfrac{L(B,i_d)}{L(B,i_c)} + q_d + 1 & \text{if } i_c \text{ is an ancestor or parent of } i_d \text{ and the relation is frequent} \\ \dfrac{L(B,i_d)}{L(B,i_c)} + q_d + 0 & \text{if } i_c \text{ is an ancestor or parent of } i_d \text{ and the relation is not frequent} \\ 0 & \text{otherwise} \end{cases}$$

These four components are explained as follows:

1. To represent the presence of each part in a BOM, each diagonal cell is populated with 1.
2. If the part is parent or ancestor of the other respective part, and the parent-part relation is frequent then the cell is populated with level information (fraction of level of corresponding two parts), quantity representation of the child or descendant node and the mandatory structural relationship weight value equals 1.
3. If the part is parent or ancestor of the other respective part, and the parent-part relation is not frequent then the cell is populated with level information (fraction of level of corresponding two parts), quantity representation of the child or descendant node and the optional structural relationship weight value equals 0.
4. If none of these are true, then the cell receives a value of 0.



	A	Q	P	S	T	Y
A	1	3/4+2+1	3/4+1+1	3/4+1+1	2/4+1+1	1/4+4+1
Q	0	1	0	0	0	0
P	0	0	1	0	2/3+1+1	1/3+4+1
S	0	0	0	1	0	0
T	0	0	0	0	1	0
Y	0	0	0	0	0	1

Figure 5: EAAM Construction

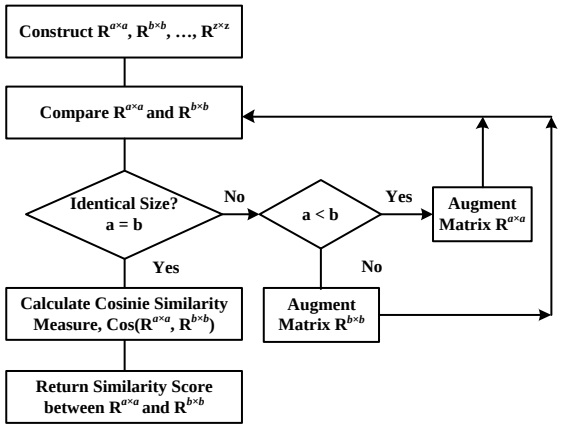


Figure 6: The Flow Chart of Calculating Similarity

Example: From figure 1, we consider the first example BOM of product model "office chair *A*" to explain the EAAM construction. Consider a BOM database that only consists of two BOM trees given in figure 1, and the minimum support is two. It means that if a subtree appears twice or more in the database, it will be considered as a frequent sub-tree. Based on this, *A-Q*, *A-R*, *A-Z*, *T-Z* and *S-R* are found infrequent relationships and considered as optional. The order of the nodes for constructing EAAM is derived using optimal traversal. Consider the cell between nodes *A* and *Q*. For this BOM tree, *A* is the parent of part *Q*, therefore the level information is added as 3/4, where the level of *A* is 4 and the level of *Q* is 3. For the child part *Q*, the quantity representation value is 2, which is added after the fraction of level into that cell. Finally, the frequent parent-part relation adds a value 1 to indicate the mandatory relationship. The overall calculated value for this cell is 3/4+2+1. The rest of the cell values are calculated following the same way.

4.4 BOM Similarity Measure

After constructing EAAMs, we use cosine similarity for matrix comparison for measuring the similarities between a BOM pair (Chowdhury & Nayak, 2013) as follows:

$$\cos(\mathrm{A},\mathrm{B}) = \frac{\sum_{x=1}^{n}\sum_{y=1}^{n} \mathrm{A}_{xy}\,\mathrm{B}_{xy}}{\sqrt{\sum_{x=1}^{n}\sum_{y=1}^{n} \mathrm{A}_{xy}^{2}}\sqrt{\sum_{x=1}^{n}\sum_{y=1}^{n} \mathrm{B}_{xy}^{2}}}$$

Where, A and B are two (n×n) matrices.

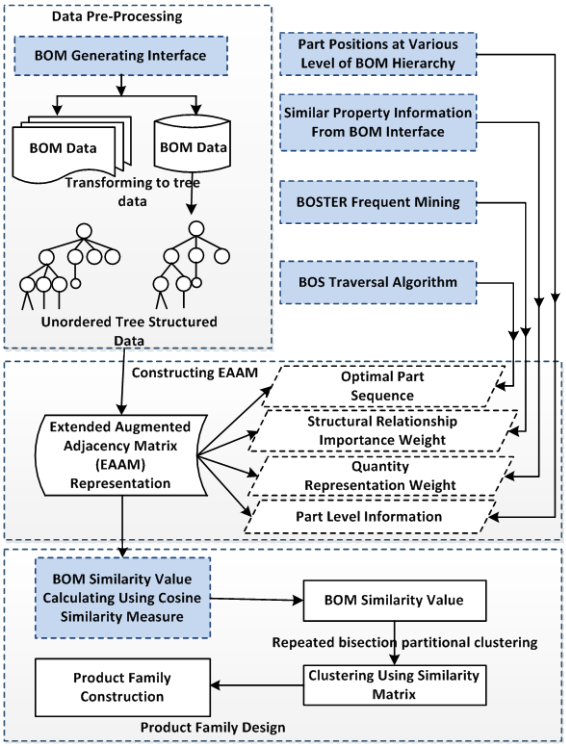


Figure 7: Framework for Product Family Design

If sizes of the two BOM trees are not same, then additional columns and rows with zero elements are padded to the smaller matrix for making the size of both matrices equal, this is called the augmentation of matrix. These two square matrices can be considered as two $|B|\times|B|$ (where $|B|$ = max $\{B1, B2\}$; B_1, B_1 are two BOM trees) dimensional vectors. The overall procedure for similarity measure is given in figure 6 using a flow chart, where matrix is represented as $\mathbf{R}^{a\times a}$, where a is the size of that matrix representing the number of the components or parts in a BOM tree.

4.5 The Proposed Framework

The proposed framework for grouping product families has three main phases as shown in figure 7. In the first phase data pre-processing is done. BOM has different storage under different enterprises; some of them store BOM data in database, some in files like XLS file. Some enterprises use part table/relationship table to express BOM, and some enterprises use a single table. All these variations need to save in memory as a BOM generating interphase, from this node the pre-processing will carry out in next.

Next phase covers the EAAM construction where all necessary steps (dotted blue boxes) are implemented for populating the feature weights.

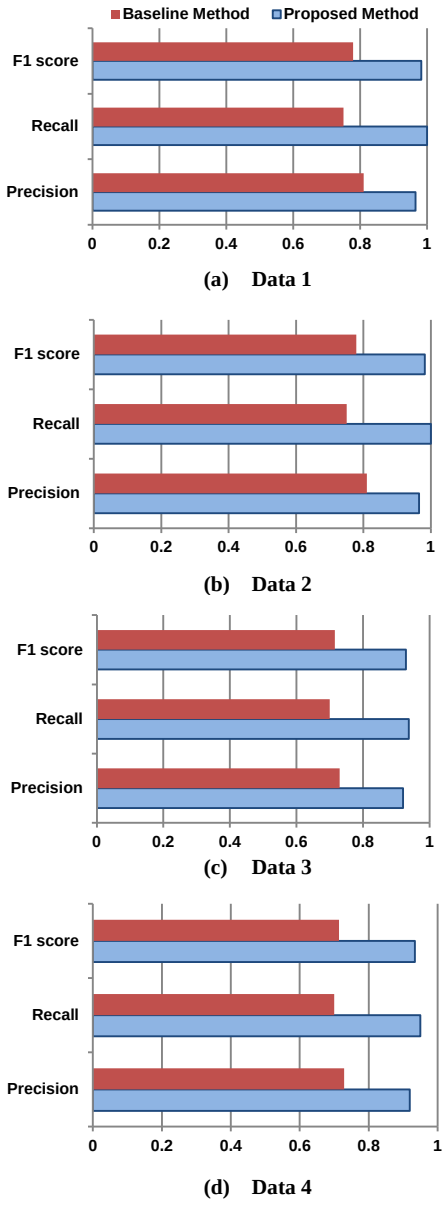


Figure 8: Accuracy Performance over Data 1(a), Data 2(b), Data 3(c) and Data4(c)

In the third and final phase, the pairwise similarity is calculated using the EAAM comparison and a similarity score is calculated between BOM pairs where a similarity score of 0 means completely dissimilar and a score of 1 means exactly similar. Using this similarity values a similarity matrix is constructed which is then employed as an input to a clustering algorithm. Table 2 shows an example of the similarity matrix. We used a well-known clustering algorithm, Repeated Bisection Partitioning (Rasmussen & Karypis, 2004), for grouping the BOMs into families. This algorithm divides trees into two groups and then selects one of the larger groups according to a clustering criterion function and bisects further. This process is repeated until the desired number of clusters is achieved. During each step of bisection, the cluster is bisected so that the resulting 2-way clustering solution locally optimizes a particular criterion function. Other clustering algorithms can also be applied. Finally from the cluster result, the product families will be identified.

	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15
B1	1.00	0.40	0.43	0.57	1.00	0.50	0.61	1.00	1.00	0.64	0.41	0.30	0.41	0.58	0.44
B2	0.40	1.00	0.43	0.47	0.40	0.49	0.37	0.40	0.40	0.42	0.32	0.43	0.32	0.54	0.39
B3	0.43	0.43	1.00	0.65	0.43	0.53	0.43	0.43	0.43	0.45	0.39	0.44	0.39	0.52	0.33
B4	0.57	0.47	0.65	1.00	0.57	0.70	0.60	0.57	0.57	0.71	0.50	0.35	0.50	0.63	0.34
B5	1.00	0.40	0.43	0.57	1.00	0.50	0.61	1.00	1.00	0.64	0.41	0.30	0.41	0.58	0.44
B6	0.50	0.49	0.53	0.70	0.50	1.00	0.62	0.50	0.50	0.71	0.65	0.34	0.65	0.71	0.35
B7	0.61	0.37	0.43	0.60	0.61	0.62	1.00	0.61	0.61	0.58	0.56	0.33	0.56	0.58	0.41
B8	1.00	0.40	0.43	0.57	1.00	0.50	0.61	1.00	1.00	0.64	0.41	0.30	0.41	0.58	0.44
B9	1.00	0.40	0.43	0.57	1.00	0.50	0.61	1.00	1.00	0.64	0.41	0.30	0.41	0.58	0.44
B10	0.64	0.42	0.45	0.71	0.64	0.71	0.58	0.64	0.64	1.00	0.56	0.31	0.56	0.72	0.39
B11	0.41	0.32	0.39	0.50	0.41	0.65	0.56	0.41	0.41	0.56	1.00	0.25	1.00	0.47	0.31
B12	0.30	0.43	0.44	0.35	0.30	0.34	0.33	0.30	0.30	0.31	0.25	1.00	0.25	0.42	0.31
B13	0.41	0.32	0.39	0.50	0.41	0.65	0.56	0.41	0.41	0.56	1.00	0.25	1.00	0.47	0.31
B14	0.58	0.54	0.52	0.63	0.58	0.71	0.58	0.58	0.58	0.72	0.47	0.42	0.47	1.00	0.31
B15	0.44	0.39	0.33	0.34	0.44	0.35	0.41	0.44	0.44	0.39	0.31	0.31	0.31	0.31	1.00

Table 2: BOM Similarity Matrix

5 Evaluation of the Proposed Framework

We implemented the proposed framework on a real manufacturing data to evaluate the performance.

This data is collected from a manufacturer of nurse calling devices (Romanowski & Nagi, 2004). It consists of 404 BOMs with four major product families. From this data set we randomly generated four samples, consisting 100 BOMs each and named them as Data 1, Data 2, Data 3 and Data 4. We used all these four datasets for empirical analysis.

For benchmarking we consider a method that used the orthogonal Procrustes problem to find the orthogonal matrix for two given matrices that will closely map one matrix to another and used this as a geometrical similarities between BOMs and then clustered them into families (Shih, 2011). For the benchmark method we used the same clustering algorithm, but we used the orthogonal Procrustes based similarity measure as input and performed the product grouping. Finally we checked the clustering results with the known product family information and compared the performances.

The main contribution of this paper is the similarity measure method of product BOMs. An efficient grouping of product families largely depends on an efficient similarity measure method. We evaluated our similarity measure approach using the well-known evaluation metrics including precision, recall and F1 score (Goutte & Gaussier, 2005) and performed on all four data samples. For these metrics, the value close to 1 is considered as an indication of better performance. From figure 8, we can see for all four data sets our proposed similarity measure method gives higher accuracy in comparison to the benchmark method. This good accuracy performance should also reflect during the clustering process, as we used this similarity method as an input for an off-the-self clustering algorithm for doing the product family grouping. Table 2 gives a partial view of the similarity matrix generated by our proposed BOM similarity measure method. For clustering we used this similarity matrix for identifying product families.

Table 3 reports the clustering performance results, where we mainly included the number of mis-clustered product BOM for each data by the proposed method and the benchmarked method. The proposed framework outperforms the baseline method.

Method	Data 1	Data 2	Data 3	Data 4
Proposed Framework	2	5	5	6
Baseline Method	19	21	25	35

Table 3: Number of Mis-Clustered BOMs for Different Data Sets

6 Conclusion

A product family is a group of related products based on a product platform, facilitating mass customization by cost-effectively providing a variety of products for different market segments. In this paper we present a data mining approach based framework for grouping various products into families. We introduced a similarity measure method for a common product data type, BOM that can be used to cluster products into families. The benchmarking results confirm the efficiency of the proposed work.

In future work, we intend to expand the study on unifying the families into a single Generic Bill of Material (GBOM) (Hegge & Wortmann, 1991) group.

7 References

Algergawy, A., Mesiti, M., Nayak, R., & Saake, G. (2011). XML data clustering: An overview. *ACM Computing Surveys (CSUR)*, **43**(4): 25.

Cha, S.-H. (2007). Comprehensive survey on distance/similarity measures between probability density functions. *International Journal of Mathematical Models and Methods in Applied Sciences,* **1**(4): 300-307.

Chen, Y., Chen, Y., Wang, C., & Chen, T. (2004). Using artificial neural networks to develop a mechanism for functional feature-based reference design retrieval.

Proc. IEEE International Conference on Engineering Management **2**: 829-833. IEEE.

Chi, Y., Muntz, R. R., Nijssen, S., & Kok, J. N. (2004). Frequent Subtree Mining - An Overview. *Fundamental Informatic.*, **66**(1-2): 161-198.

Choudhary, A. K., Harding, J. A., & Tiwari, M. K. (2009). Data mining in manufacturing: a review based on the kind of knowledge. *Journal of Intelligent Manufacturing*, **20**(5): 501-521.

Chowdhury, I. J., & Nayak, R. (2013). A Novel Method for Finding Similarities between Unordered Trees Using Matrix Data Model. *Proc. WISE 14th International Conference on Web Information Systems Engineering*, **8180**: 421-430. Springer Berlin Heidelberg.

Chowdhury, I. J., & Nayak, R. (2014a). BEST: An Efficient Algorithm for Mining Frequent Unordered Embedded Subtrees (*In Press*) *Proc. PRICAI 13th Pacific Rim International Conference on Artificial Intelligence*, Gold Coast, Australia, **8862** Springer Berlin Heidelberg.

Chowdhury, I. J., & Nayak, R. (2014b). BOSTER: An Efficient Algorithm for Mining Frequent Unordered Induced Subtrees. *Proc. WISE 15th International Conference on Web Information Systems Engineering*, Athens, Greece, **8786**: 146-155. Springer Berlin Heidelberg.

Clement, J., Coldrick, A., & Sari, J. (1992). *Manufacturing Data Structures; Building Foundations for Excellence with Bills of Materials and.*. New York, John Wiley & Sons, Inc.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery: an overview. In *Advances in Knowledge Discovery and Data Mining*. 1-34 Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P. And Uthurusamy, R. (eds). AAAI Press/MIT Press.

Fogarty, D. W., Blackstone, J. H., & Hoffmann, T. R. (1991). *Production & Inventory Management*, South-Western Publishing Company.

Goutte, C., & Gaussier, E. (2005). A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation. In *Advances in Information Retrieval*. Vol. 3408, 345-359 Losada, D., Fernández-Luna, JuanM (eds). Springer Berlin Heidelberg.

Harhalakis, G., Kinsey, A., & Minis, I. (1992). Automated group technology code generation using PDES. *Proc. Third International Conference on Computer Integrated Manufacturing*, 4-14. IEEE.

Hegge, H. M. H., & Wortmann, J. C. (1991). Generic bill-of-material: a new product model. *International Journal of Production Economics*, **23**(1–3): 117-128.

Iyer, S., & Nagi, R. (1997). Automated retrieval and ranking of similar parts in agile manufacturing. *IIE Transactions*, **29**(10): 859-876.

Kao, Y., & Moon, Y. (1991). A unified group technology implementation using the backpropagation learning rule of neural networks. *Computers & Industrial Engineering*, **20**(4): 425-437.

Kutty, S., Tran, T., Nayak, R., & Li, Y. (2008). Clustering XML documents using closed frequent subtrees: A structural similarity approach. In *Focused Access to XML Documents*. 183-194(eds). Springer.

Lee-Post, A. (2000). Part family identification using a simple genetic algorithm. *International Journal of Production Research*, **38**(4): 793-810.

Liu, F., Yang, B., Bai, Z., & Tan, R. (2008). Research on product combinatorial design based on functional similarity. *International Journal of Design Engineering (IJDE)*, **1**(3): 333–356.

Marion, D., Rubinovich, J., & Ham, I. (1986). Developing a group technology coding and classification scheme. *Industrial Engineering*, **18**(7): 90-97.

Matías, J., Garcia, H. P., Garcia, J. P., & Idoipe, A. V. (2008). Automatic generation of a bill of materials based on attribute patterns with variant specifications in a customer-oriented environment. *Journal of Materials Processing Technology*, **199**(1): 431-436.

Rasmussen, M., & Karypis, G. (2004). gcluto: An interactive clustering, visualization, and analysis system, *UMN-CS TR-04* (Vol. 21).

Romanowski, C. J., & Nagi, R. (2004). A data mining approach to forming generic bills of materials in support of variant design activities. *Journal of Computing and Information Science in Engineering*, **4**(4): 316-328.

Romanowski, C. J., & Nagi, R. (2005). On comparing bills of materials: a similarity/distance measure for unordered trees. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, **35**(2): 249-260.

Romanowski, C. J., Nagi, R., & Sudit, M. (2006). Data mining in an engineering design environment: OR applications from graph matching. *Computers & Operations Research*, **33**(11): 3150-3160.

Rosen, K. (2011). *Discrete Mathematics and Its Applications 7th edition*, McGraw-Hill Science.

Shih, H. M. (2011). Product structure (BOM)-based product similarity measures using orthogonal procrustes approach. *Computers & Industrial Engineering*, **61**(3): 608-628.

Valiente (2002). *Algorithms on Trees and Graphs*. New York, Springer, Berlin Heidelberg.

Ye, X., & Gershenson, J. K. (2008). Attribute-based clustering methodology for product family design. *Journal of Engineering Design*, **19**(6): 571-586.

Evolving Wavelet Neural Networks for Breast Cancer Classification

Maryam Mahsal Khan **Stephan K. Chalup** **Alexandre Mendes**

School of Electrical Engineering and Computer Science
The University of Newcastle,
Callaghan NSW 2308, Australia
Email: MaryamMahsal.Khan@uon.edu.au
Email: Stephan.Chalup@newcastle.edu.au
Email: Alexandre.Mendes@newcastle.edu.au

Abstract

Digital Mammograms are x-ray images of the breast and one of the preferred early detection methods for breast cancer. However, mammograms are still difficult to interpret, and associated with this problem is a high percentage of unnecessary biopsies, misdiagnoses and late detections.

The focus of this research is to use neuroevolutionary mechanisms for detecting breast cancer from mammographic images. The aim is to design a sophisticated classification tool that detects breast cancer at its early stages, so that treatment has a better chance of success.

Wavelet neural networks have the ability to capture and extract information at various frequency levels by using different dilation and scaling values of the wavelet function. In this work, the wavelet neural network parameters are evolved using on the concept of Cartesian Genetic Programming, resulting in an evolved neural network which is trained for mass diagnosis.

In the reported study the proposed algorithm achieves a classification accuracy of 89.57% on a real dataset composed of 200 images. Such a computer-based classification system has the potential to provide a second opinion to the radiologists, thus assisting them to diagnose the malignancy of breast cancer more precisely.

Keywords: Breast Cancer, Mammography, Cartesian Genetic Programming, Evolution, Neural Networks, Wavelet Neural Networks, Neuroevolution

1 Introduction

Breast cancer is the second leading cause of cancer-related deaths in Australian women, accounting for 15.5% of them. It is estimated that one in eight Australian women will be diagnosed with the disease before the age of 85 (*Breast Cancer Care WA; accessed June* 2014).

Digital mammograms are digital x-ray images of the breast; and regularly used for cancer screening. It is one of the earliest and most reliable detection methods, with cancer being indicated by the presence of a microcalcification - calcium deposit within the breast tissue or masses. The identification and assessment of potentially cancerous areas is a tedious, time-consuming and challenging task, which requires specialised expertise. Such assessments might also lead to misdiagnosis, which is why Computer Aided Detection (CAD) systems provide a valuable second opinion for the classification of suspicious areas as cancerous (malignant) or non-cancerous (benign).

Artificial Neural Networks (ANN) are a computational model represented by simulated neurons (called units) with weighted connections between them. There are a number of evolutionary algorithms devised in the past decade with different strategies of evolving either connection weights, network topology, or both (this last case known as TWEANN – Topology and Weight Evolving Artificial Neural Networks). An important example of such evolving networks is the NEAT (Neuroevolution of Augmented Topology), proposed by (Stanley & Miikkulainen 2002). The algorithm has the capability to evolve both structures and weights depending on the complexity of the problem, and is not dependent on a predefined network structure. Also the recently proposed neuroevolutionary algorithm namely ANN evolved via Cartesian Genetic Programming (CGPANN) has also been applied on different domains of engineering with success (Khan et al. 2013).

Standard classifiers like Support Vector Machines and Multilayer Perceptron, despite their success in many domains, display some limitation on varying complex tasks, e.g. intelligent control, language learning, etc (Byun & Lee 2002, Muhlenbein 1990). Wavelets - referred to as a 'microscope' in mathematics (Cao et al. 1995), act as high compression nodes that represent non-linearities effectively (Fang & Chow 2006). Wavelet neural network has been applied on a variety of problems with great success, e.g. time-series analysis and prediction (Cao et al. 1995, Chen et al. 2006), signal de-noising (Zhang 2007), classification and compression (Kadambe & Srinivasan 2006, Subasi et al. 2005), density estimation (Hasiewicz 1997), non-linear modelling (Billings & Wei 2005), etc. Cartesian Genetic programming has been used particularly in digital circuit optimization(Miller & Thomson 2000). In this research the wavelet neural network parameters are evolved using the Cartesian Genetic Programming so that the evolved wavelet neural network benefits from the strength of wavelets and overcome the limitations of standard classifiers; thus possibly contributing to classification, prediction and control problems.

Our research focuses on the development of a novel neuroevolutionary algorithm not only to classify mass abnormalities identified in digital mammograms; assessing its potential to be a part of a high-confidence CAD system for diagnosis, but also to exploit it further in the intelligent control domain. For this reason the potential of the algorithm is first tested on a well-

researched and a reasonably challenging dataset, that has been used by many researchers (McLeod & Verma 2013*b*, Zhang et al. 2010, Verma et al. 2009*a*). The dataset is also tested on two existing neuroevolutionary algorithms, namely ANN evolved via Cartesian Genetic Programming and Neuroevolution in Augmented Topology.

The remainder of the paper is divided into five sections. Section 2 describes Cartesian Genetic Programming (CGP) and the proposed algorithm - Wavelet Neural Networks evolved via CGP. Section 3 highlights the dataset for cancer detection and the literature review surrounding that database. Section 4 describes the methodology, followed by analysis and discussion of the results. Finally, Section 6 concludes the paper.

2 Background

2.1 Cartesian Genetic Programming

Cartesian Genetic Programming (CGP) (Miller & Thomson 2000) is an evolutionary programming technique used particularly for digital circuits optimisation. CGP genotypes are of finite length and have an integer representation, where genes represent nodes and each node corresponds to sets of input genes and a function. The input genes can be a program input, or outputs of other nodes. Functions are either logical or arithmetical e.g. AND, OR, addition, subtraction, etc. The output of the genotype is either a node output or the program input itself.

There are two basic types of evolution strategies (μ, λ)-ES and $(\mu + \lambda)$-ES (Beyer & Schwefel 2002). μ represents the number of parent population and λ refers to the number of offspring produced in a generation. In (μ, λ) offspring replaces the parent as the fittest is selected from λ, while in $(\mu + \lambda)$-ES the fittest is selected from both parents and offspring for the next generation. Cartesian Genetic programming uses the $(1 + \lambda)$ strategy, where $\lambda = 4$, is commonly adopted. i.e. a single parent is mutated based on a mutation rate 'τ' to produce 4 offsprings.

Figure 1(a) is an example of a finite length genotype with two inputs (x_0, x_1) and one output, where the encircled number represents the output. It represents a 2×2 architecture i.e. it has 2 rows and 2 columns. The number of inputs to each node is 2. The functions used are the logical OR and AND gates, displayed as f_0 and f_1, respectively. Figure 1(b) is the graphical representation of the genotype. The graph represents active and inactive nodes. Inactive nodes are nodes that do not participate in the output computational process (shown in light grey). Based on the output x_5, the phenotype of the corresponding genotype is shown in Figure 1(c).

2.2 Cartesian Genetic Programming Wavelet Neural Network (CGPWNN)

Wavelet Neural Networks have three layers, namely input, hidden and output layers. The input layer represents the input to the network. The hidden layer consists of wavelet neurons ψ, known as wavelons, with scaling α and translation β parameters, as shown in Figure 2(a). Therefore, the input presented to the wavelons are scaled and translated, which transforms the input pattern. The output layer approximates, or sums the input coming from the hidden layer. Each output from the hidden layer is multiplied by a weight w_i, where i corresponds to the wavelet neuron index. There are a number of wavelet functions ψ that can be

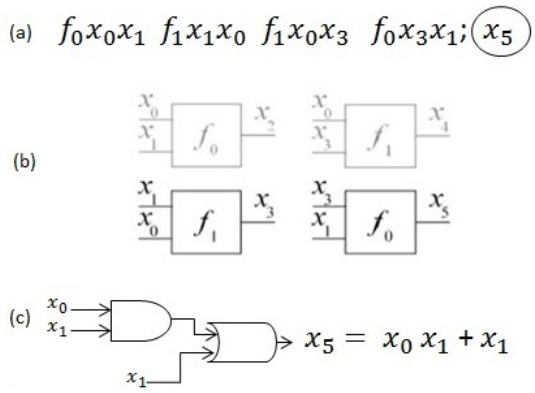


Figure 1: (a) An example of a 2×2 CGP genotype with 2 inputs and 1 output. (b) Graphical representation of the genotype in (a). (c) Phenotype corresponding to the genotype in (a).

used, e.g. Gaussian derivative, Mexican hat, Morelet, Haar, etc. The choice of wavelet used in the application depends on the system itself. The structure of a single hidden layer wavelet network is displayed in Figure 3. The output of the network is mathematically expressed in Eq. (1):

$$y(x) = \theta + \sum_{i=1}^{m} w_i \psi_i(x) + \sum_{j=1}^{n} c_j x_j \qquad (1)$$

where θ is the bias to the output neuron and $c_j x_j$ represents the direct connection of input to the output representing a linear model (Alexandridis & Zapranis 2013).

The CGP representation has been used to evolve artificial neural networks previously (Khan et al. 2013). Similarly, in the current paper we will use the CGP representation, but to evolve wavelet neural networks. A node in CGP corresponds to a wavelet neuron in CGPWNN. Figure 2(b) shows a wavelon of CGPWNN. The genes that make up a wavelon include: input x_{ij}, connection c_{ij}, motherwavelet ψ, translation β and scale α where x_{ij} = [1, number of program inputs], c_{ij} = {0, 1}, ψ = [1, number of wavelet functions], β = [0, 1] and α = [0, 1] respectively. The input and connection genes occur in pairs, i.e. if the input to a wavelon is 2, then it would constitute two inputs and two connection genes.

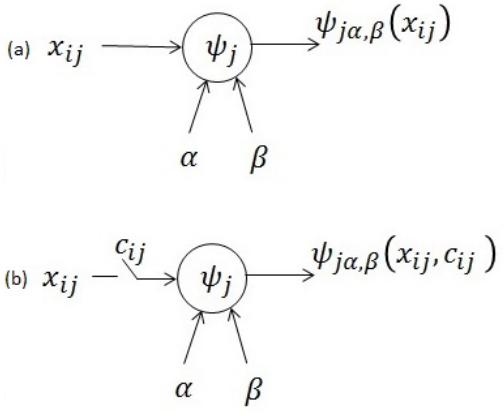


Figure 2: (a) Diagram of a Standard Wavelet Neuron. (b) CGPWNN-Wavelet Neuron.

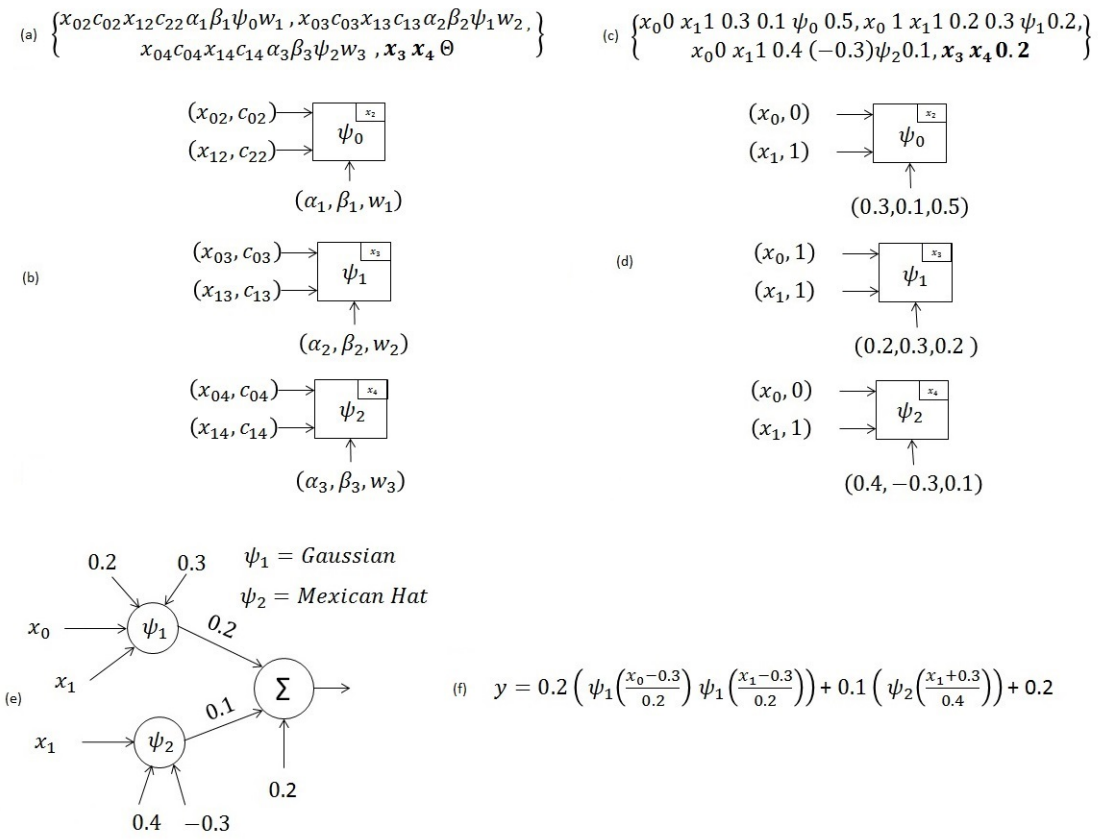


Figure 4: (a) A 3×1 example genotype of CGPWNN. (b) Graphical representation of the genotype. (c) Random values assigned to (a). (d) Graphical representation of the genotype with assigned random values from (c). (e) Phenotype of (d). (f) Mathematical representation of phenotype.

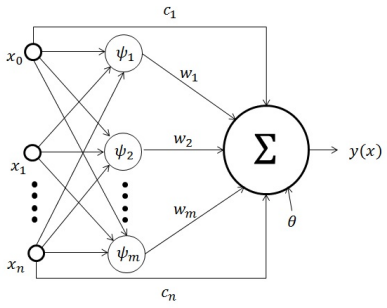


Figure 3: Structure of a Wavelet Neural Network.

Figure 4(a) is an example of a 3×1 CGPWNN genotype with two inputs: x_0 and x_1. The number of inputs to each wavelet neuron is 2. The number of outputs from the network is also 2, i.e. x_3 and x_4. The wavelet functions used are Gaussian, Mexican hat and Haar, labelled as ψ_1, ψ_2 and ψ_3, respectively. The bias gene is incorporated at the end of the genotype, and marked as θ. Figure 4(b) is the graphical representation of such genotype. The number at the top right corner of each wavelet neuron corresponds to the node index. Suppose that we assign random values to the genes, as shown in Figure 4(c). In that case, its graphical representation is displayed in Figure 4(d). The phenotype of the assigned genotype is Figure 4(e), and can be mathematically expressed as in Figure 4(f).

3 Case study: Mass classification

Classifying suspicious areas in digital mammograms is a crucial, difficult problem, and one of the significant processes for the early detection of breast cancer. In the current paper we are investigating one of the challenging and publically available benchmark datasets for breast cancer diagnosis using evolutionary neural networks. In this section we shall discuss in detail the features used for mass classification and the associated literature survey.

3.1 Database and features utilized

The Digital Database for Screening Mammography (DDSM), from the University of South Florida, is an online repository of mammographic images collected from different hospitals, with different resolutions (*Digital Database for Screening Mammography; accessed May* 2014, Heath et al. 1998, 2001). The suspicious regions are manually marked on the film by two experienced radiologists. These regions are represented as chain codes which can be easily extracted from the image file for further analysis.

A total of 25 features are extracted from the regions marked on the mammographic images scanned on the HOWTEK scanner, at 43.5 micron per pixel spatial resolution (Kumar, Zhang, & Verma 2006). The feature set includes 18 grey level features, based on the grey level pixel values of suspicious areas using the statistical formulas shown in Table 1 (where T = total number of pixels; g = index value of image I; k = number of grey levels (4096); $I(g)$ = grey level of pixel g in image I; and $P(g)$ is the proba-

bility of the grey level g occurring in image I. Also, each case in DDSM contains information specified by an expert radiologist using BIRADS (Breast Imaging Reporting and Data System) lexicon. The four BI-RADS attributes are *density*, *mass shape*, *mass margin* and *assessment*. *Patient age* and *subtlety* values are also extracted from each mammographic record, while *calcification association* is added by (Kumar, Zhang & Verma 2006). Those 7 features are human interpreted. The list of features along with their descriptions is shown in Table 1.

3.2 Literature Survey

There are many research papers surrounding breast cancer diagnosis and classification. The following literature focuses on research that used the dataset features reported above, for the purpose of comparing results. In 2005, (Zhang, Kumar & Verma 2005) proposed a hybrid classifier that used statistical classifiers (Logistic Regression (LR) and Discriminant Analysis (DA)) output probabilities as second order features, combined in a feature set with other 14 grey level and 6 human extracted features. The modified feature set was then tested on several classifiers, including neural networks, and genetic neural networks, with 3 random splits of the dataset. A maximum accuracy of 91% for the LRDA-GNN classifier was obtained. Furthermore, in their papers, (Zhang & Kumar 2006) statistically analyzed the various features using SPSS, and 4 key features (assessment, age, margin and shape) were identified. They were then used in conjunction with neural networks and decision trees (CART) (Breiman et al. 1984, Steinberg & Colla 1997, Steinberg & Golovnya 2006) and C5.0 (Quinlan 1993, RuleQuest-Research 2014) for classification purposes. Accuracy was higher in comparison to using the whole feature subset, which meant that feature extraction improved the performance of the classification. Also, they proposed that using Logistic Regression alone on the 7 human extracted features attains high classification accuracy, with an AUC (area under curve) of 0.979 (Zhang et al. 2010).

(Panchal & Verma 2006) exploited different feature subsets in terms of its classification accuracy using auto associative and classifier neural networks. A total of 14 grey level, 4 BIRADS, plus patient age and subtlety features were selected, and subsequently divided into six feature subsets. The main objective behind the research was to identify key features in breast cancer detection. The study determined that grey level and BIRADS features perform better, with a training accuracy of 100% and testing accuracy of 92%. Training and testing was done using a 50/50 data split.

(Kumar, Zhang, & Verma 2006) used decision trees (CART and C5.0) at different cost ratios for mass classification on the whole feature set. Data was also split 50/50 for training and testing. Results showed a maximum of 91% accuracy, for a cost ratio of 1:1 using CART; at the cost of a higher standard deviation.

Verma's series of works introduced a number of algorithms for classifying the mass dataset (Verma 2008, Verma et al. 2009*b*,*a*). In one of those, an additional neuron in the hidden layer was proposed (based on the number of classes), improving both the memorization and generalization ability of the network. A different training mechanism for the additional neurons was also devised. Based on that approach, the training and testing accuracy improved to 100% and 94% respectively, where the classifier was trained and tested on a 50/50 data split using 6 of the human extracted features. Also, Verma introduced two soft cluster-based neural networks, where the clusters were formed within a neural network layer i.e. SCBDL (soft cluster based direct learning) & SCNN (soft cluster neural network). By using 10-fold cross validation with both algorithms, a maximum of 94% in SCNN and 95% in SCBDL accuracy was achieved. A comparison with other clustering algorithms (SVM, K-means and SOM) showed accuracies of 86.5%, 84.5% and 76%, respectively.

In 2011, (McLeod & Verma 2011) proposed a multi-cluster support vector machine for mass classification. The K-means algorithm was used to generate the clusters for benign and malignant classes. The resulting clusters were then used for classification on a standard SVM. The MCSVM obtained an average accuracy of 94.5%, while standard SVMs reached 87.5% using 10-fold cross validation and 6 human extracted features. McLeod also observed an approximate increase of 3% in accuracy using the same cluster based approach on other classifiers, namely radial basis function networks and multilayer perceptrons (McLeod & Verma 2010). The accuracy of the classifiers were improved further by using neural network ensemble classifiers in (McLeod & Verma 2012*a*,*b*, 2013*a*,*b*). The networks in the ensemble varied the number of neurons in the hidden layer, between 2 to 150 neurons. The maximum number of classifiers in the ensemble was limited to 40 in (McLeod & Verma 2012*b*), and 202 in (McLeod & Verma 2013*a*). A 10-fold cross validation was used for testing the methods. The final ensemble network was composed of 127 classifiers, which attained an accuracy of 99%. It is noteworthy that in order to classify 200 data rows a total of 127 classifiers was needed. Similar improvement was observed in an LCA ensemble (94%), compared to LCA (87%) alone in (Pour et al. 2012).

3.3 Training and testing sets

A total of 200 suspicious areas were manually extracted from the Digital Database for Screening Mammography dataset. Half of those areas represented benign tumours and the other half was malignant (Zhang, Kumar & Verma 2005).

3.3.1 Training on 70% of the data

The first part of the experiment involved training and testing the classifiers by splitting the dataset into 70% and 30%, respectively, with equal contribution of benign and malignant samples to each group. Similar to Verma's and McLeod's studies, 6 human-interpreted features were used in all of our experiments (breast density, mass shape, mass margin, assessment, subtlety and patient age).

3.3.2 10-fold cross validation

The second part of the simulation incorporated a 10-fold cross validation strategy for testing the classifiers. In that case, the dataset is divided into 10 subsets, where 9 subsets are combined into a training set, and the remaining subset is used as the test set. This is repeated 10 times, always with a different selection of subset for testing, and the average accuracy is reported.

In this work, the classifier's output is thresholded, based on a threshold value of $\theta = 0$, to classify the samples as benign (0) or malignant (1). That is mathematically expressed by Eq. (2).

Table 1: Features and description of the Digital Database for the Screening Mammography (DDSM) dataset. (Zhang, Verma & Kumar 2005)

Features	**Description**
	Grey Level Features
Minimum Grey Level	Minimum grey level in the suspicious area
Maximum Grey Level	Maximum grey level in the suspicious area
Perimeter of Suspicious Area	Count of pixels at the boundary of the extracted area
Mean Boundary Grey Level	BAG = Average Grey Level at the boundaries
Number of Pixels	Count of pixels in extracted area
Mean Histogram	$AHg = (1/k)\sum_{j=0}^{k-1} N(j)/T$
Energy	$Egy = \sum_{g=0}^{k-1}[P(g)]^2$
Entropy	$Etp = -\sum_{g=0}^{k-1} P(g)log_2[P(g)]$
Standard Deviation	$\sigma = \sqrt{\sum_{g=0}^{T-1}(g-AG)^2P(g)}$
Skew	$Skw = (1/(\sigma_j)^3)\sum_{g=0}^{k-1}(g-AG)^3P(g)$
Modified Energy	$MEgy = \sum_{g=0}^{T-1}[P(I(g))]^2$
Modified Entropy	$MEtp = -\sum_{g=0}^{T-1} P(g)log_2[P(I(g))]$
Modified Standard Deviation	$m\sigma = \sqrt{\sum_{g=0}^{T-1}(I(g)-AG)^2P(I(g))}$
Modified Skew	$MSkw = (1/\sigma_j^3)\sum_{g=0}^{T-1}(I(g)-AG)^3P(I(g))$
Kurtosis	$Kur = (1/(\sigma_j)^4)\sum_{g=0}^{k-1}(g-AG)^4P(g)$
Mean Grey Level	$AG = 1/T\sum_{g=0}^{T-1} I(g)$
Difference	$Dff = AG - BAG$
Contrast	$Ctr = Dff/(AG+BAG)$
	Human Interpreted Features - BIRADS
Breast Density	Density of breast tissue; rated 1-4
Abnormality Assessment Rank	Seriousness of abnormality; rated 1-5
Mass Shape	Morphological descriptor, e.g. round, oval, lobulated, irregular etc.; rated 1-9
Mass Margin	Morphological descriptor, e.g. circumscribed, microlobulated, obscured etc.; rated 1-5
	Human Interpreted Features - Others
Subtlety	Subjective abnormality measure; rated 1-5
Patient Age	Age of patient at the time of mammography
Calcification Association	Relation of mass to calcification; categorized as yes or no

$$ClassifierOutput = \begin{cases} 0, \text{if } Output \geq \theta \\ 1, \text{if } Output < \theta \end{cases} \quad (2)$$

3.3.3 Performance measures

The performance of the classifiers is evaluated based on the following metrics:

1. **Training Accuracy** (Tr_{Acc}): fraction of correctly trained samples.
2. **Testing Accuracy** (Te_{Acc}): fraction of correctly classified samples as expressed in Eq. (3), also known as the classification accuracy. The higher the percentage, the better is the classifier performance.

$$Te_{Acc} = \frac{(TP+TN)}{P+N} \quad (3)$$

where TP represents true positive cases, i.e. accurate classification of benign samples; TN represents true negative cases, i.e. accurate classification of malignant samples; and $(P+N)$ is the total number of positive and negative test samples.

3. **Sensitivity** ($Sens$): measurement of the fraction of true positive cases, mathematically represented in Eq. (4):

$$Sens = \frac{TP}{(TP+FN)} \quad (4)$$

where FN is the number of false negatives – *Type 2* error – where the classification of a malignant case as benign is a severe mistake.

4. **Specificity** ($Spec$) Statistical measurement of the fraction of true negative cases mathematically represented as in Eq.(5):

Table 2: Performance of CGPANN with 70/30 split between training and testing samples. The best configuration is with $[1 \times 100]$, $I_E = 3$ and $O_p = 4$, indicated in bold. That configuration was then tested using 10-fold cross-validation (bottom row).

Configuration			Accuracy %				Active Parameters	
Structure	I_E	O_p	Tr_{Acc}	$Te_{Acc}(\sigma)$	Sens	Spec	Neurons(%)	Features
1×50	3	2	91.85	87.05(2.87)	83.31	91.73	18.13	4.53
		4	92.83	87.83(3.28)	85.06	91.07	21.60	4.57
	6	2	92.67	87.67(2.56)	85.02	90.74	32.33	5.70
		4	**93.57**	87.72(3.63)	87.10	88.36	32.87	5.70
1×100	3	4	92.59	**89.11**(2.84)	88.45	92.00	18.83	4.87
		8	93.14	88.22(3.46)	87.63	88.82	25.90	5.23
	6	4	92.90	87.61(2.90)	85.51	89.96	28.03	5.97
		8	92.52	87.73(3.58)	86.28	89.30	43.63	6.00
Best configuration - 10-fold cross-validation								
1×100	3	4	92.57	87.15(5.24)	86.10	88.91	16.92	5.01

Table 3: Performance of CGPWNN & CGPWNN with linearity disabled on 70% Training and 30% Testing Dataset. The best configuration indicated in bold was then tested using 10-fold cross-validation.

Configuration			Accuracy %				Active Parameters	
Structure	I_E	O_p	Tr_{Acc}	$Te_{Acc}(\sigma)$	Sens	Spec	Wavelons(%)	Features
CGPWNN								
50×1	3	4	94.02	89.16(2.86)	84.00	96.19	7.93	4.33
		8	93.47	87.67(3.32)	82.65	94.48	15.87	4.97
		12	94.07	87.94(2.94)	83.57	93.61	23.33	5.67
	6	4	93.02	88.27(3.48)	83.87	94.00	7.93	4.83
		8	92.07	87.27(5.73)	82.16	94.31	15.60	5.87
		12	93.78	88.61(3.16)	85.71	92.01	23.93	6.00
100×1	3	4	93.76	89.50(3.17)	84.61	95.98	3.97	4.10
		8	93.45	88.27(3.23)	83.60	94.45	7.90	5.17
		12	93.95	88.16(2.76)	83.64	94.09	11.87	5.60
	6	4	93.80	**89.57**(2.85)	84.90	95.64	4.00	5.00
		8	**94.14**	88.11(3.06)	83.96	93.41	8.00	5.97
		12	93.97	88.61(1.82)	84.10	94.49	12.00	6.00
Best configuration - 10-fold cross-validation								
100×1	6	4	92.99	88.60(4.83)	86.84	91.14	3.99	4.94
CGPWNN with linearity disabled								
50×1	3	4	93.59	88.22(2.94)	84.60	92.67	8.00	3.93
		8	94.30	**89.57**(3.64)	85.38	94.38	16.00	4.97
		12	94.28	88.83(2.89)	85.41	92.98	24.00	5.60
	6	4	93.47	87.89(2.68)	84.23	92.41	8.00	4.80
		8	93.61	89.11(2.30)	85.12	94.11	16.00	5.70
		12	93.83	88.89(3.25)	85.35	93.20	24.00	5.97
100×1	3	4	93.78	89.22(2.60)	85.22	94.23	4.00	3.87
		8	**94.59**	88.27(3.23)	84.69	92.68	8.00	5.23
		12	94.26	89.33(2.41)	85.40	94.25	12.00	5.60
	6	4	93.59	88.67(2.63)	83.72	95.31	4.00	4.73
		8	93.97	88.72(2.18)	84.53	94.05	8.00	5.87
		12	93.52	88.67(2.83)	84.31	94.27	12.00	5.97
Best configuration - 10-fold cross-validation								
50×1	3	8	94.09	88.03(5.36)	86.70	89.92	16.00	5.08

$$Spec = \frac{TN}{(TN + FP)} \tag{5}$$

where FP is the number of false positives – *Type 1* error – corresponding to the classification of a benign sample as malignant.

4 Experimental setup

4.1 CGPANN parameters

The first part of the experiment involves the evolution of genotypes under two random architectures $[1 \times 50]$ and $[1 \times 100]$, where rows = 1 and columns = 50 and 100, respectively, and with different parameter settings. The intent of having a single row is to have

a fully connected feedforward network - a standard CGP configuration. The number of inputs to each neuron I_E were 3 and 6 and the number of outputs O_p were 2, 4 and 8, respectively. A $(1+9)$-ES, with $\lambda = 9$, and a mutation rate of 0.1% was used in all of the simulations, similar to (Khan et al. 2013). Each network was evolved for 50,000 generations. The activation functions used were sigmoid and hyperbolic tangent. Table 2 shows the different configuration of parameters for the network and their performance, using a 70/30 split between training and testing samples. The table shows the figures for training accuracy, testing accuracy along with the standard deviation of the accuracy of the 30 genotypes, sensitivity, specificity, active neurons and the number of selected features.

The results are averaged over 30 independent evolutionary runs. The best result was with $[1 \times 100]$, $I_E = 3$ and $O_p = 4$, with $Tr_{Acc} = 92.59$ and $Te_{Acc} = 89.11$. That configuration then proceeded for further testing, now using 10-fold cross-validation. Results indicate a training accuracy of 92.57% and a testing accuracy of 87.15%, using the average for 30 independent runs of the cross-validation. Sensitivity and specificity were 86.10% and 88.91%, respectively.

4.2 CGPWNN parameters

Similarly to CGPANN, two random CGPWNN architectures $[50 \times 1]$ and $[100 \times 1]$ were used. The number of columns was set to 1, as the number of hidden layers in a wavelet neural network is also 1. The number of inputs to each wavelon I_E was set at 3 and 6; and the number of outputs O_p were 4, 8 and 12. As in the previous case, a $(1+9)$-ES, with $\lambda = 9$, and a mutation rate of 0.1% was used in all of the simulations. Each network evolved for 50,000 generations. Wavelet functions used in the experiments were Gaussian, Mexican hat and Haar wavelets. The networks were trained on 70% of the data and tested on the remaining 30%, as before. The performance for each network configuration is shown in Table 3 and results represent the average of 30 independent evolutionary runs. Similar parameters were also used to train a modified version of CGPWNN - disabling direct connection of input to the output. The intent was to know whether input features modeled non-linearly would perform better; results are shown in the same table.

4.3 NEAT parameters

The main parameters used in NEAT for evolving neural network structure is shown in Table 4. The NEAT classifier was trained under both 70% training, 30% testing; and the 10-fold cross validation strategies. Table 5 shows the result of each training and testing set which is the average of 30 independent evolutionary runs.

5 Results and discussion

In Tables 2 and 3, there is no observable trend for accuracy as the networks' structure vary. Maximum training and testing accuracies achieved by CGPANN (from Table 2) are 93.57% and 89.11%. The maximum training accuracy of CGPWNN in Table 3 is 94.14% with a 100×1 structure with 6 inputs and 8 outputs. Analogously, the maximum testing accuracy was 89.57% with a network structure of 100×1 with 6 inputs and 4 outputs. By disabling the direct input connectivity to the output (see Table 3),

Table 4: NEAT algorithm parameters

Attribute	Value
Population size	150
Speciation (c1, c2, c3)	(1, 1, 0.4)
Crossover percentage	0.8
Mutation probability: Add node	0.03
Mutation probability: Add connection	0.05
Mutation probability: Recurrency	0.0
Mutation probability: Mutate weight	0.9

the maximum training accuracy was 94.59% with a 100×1 network with 3 inputs and 8 outputs; and testing accuracy remained same i.e. 89.57% with a 50×1 structure with 3 inputs and 8 outputs. These results implies that the feature set can be modeled either way.

As mentioned in Section 4, the network with the maximum accuracy is used to train the data samples using 10-fold cross validation. Since 10-fold cross validation is a considerably more robust test strategy compared to training/testing split, we observed a relatively small performance decrease. CGPANN attained average testing accuracy of 87.15% (Table 2) while CGPWNN and its modified version (Table 3) obtained an average testing accuracy of 88.60% and 88.03%, respectively.

From Table 5, NEAT obtained training and testing accuracies of 90.59% and 89.11%, respectively. Both CGPANN and NEAT achieved the same accuracies, but the sensitivity of CGPANN was found to be 88.45% while that of NEAT was 86.67%. The 10-fold cross validation again resulted in a small decrease in the performance of the NEAT classifier to 84.63%. A similar reduction in classification accuracy was also observed in (McLeod & Verma 2011, Verma et al. 2009*a*).

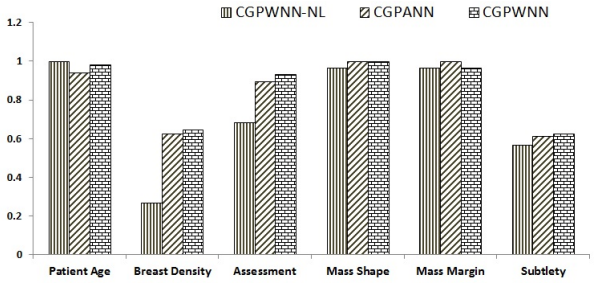


Figure 5: Feature selection of the evolved neural networks, where patient age, mass margin and mass shape were selected by the three algorithms all the time.

Figure 5 shows a histogram of the features selected using the networks (CGPWNN, CGPWNN-NL, CGPANN) with the best testing accuracy. The results are averaged over 30 independent evolutionary runs

Table 5: Performance of NEAT using 70/30 split and 10-fold cross validation strategies.

	Accuracy %			
Strategy	Tr_{Acc}	$Te_{Acc}(\sigma)$	$Sens$	$Spec$
70/30 split	90.59	89.11(3.02)	86.67	92.48
10-fold cross valid.	91.14	84.63(4.50)	85.55	85.93

Table 6: Performance of neuroevolutionary algorithms in terms of number of evaluations and computational time

Algorithm	Average number of evaluations	CPU time (in hours)
CGPANN	171,090	2.96
CGPWNN	179,670	1.35
CGPWNN-NL	154,160	1.30

and shown as percentages. In CGPANN and CGPWNN most of the genotypes selected four features: patient age, assessment, mass margin and mass shape – similar to (Zhang & Kumar 2006); breast density and subtlety were selected less often (approximately 60% of the time). Finally, CGPWNN - NL selected 3 features every time: patient age, mass margin and mass shape. Thus, one can argue that those are the most robust features from the features list, as they are selected by all methods all the time.

In Table 2, it is also observed that the maximum number of active neurons in the search space of 1×50 and 1×100 is 43.63%. Even though more than 57% of the genotype represents inactive genes or junk nodes, it has been shown that the presence of inactive genes actually is useful to the efficiency of the evolutionary process, due to the concept of neutrality (Miller & Smith 2006, Vassilev & Miller 2000, Yu & Miller 2001, 2002). Active nodes are only involved in the computational process which implies lesser delays. By increasing the number of outputs, the contribution of neurons from the pool of resources increases (Table 2).

In CGPANN, the computational process cannot be controlled via any genes (input, connection, weight, function, output, etc). On the other hand, in CGPWNN, since the architecture is forcibly $[n \times 1]$, the number of outputs ultimately controls the range of active wavelons in the search space. From Table 3, with increasing numbers of output nodes, the active wavelons in the search space increases, restricted to an upper limit of the total number of outputs; thus, forcing the evolutionary process to search for optimum solutions under a controlled computational environment.

A (1+9)-ES (used in our simulations) implies an evaluation of 10 genotypes in each generation. Table 6 shows the average number of evaluations for 30 independent evolutionary runs of CGPANN, CGPWNN and CGPWNN-NL genotypes with the maximum classification accuracy along with the average CPU time (in hours) to complete 50,000 generations. CGPWNN-NL has the lowest number of evaluations, at 154,160, as compared to CGPANN (171,090) and CGPWNN (179,670) and therefore, is the fastest learning algorithm among the three. For the average CPU time, the platform was a single core CPU @3.40GHz with Windows-XP 64-bit. We can clearly see that CGPANN took the longest time (2.96 hrs) in evaluating genotypes, while CGPWNN and CGPWNN-NL took 1.35 hrs and 1.30 hrs, respectively. That was somewhat expected, as in CGPWNN computational delay is controlled via outputs; hence the time required to complete the same number of generations is shorter.

Even though CGPWNN requires less CPU time compared to CGPANN, it still produces better and equivalent accuracy results. The strength of CGPWNN is the wavelet functions and the CGP representation itself. Such wavelet function modifies input in a manner that has an equivalent effect of multiple neurons together.

Table 7 shows the performance of different classifiers in classifying the breast mass dataset. We can see that the proposed classifiers have outperformed most of the so-called *standard classifiers*, i.e. those not based on clustering or ensemble architectures.

6 Conclusions

This work compared existing (CGPANN, NEAT) and novel neuroevolutionary algorithms (CGPWNN) for their ability to classify malignant and benign patterns in digital mammograms.

CGPWNN achieved a classification accuracy of 89.57%, while CGPANN and NEAT reached 89.11%, respectively. Three features were consistently selected during the evolutionary process. *Patient age*, *mass margin* and *mass shape* thus play an important role in the correct classification of tumours. The CGPWNN algorithm was also found to be less computationally expensive and managed to search for higher quality solutions faster, compared to the other approaches.

The proposed technique was also found to perform comparatively to other techniques mentioned in literature and outperformed most of the standard classification algorithms.

Currently, a predefined structure for the search space is provided to the algorithm. In our future research, we intend to investigate a developmental form of CGPWNN that would evolve with each time-step. The performance of CGPWNN shall also be exploited further by introducing an adaptive threshold gene and the effect of having no bias. In addition, we plan to test the algorithm on other biomedical benchmark case studies.

7 Acknowledgments

We would like to acknowledge Dr. Ping Zhang, Research Scientist at CSIRO, for sharing the dataset features for research purposes.

References

Alexandridis, A. & Zapranis, A. (2013), 'Wavelet neural networks: A practical guide', *Neural Networks* **42**, 1–27.

Beyer, H. & Schwefel, H. (2002), 'Evolution strategies: A comprehensive introduction', *Natural Computing* **1(1)**, 3–52.

Billings, S. A. & Wei, H. (2005), 'A new class of wavelet networks for nonlinear system identification', *IEEE Transactions on Neural Networks* **16(4)**, 862–874.

Breast Cancer Care WA; accessed June (2014), `http://www.breastcancer.org.au/`.

Breiman, L., Friedman, J. H., Olshen, R. A. & Stone, C. J. (1984), *Classification and Regression Trees*, Pacific Grove:Wadsworth, Belmont, CA.

Byun, H. & Lee, S.-W. (2002), Applications of support vector machines for pattern recognition: A survey, *in* 'Proceedings of the First International Workshop on Pattern Recognition with Support Vector Machines', SVM '02, Springer-Verlag, London, UK, pp. 213–236.

Table 7: Comparison of different classifiers on the DDSM using 6 features

Algorithm	Accuracy(%)	Sensitivity	Specificity	References
Standard classifiers				
LCA	87.00	80.50	93.90	(Pour et al. 2012)
AANN	91.00	90.00	92.00	(Panchal & Verma 2006)
SVM	87.50	88.40	91.60	(McLeod & Verma 2011)
K-means	84.50	–	–	(Verma et al. 2009*b*)
SOM	76.00	–	–	(Verma et al. 2009*b*)
NN	90.00	91.60	88.40	(Pour et al. 2012)
CART	91.00	–	–	(Kumar, Zhang, & Verma 2006)
C5.0	89.00	–	–	(Kumar, Zhang, & Verma 2006)
GANN	89.00	–	–	(Kumar, Zhang, & Verma 2006)
BPNN	88.00	–	–	(Kumar, Zhang, & Verma 2006)
Ensemble & clustering classifiers				
SCBDL	97.50	97.50	97.50	(Verma et al. 2009*a*)
SCNN	94.00	97.83	90.74	(Verma et al. 2009*b*)
MCSVM	94.50	94.00	94.00	(McLeod & Verma 2011)
NN Ensemble	99.00	–	–	(McLeod & Verma 2013*a*)
LCA Ensemble	94.00	82.70	95.20	(Pour et al. 2012)
Neuroevolutionary classifiers				
CGPANN	89.11	88.45	92.00	-
CGPWNN	89.57	84.90	95.64	-
CGPWNN-NL	89.57	85.38	94.38	-
NEAT	89.11	86.67	92.48	-

Cao, L., Hong, Y., Fang, H. & He, G. (1995), 'Predicting chaotic time series with wavelet networks', *Physica* **D85**, 225–238.

Chen, Y., Yang, B. & Dong, J. (2006), 'Time-series prediction using a local linear wavelet neural wavelet', *Neurocomputing* **69**, 449–465.

Digital Database for Screening Mammography; accessed May (2014), `http://marathon.csee.usf.edu/Mammography/Database.html`.

Fang, Y. & Chow, T. (2006), Wavelets based neural network for function approximation, *in* 'Advances in Neural Networks ISNN, Lecture Notes in Computer Science (LNCS)', Vol. 3971, Springer Berlin Heidelberg, pp. 80–85.

Hasiewicz, Z. (1997), Wavelet neural network for density estimation, *in* 'Proceedings of Third Conference on Neural Networks and Their Applications', pp. 136–141.

Heath, M., Bowyer, K., Kopans, D., Kegelmeyer, W., , Moore, R.and Chang, K. & MunishKumaran, S. (1998), Current status of the digital database for screening mammography, *in* 'Proceedings of the Fourth International Workshop on Digital Mammography', Kluwer Academic Publishers, pp. 457–460.

Heath, M., Bowyer, K., Kopans, D., Moore, R. & Kegelmeyer, W. (2001), The digital database for screening mammography, *in* 'Proceedings of the Fifth International Workshop on Digital Mammography', Medical Physics Publishing, pp. 212–218.

Kadambe, S. & Srinivasan, P. (2006), 'Adaptive wavelets for signal classification and compression', *International Journal of Electronics and Communications* **60**, 45–55.

Khan, M., Khan, G., Ahmad, A. & Miller, J. (2013), 'Fast learning neural networks using cartesian genetic programming', *Neurocomputing* **121**, 274–289.

Kumar, K., Zhang, P., & Verma, B. (2006), Application of decision trees for mass classification in mammography, *in* '2nd International Conference on Natural Computation , Advances in Natural Computation and Data Mining', Xidian University Press, China, pp. 365–375.

Kumar, K., Zhang, P. & Verma, B. (2006), Application of decision trees for mass classification in mammography, *in* 'Proceedings of Advances in Natural Computation and Data Mining', Xidian University Press, China, pp. 365–375.

McLeod, P. & Verma, B. (2010), A classifier with clustered sub classes for the classification of suspicious areas in digital mammograms, *in* 'International Joint Conference on Neural Networks (IJCNN)', IEEE, Barcelona, pp. 1–8.

McLeod, P. & Verma, B. (2011), Multi-cluster support vector machine classifier for the classification of suspicious areas in digital mammograms, *in* 'International Journal of Computational Intelligence and Applications', Vol. 10(4), Imperial College Press, pp. 481–494.

McLeod, P. & Verma, B. (2012*a*), Clustered ensemble neural network for breast mass classification in digital mammography, *in* 'World Congress on Computational Intelligence (WCCI)', IEEE, Brisbane Australia, pp. 1–6.

McLeod, P. & Verma, B. (2012*b*), A multilayered ensemble architecture for the classification of masses in digital mammograms, *in* 'AI 2012: Advances in Artificial Intelligence - 25th Australasian Joint Conference', Vol. 7691, Springer Berlin Heidelberg, Sydney, Australia, pp. 85–94.

McLeod, P. & Verma, B. (2013*a*), Effects of large constituent size in variable neural ensemble classifier for breast mass classification, *in* 'Neural Information Processing - 20th International Conference ICONIP', Vol. 8228, Springer Berlin Heidelberg, Daegu, Korea, pp. 525–532.

McLeod, P. & Verma, B. (2013*b*), Variable hidden neuron ensemble for mass classification in digital mammograms, *in* 'IEEE Computational Intelligence Magazine', Vol. 8(1), IEEE, pp. 68–76.

Miller, J. & Smith, S. (2006), 'Redundancy and computational efficiency in cartesian genetic programming', *IEEE Transactions on Evolutionary Computation* **10(2)**, 167–174.

Miller, J. & Thomson, P. (2000), Cartesian genetic programming, *in* 'European Conference on Genetic Programming, Lecture Notes in Computer Science (LNCS)', Vol. 1802, Springer-Verlag, pp. 121–132.

Muhlenbein, H. (1990), 'Limitations of multi-layer perceptron networks - steps towards genetic neural networks', *Parallel Computing* **14(3)**, 249–260.

Panchal, R. & Verma, B. (2006), 'Neural classification of mass abnormalities with different types of features in digital mammograms', *International Journal of Computational Intelligence and Applications* **6(1)**, 61–75.

Pour, S., McLeod, P., Verma, B. & Maeder, A. (2012), Comparing data mining with ensemble classification of breast cancer masses in digital mammograms, *in* 'Second Australian Workshop on Artificial Intelligence in Health: AIH 2012, held in conjunction with the 25th Australasian Joint Conference on Artificial Intelligence', The Netherlands, Tilburg University, Sydney, Australia.

Quinlan, J. (1993), *C4.5: Programs for Machine Learning*, Morgan Kaufmann, ISBN 1-55860-238-0.

RuleQuest-Research (2014), 'C5.0: An informal tutorial; accessed august', `http://www.rulequest.com/see5-unix.html`.

Stanley, K. & Miikkulainen, R. (2002), Efficient reinforcement learning through evolving neural network topologies, *in* 'GECCO', Vol. 9, San Francisco, Morgan Kaufmann, pp. 567–577.

Steinberg, D. & Colla, P. (1997), *CART - Classification and Regression Trees*, San Diego, CA:Salford Systems.

Steinberg, D. & Golovnya, M. (2006), *CART 6.0 User's Manual*, San Diego CA:Salford Systems.

Subasi, A., Alkan, A., Koklukaya, E. & Kiymik, M. K. (2005), 'Wavelet neural network classification of eeg signals by using ar model with mle pre-processing', *Neural Networks* **18**, 985–997.

Vassilev, V. & Miller, J. (2000), The advantages of landscape neutrality in digital circuit evolution, *in* 'International Conference on Evolvable Systems,Lecture Notes in Computer Science (LNCS)', Vol. 1801, Springer, pp. 252–263.

Verma, B. (2008), 'Novel network architecture and learning algorithm for the classification of mass abnormalities in digitized mammograms', *Artificial Intelligence in Medicine* **42(1)**, 67–79.

Verma, B., McLeod, P. & Klevansky, A. (2009*a*), 'Classification of benign and malignant patterns in digital mammograms for the diagnosis of breast cancer', *Expert Systems with Applications* **37(4)**, 3344–3351.

Verma, B., McLeod, P. & Klevansky, A. (2009*b*), 'A novel soft cluster neural network for the classification of suspicious areas in digital mammograms', *Pattern Recognition* **42(9)**, 1845–1852.

Yu, T. & Miller, J. (2001), Neutrality and the evolvability of boolean function landscape, *in* 'European Conference on Genetic Programming, Lecture Notes in Computer Science (LNCS)', Vol. 2038, Springer, pp. 204–217.

Yu, T. & Miller, J. (2002), Finding needles in haystacks is not hard with neutrality, *in* 'European Conference on Genetic Programming, Lecture Notes in Computer Science (LNCS)', Vol. 2278, Springer, pp. 13–25.

Zhang, P., Doust, J. & Kumar, K. (2010), Presenting a simplified assistant tool for breast cancer diagnosis in mammography to radiologists, *in* 'Medical Biometrics - Second International Conference ICMB', Vol. 6165, Springer, Hong Kong China, pp. 363–372.

Zhang, P. & Kumar, K. (2006), Analyzing feature significance from various systems for mass diagnosis, *in* 'International Conference on Computational Intelligence for Modelling Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMA-IAWTIC)', IEEE, pp. 141–146.

Zhang, P., Kumar, K. & Verma, B. (2005), A hybrid classifier for mass classification with different kinds of features in mammography, *in* 'Fuzzy Systems and Knowledge Discovery - Second International Conference, FSKD', Vol. 3614, Springer, Changsha, China, pp. 316–319.

Zhang, P., Verma, B. & Kumar, K. (2005), Neural vs. statistical classifier in conjunction with genetic algorithm based feature selection, *in* 'Pattern Recognition Letters', Vol. 26(7), pp. 909–919.

Zhang, Z. (2007), 'Learning algorithm of wavelet network based on sampling theory', *Neurocomputing* **71(1)**, 224–269.

Comparison of Athletic Performances across Disciplines

Chris Barnes
University of Canberra
christopher.john.barnes@gmail.com

Abstract
The extreme value (EV) distribution describes the asymptotic behaviour of all stationary distributions in terms of a limiting, three parameter distribution function. This result can be used to compare elite sport performances for several purposes.
1. By regressing out most significant fixed and random effects for a given discipline, gender and class combination, the resulting residuals can be fitted to an EV distribution to determine the optimised parameters describing gender/discipline/class combinations and their uncertainties. This allows the objective ranking of athletes across events in terms of their percentiles.
2. Use of the regression models allows objective estimates of likely future event standards for performances in heats, semis and finals placings.
3. Determinations of distributional parameters allow comparison between gender, class (including able-bodied or AWD performances) and disciplines via percentiles.
4. Deviations in the regression parameters may indicate the varying effects of performance enhancement over the time-span of Olympic cycles.

Keywords: Elite sport; performance analysis; Extreme Value theorem; Weibull distribution; performance prediction

1. Introduction

In selecting athletes for a track and field team, generally fairly stringent constraints are placed on the number of athletes that can be accommodated. Perhaps the most stringent of these, at least for major meets, are those set in place by the organisers because of logistic constraints limiting the total number of competitors. There are also constraints imposed by the team management from their own logistics and budget, and also through consideration of the standard of the meet: there is generally little advantage in sending a large number of athletes likely to be eliminated in their first round or heat, or who have no realistic chance of placing near the front of the race. It will almost always be preferable for these athletes to compete at a lower level, and gain experience and confidence from meets where they have some realistic chance of influencing the major placings.

But these criteria on selection are somewhat nebulous, and without an obvious clear definition of eligibility; whereas eventually a choice must be made that an athlete is, or is not, of the requisite performance standard. Furthermore, since the entrance standard set by the organisers is usually the more stringent, it is often the case that the last athlete selected will be at the expense of an athlete in an entirely different discipline or event. For a number of reasons it is preferable that these decisions should be as transparent as possible, and based on objective criteria (rather than purely on the selectors subjective experience of relative merits).

In Athletes With Disabilities (AWD) track and field competition, there is the additional difficulty that at anything but World Championship (WCh) or Paralympic level competition, there may be too few competitors in any one class to yield the requisite closeness of competition or excitement that makes a good spectacle; but the entertainment spectacle is where an increasingly large proportion of financial support originates. This is because the numbers of AWD athletes in an event is virtually always less than the potential pool for able-bodied (AB) athletes in their corresponding event; and there are often up to or greater than 30 AWD classes for each equivalent AB event. Even at the highest international level, available numbers are insufficient to guarantee an appropriate level of competition in some event/class combinations; let alone at the far more numerous sub-national AWD competitions that allow us to identify athletic talent initially.

In the past, a number of somewhat ad hoc solutions to this dilemma have been experimented with; ranging from a failure to cater for the particular event/class combination (or equivalently, allowing more severely handicapped classes to compete in a higher

class, where they will be sorely outgunned), to introducing a class-specific handicap to allow more even competition, but requiring the development of a "fair" handicapping system. This latter choice has been adopted sometimes at the highest levels, but no such handicapping system has been found fully satisfactory, and each system tried has eventually lost support. Intuitively, for universal acceptance, a handicapping system is required to be fair (at least in the spirit of "amateur" competition on which both the Olympics and Paralympics are still ideologically based); but also sufficiently understandable that they can be "seen" to be fair. *Such a system is yet to be found, and arguably cannot reasonably be expected to exist*, because of the almost diametrically opposite features required by each of these requirements.

Given this last statement, what are the alternatives?

Historically, many handicap schemes for AWD athletics have attempted to err on the side of simplicity, while attempting to maintain as fair a system as possible. So, a number of systems based on the use of the current world's best performance(s) have been proposed and often used. While experienced proposers were often able to do a very good job in making their scheme appear fair to the majority of competitors, unfortunately in all cases competitors were found that were now patently disadvantaged by the new scheme, leading eventually to the demise of that particular system. An alternative type of handicapping system, while attempting to retain simplicity as far as possible, elected to make fairness their top priority. These systems were necessarily more complex than the first type; but all such systems proposed generally also came up short in terms of fairness. Unsurprisingly, such systems (somewhat complex, and clearly not totally fair, but in a complicated way) enjoyed even less support than the former type in the long-term.

In this paper, we take another look at the second category of solutions to the simplicity vs accuracy dilemma: we introduce a relatively complex type of analysis where fairness is more-or-less guaranteed, at least given sufficient data. While initially the fairness is only approximately guaranteed, with increasing amounts of data (guaranteed by time) it becomes increasingly more precise. For many AWD event/class combinations, the methodology is already capable of good precision in comparing performances across events and classes; possibly including comparisons between AB and AWD performances. As a side product, it also throws light on variations in elite performances over time, and also on the probability (prediction) of different levels of performance in the near future.

2. Methodology

This methodology combines a number of statistical and data-mining processes to characterise performances in any athletics event (or event/class combination) in terms of a small number of parameters specific to each event, but *independent of time*. In this way, a single elite performance may be compared accurately with other performances achieved under somewhat different conditions at different times. One price paid for this generality is the introduction of a degree of uncertainty in deciding the rankings of superior performances; representing the uncertainty due to unknown (ignored but presumed random) effects on performance.

In summary, we start with a regression model that includes all measured and identified significant effects (such as gender, wind speed and direction (if known), date of performance etc.); also including the identities of each athlete as random effects, since the data set will normally contain repeated measures for at least some athletes. With all significant effects identified in this way, it is assumed that the residuals (measured-modelled) are independent and identically distributed (iid): in particular they should be independent of time if the regression model is adequate. In other words, given the adequacy of the regression model, the residual series can be represented by a stationary iid distribution.

Under these conditions, the Extreme Value (EV) theorem tells us that the tail of this distribution of residuals asymptotically has a fixed form, fully described by only two (or three) parameters. By considering the average speed, rather than the total time, for track events and similar, we can assume in all athletics events that elite performances are characterised

by the upper tail of the distribution (bigger is better); the relevant form of the distribution is then the Weibull (also known, in transformed form, as the Extreme Value) distribution. This distribution is defined on the [0,∞] interval, so that it is assumed that there is a non-zero possibility of any finite performance, although the likelihood of really high performances becomes vanishingly small.

However, the asymptotic adherence to the Weibull form does not guarantee the stationarity of the series of residuals: our assumptions require also that the distribution of performance residuals for each year or period should individually follow this same asymptotic form of distribution obtained for the whole series. This somewhat more stringent requirement allows us to examine our assumptions rather more critically.

Putting all this together allows us to characterise performances for a single event as being comprised of three parts:

a) A fixed part that takes into account the known effects (such as gender, time and wind speed);

b) An individual part that allows for the specific prowess of individual athletes (the random effect, representing inter-athlete differences); and

c) A random part that allows for unknown effects that characterise the typical intra-athlete variation in performance for the athlete population in question. An application of the EV theorem shows that the upper part of this distribution is characterised by a Weibull distribution.

3. Results

Recently, (July/August) the 2014 British Commonwealth Games were held in Glasgow, Scotland. Part of these Games took the form of a swimming competition for both men and women, including AWD (Athletes-With-Disabilities) and AB (Able-Bodied) athletes. To introduce the methodology, we consider results for females for the freestyle swimming stroke at all distances raced, varying from 50m to 1500m. In order to maximise the accuracy of our prediction analysis, we used a comprehensive dataset obtained from the Infostrada website (www.infostradasports.com). The data importation, and all statistical and mathematical analysis, was carried out using SAS JMP, v. 11.2.0. I have adhered to their nomenclature wherever possible.

For free-style (FRS), the data consists of > 23,000 records (extracted on 07 Aug 2014); each record represents an individual race performance (accepted by FINA as an official time for all recent performances).

The first part of this analysis uses a linear regression model to account for effects due to any of the following that are applicable:

i. the Pool Length (SCM or LCM);
ii. the race Distance;
iii. the Stroke (FRS or free-style, in this example);
iv. Gender (Female);
v. Competition Type (Olympics, World Championships, Grand Prix etc.);
vi. Competition Phase (Heats, Semis, Finals…);
vii. the Date of the event; and
viii. Athlete identity (since relative athletic prowess varies with distance, the Random Effect was taken as race Distance nested within Athlete ID).

The regression y-variable was identified as the natural logarithm of the average speed; this had the happy outcome that small changes in regression coefficients could be interpreted as % relative changes, and also satisfied the "bigger is better" requirement (see above).

In addition, the time-variable (Date) was broken up into two parts: the Olympic quadrennium represented by the Olympic Year (as an ordinal variable), which allowed the aggregation of the data from 4 years rather than just one; and the Olympic Cycle (year (mod 4)), which allowed for a periodicity in aggregated performance times that is observed in some events.

Interactions between main effects were fitted, but kept in the final model only if they were found to be significant in at least one of the regressions.

4 Case study: Female FRS – "Average" elite performance

For women's freestyle, the resulting fit accounted for 98.8% of the variation (N=19,573 plus 271 from the 2014 Commonwealth Games, which were excluded from the initial analysis). The goodness of fit is somewhat artificially enhanced due to the wide range of Distance included in the models; if Distance was included as a by-variable instead, essentially the

same analyses resulted, but with much lower regression R^2s.

The actual model fitted in this case was:

Ln (speed) ~ Name [Distance] (Random) +Competition Type + Olympic_Year [Distance] + Olympic_Cycle + Distance,

accounting for 110 degrees of freedom in all for this model. The only interaction term considered for main effects was Olympic_Year [Distance], which allows for the possibility that variations of swim speed with time may depend on the distance being swum. Also, with this large number of observations, almost all reasonable interactions may be expected to be significant, even though they are found to be practically negligible. In fact, such was the case for the fixed effect Olympic_Cycle, which was included for comparison with other regressions

.

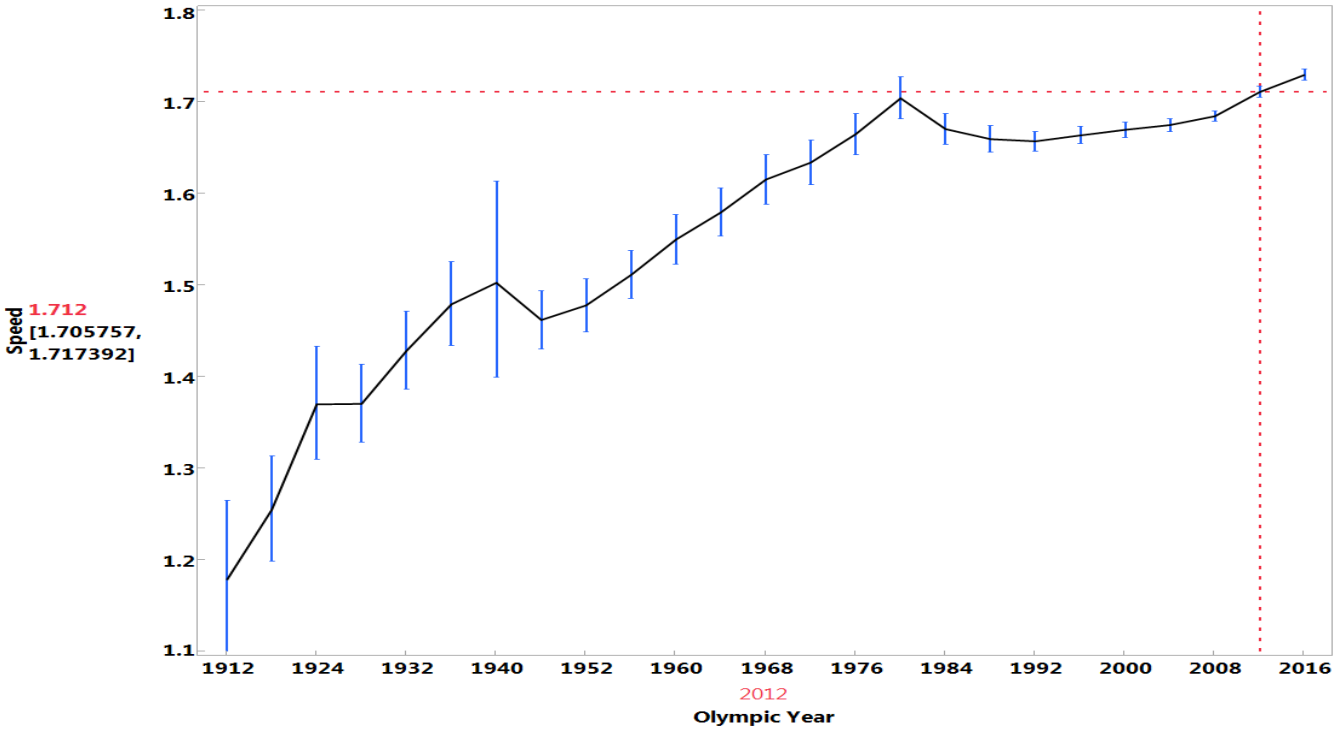


Fig.1a The profile of "average" swim speed vs time. Other parameters were: Olympic cycle=0; Competition=Olympic Games; Distance=100m. The figures on the left show the swim speed and uncertainty for the Olympic main fixed effects which also shows their interactions. Year 2012. Estimated standard errors are also shown

After fitting this regression, we obtain a "profile" of the variation in speed against all the main fixed effects which also shows their interactions. In this case we have four fixed effects: Olympic Cycle, Competition Type, Olympic Year and Distance; but only the last two have a major influence, although all are statistically highly significant. Graphs of speed against Olympic_Year and Distance account for the vast majority of the regression variance (Figures 1a and 1b).

Figure 1a represents the regression fit for swimming speed, from 1896 to the present, for the Infostrada data. Ignoring the inevitable bias inherent in the data due to the different qualities of the fields and the number of events represented by each 4 year period (the 1896 data is all from the Olympic Games, whereas more recent data includes local, national, regional and global meets), this data shows interesting features of the generally increasing event speed with time. Of particular interest is the section of the graph post 1980, where the regression speed at first peaks and then decreases, before starting to rise relatively rapidly in recent years (especially post 2008). Reasons for this behaviour potentially include interactions between: the use of faster swim suits and their subsequent banning post 2000; a reaction to increasing detection of illegal performance enhancement drugs; and the effects of better coaching techniques coupled with better understanding from sports science, particularly around starts and turns Also of interest is the effect of distance on average speed (or pacing) (Fig. 1b), which shows a rapid diminution in speed as the distance increases from 50m to 200m, perhaps even a suggestion that it diminishes more rapidly between 100m to 200m than between 50m and 100m. This may be influenced by the differential effect of the number of turns relative to the start at different distances.

Finally, for the regression model, the values of the random effect estimates, or the BLUP (Best Linear Unbiased Predictor), (essentially the overall advantage of a particular swimmer over the "average" swimmer of the regression fit, considered as a % advantage), are also of interest. This gives rise to a normalised ranking list, allowing the comparison of individual swimmers (and distances) over time. For instance, the top 10 Female 100m swimmers from this list are given in Table 1, together with their nationality and years of competitive swimming.

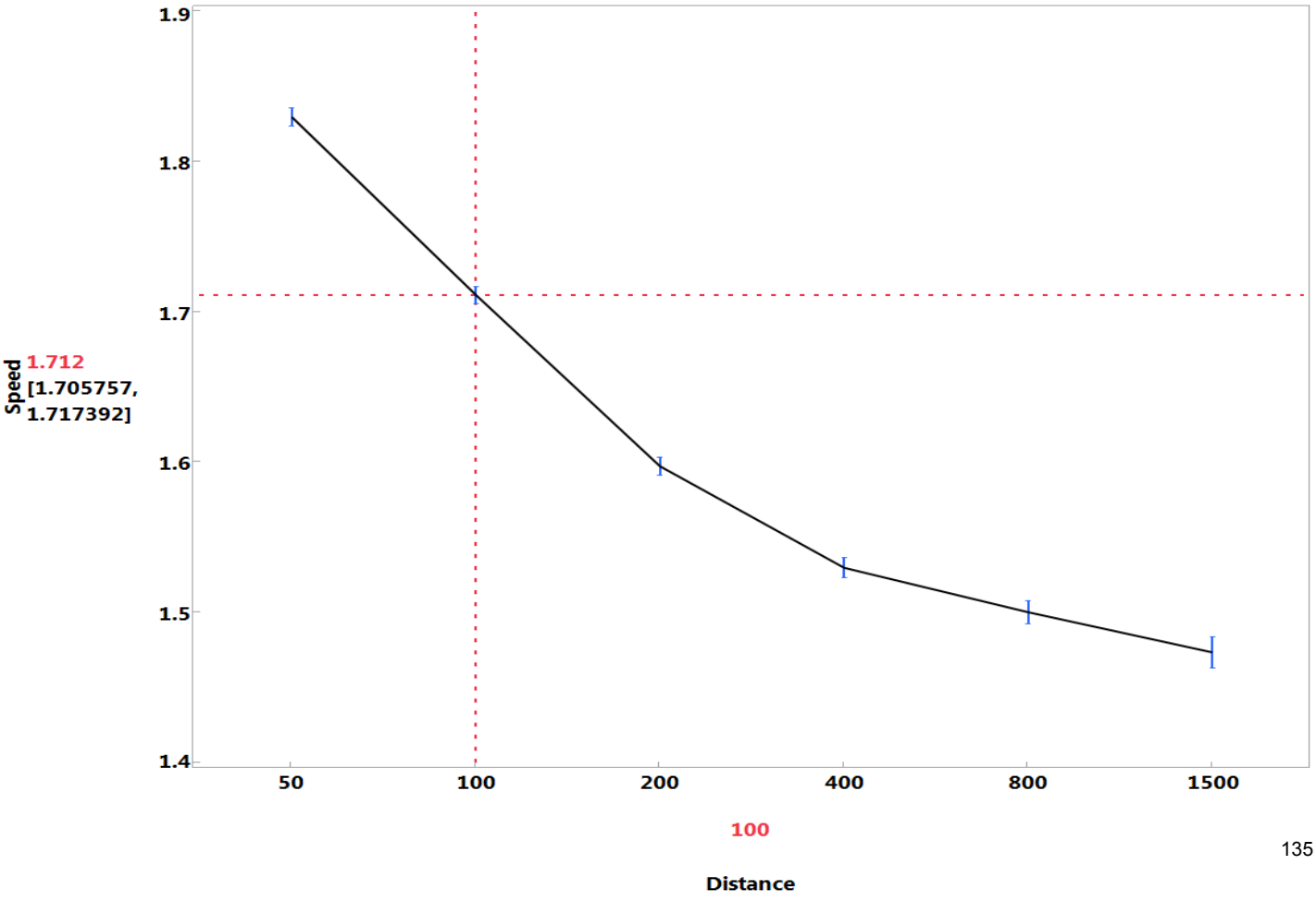


Table 1 Top 10 ranking list for 100m (relative to peers)

Name	BLUP	Lower 95%	Upper 95%	Nation	Min(Year)	Max(Year)
Lu Bin	0.110	0.090	0.129	CHN	1992	1994
Libby Trickett	0.102	0.096	0.108	AUS	2003	2012
Jodie Henry	0.099	0.092	0.106	AUS	2000	2007
Le Jingyi	0.096	0.084	0.108	CHN	1992	1996
Cate Campbell	0.095	0.088	0.103	AUS	2007	2014
Britta Steffen	0.094	0.087	0.101	GER	2006	2013
Inge de Bruijn	0.093	0.083	0.102	NED	1991	2004
Nicole Haislett	0.092	0.078	0.107	USA	1991	1994
Marleen Veldhuis	0.092	0.085	0.100	NED	2003	2012
Franziska Van Almsick	0.092	0.083	0.100	GER	1992	2004

The REML analysis shows that only about 4.5% of the variance is accounted for by the fixed effects, while the random effect of "inter-athlete advantage" accounts for over 95% of the overall variance. If the empirical distribution of the 6700 individuals in the database is plotted separately by distance (Figure 3), it is clear that the dispersion is somewhat greater for the lower distances (50m-100m) than for the other distances. In other words, relatively speaking, there is a somewhat greater spread for the smaller distances than for the higher ones.

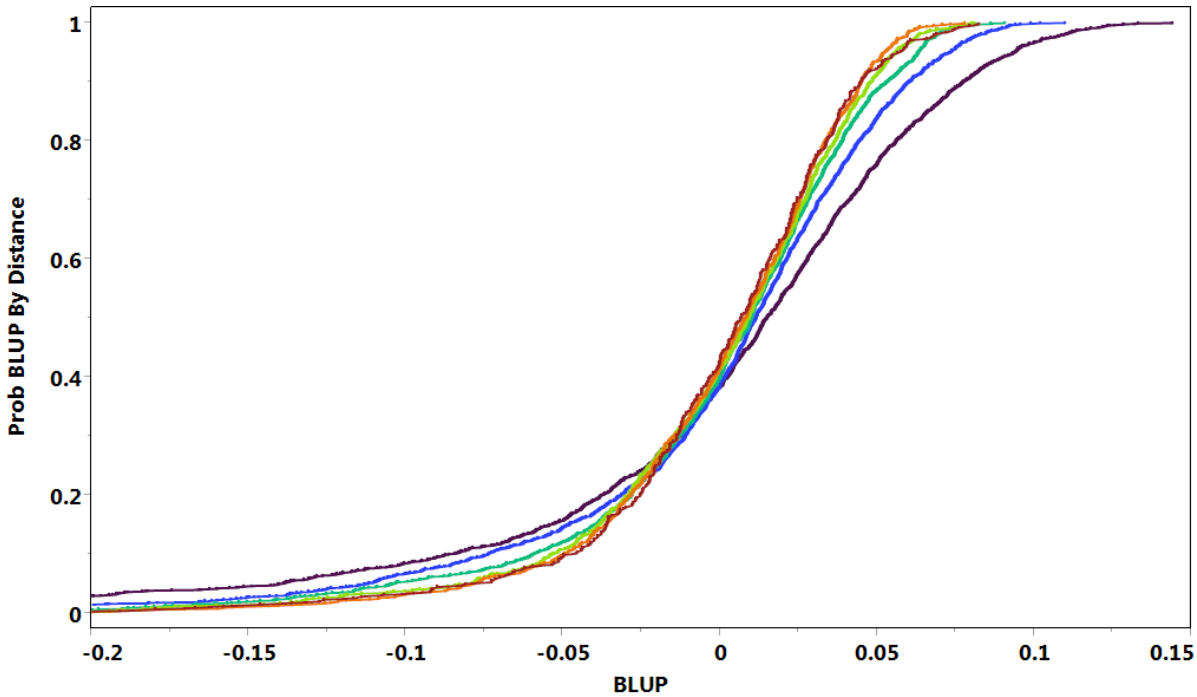


Figure 2 The BLUP distributions for each of the 6 freestyle distances analysed: Distance=50m. 100m. 200m. 400m. 800m. 1500m.

5 Residual analysis

So far, the analysis has accounted for the important fixed effects and interactions, and the random effects associated with intra-athlete differences. The remaining variance may be considered as irreducible error caused by ignored or unknown effects, such as seasonal effects and taper (only partially included in competition type), or influences such as emotional and mood swings, factors affecting mental preparation, and so on. This irreducible error still remains even if we have an effectively infinite number of performances by an athlete in our data.

Comparing each individual performance with the value predicted by the regression (including the random effect for the athlete), we obtain the "conditional residual" as the difference
:

conditional_residual=speed-conditional_predicted_speed.

Since we have more or less "regressed out" all significant effects, including the repeated measures, we assume that the conditional residuals are iid. The Extreme Value theory affirms that the upper extreme of such a distribution will be asymptotically Weibull, above a certain threshold. We fitted a 3-parameter Weibull distribution (the location parameter α, the shape parameter β, and the threshold parameter θ) to the upper part of the empirical cumulative distribution of residuals:

conditional_residuals ~ W(α, β, θ);

where parameter values and confidence intervals are given in the table
:

Parameter	Estimate	ApproxStdErr
alpha	0.0482	0.00019
beta	2.281	0.0072
theta	-0.0478	0.00019

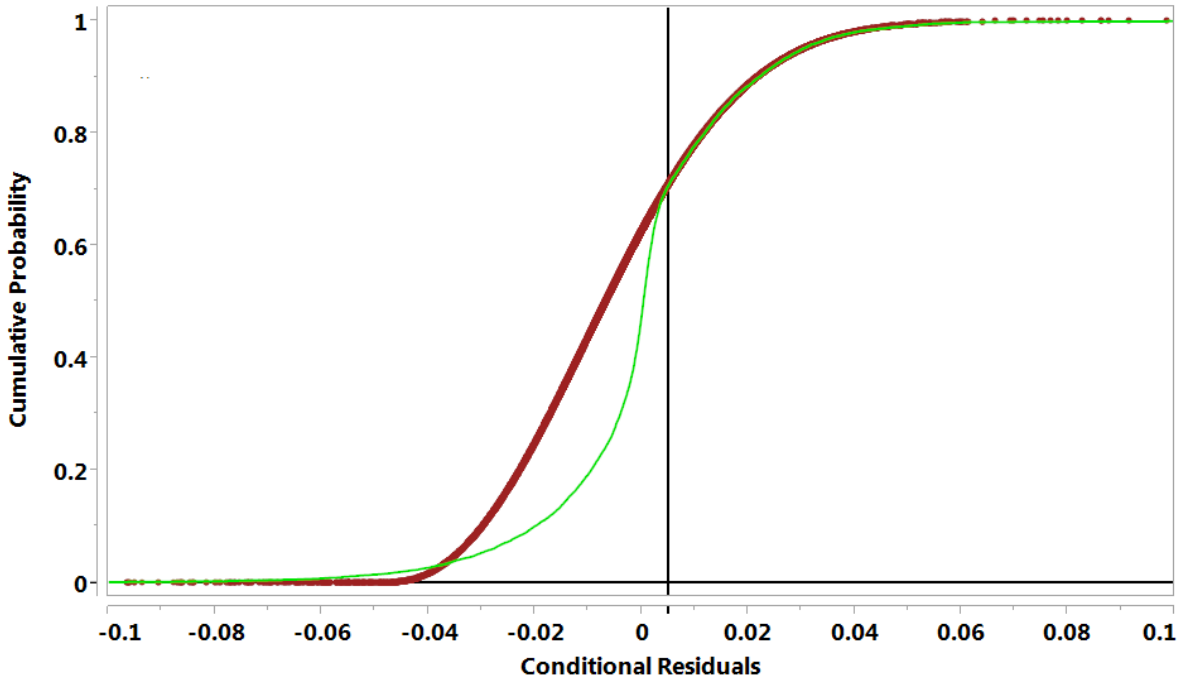


Figure 3 Plot of the empirical distribution of residuals (heavy line), and the fitted Weibull distribution, showing the close fit above a threshold of 0.005, as demanded by the EV theorem. Note that the three estimates (α, β, θ) are generally highly correlated.

Although it is possible to determine separate distributions for each athlete, in practice this is likely to lead to difficulties, as many emerging athletes of interest may have few performances on record. By using an average value for all athletes it is possible to give a reasonable estimate of the typical variation that has been observed in elite athletes, making for a more robust approach, though potentially slightly less than optimal.

The 5th and 95th percentiles of the above distribution give the range of the uncertainty inherent in predicting the performance from an elite swimmer, in any individual freestyle event. In fact, as for the BLUP estimates (cf. Table 1), there are small differences between results for different distances, but the differences are small and for simplicity we choose to ignore them.

6 Performance prediction

Finally, we can predict the performance of any athlete in the 2014 Commonwealth Games in Glasgow freestyle competition. Firstly, we calculate the prediction for each athlete in 2014 from the regression equation, augmented by their specific athlete advantage, and also the confidence limits around this expected value which allows for the uncertainties in the athlete advantage explicitly obtained in the same process. This 90% confidence region is further widened by adding the Weibull uncertainties obtained from the pooled residuals analysis above to obtain an approximate confidence interval for the predictions; expecting that about 90% of the data should fall within this augmented region.

The estimated uncertainty in the BLUP estimates for athlete advantage contributes a factor of

$$\mathbf{F\pm = exp(\pm 1.96 * SEofBLUP)}$$

to the confidence interval, which must be multiplied by the conditional estimate of the speed for each athlete.

We have also shown above that, to a good approximation, the uncertainty bounds in the predicted speed for any athlete is given by the Weibull distribution:

$$(Wq(0.05, \alpha, \beta, \theta) , Wq(0.95, \alpha, \beta, \theta))$$

Where $Wq(x, \alpha, \beta, \theta)$ represents the x^{th} quantile of the 3-parameter Weibull distribution.For the 100m women's freestyle for Glasgow 2014 we obtain the following graph of predictions and results for the top 45 places, by simply omitting those who were not present in the all-time performance ability list, and combining error estimates of different sources using a root sum of squares addition:

$$\mathbf{SE = (SE_1^2 + SE_2^2)^{1/2}}$$

where the subscripts 1 and 2 refer to the BLUP standard error (expressed in absolute terms), and the Weibull standard error respectively.

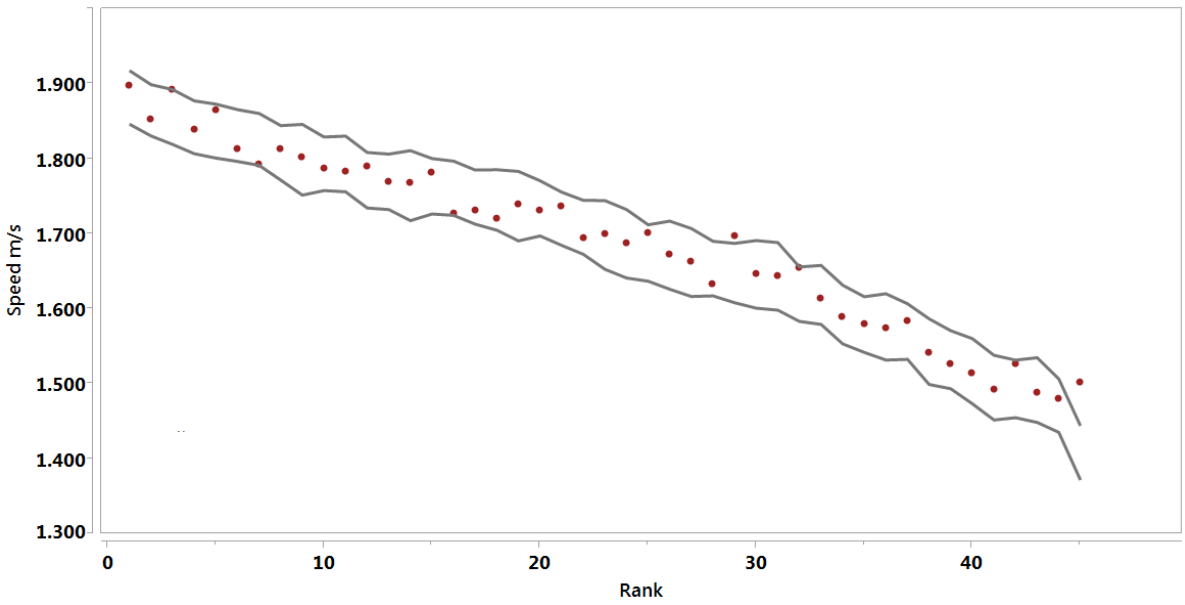


Figure 4 Predicted performances (grey lines as 90% confidence intervals) and ranks for Female 100m FRS at Glasgow 2014, compared with actual performances (closed dots).

The calculated bounds appear to be possibly too loose, as only 2-3 points appear to lie outside the bounds (all above), whereas 4-5 points may theoretically be expected; but overall the

predictions are matched quite well. The variation in ranks from those predicted illustrates the uncertainty inherent in those predictions: predicted ranks (1,2,3,4,5,6,7,8,9) become actual (1,4,2,5,3,6,9,7,8) for the first 9 competitors). In other words, one athlete who should have expected a silver medal on rankings was relegated to bronze, whereas another potential podium finisher missed out, and a third person was able to gain a silver medal when expectations were not to finish in the top 3 at all.

7 Conclusion

In conclusion we have demonstrated a methodology that essentially ranks athletes against their peers over time, by measuring their relative strength against as many of their peer performances we have available, taking into account all measureable influences through a regression approach. We have shown that this approach can be highly accurate, being largely free of the instabilities which have plagued other methods of comparison. This makes it suitable even for comparing athletes across classes, or even different events, such as is required in selection, or for combining AWD classes in the same event.

We have also shown that the methodology may throw light on the way in which elite performances change over time, having uncovered seemingly irregular behaviour in the generally highly stable average quality of performance over time. These irregularities suggest further work to disentangle the influences combining to produce this observed behaviour.

Although due to lack of space, the swimming discipline was chosen as the sole example, it is clear that the same technique (with some modifications) could be used for any athletic event (100m, shot), or for rowing events, for example. Indeed, for AWD events, where we must match performances across disability classes, the same conclusion holds, providing there are sufficient results to guarantee the desired accuracy; which can be guaranteed by combining results from a number of events over time through a regression fit.

Finally, we have demonstrated a methodology to characterise the random variations in elite performances, whether for a single athlete or for the whole population of elite athletes. We have shown that, removing temporal influences by regression, the remaining stationary distribution (of residuals) must be characterised by a single type of distribution, the Weibull, whose parameters can be accurately determined for each combination of classes. Equating performances across classes via the distribution percentiles then allows quantitative comparisons of performance merit.

We have shown that the combination of these three parts of this methodology can be used to predict both the expected performance of elite athletes, and also to quantify the uncertainty inherent in those performances. This potentially allows us to study in much greater detail such effects as the efficacy of different types of taper, of mental preparation or of diet, for example.

References

Infostrada (2014): Data obtained by subscription from http://www.infostradasports.com/.

JI McCool, 2012, *Using the Weibull Distribution: reliability, modelling and inference, John Wiley.*

JMP Pro 11.2.0 (2013) © SAS Institute Inc.

Locality-Sensitive Hashing for Protein Classification

Lawrence Buckingham, James M. Hogan, Shlomo Geva, Wayne Kelly
School of Electrical Engineering and Computer Science,
Queensland University of Technology,
GPO Box 2434, Brisbane 4001, Queensland.
(l.buckingham,j.hogan,s.geva,w.kelly)@qut.edu.au

Abstract

Determination of sequence similarity is a central issue in computational biology, a problem addressed primarily through BLAST, an alignment based heuristic which has underpinned much of the analysis and annotation of the genomic era. Despite their success, alignment-based approaches scale poorly with increasing data set size, and are not robust under structural sequence rearrangements. Successive waves of innovation in sequencing technologies – so-called Next Generation Sequencing (NGS) approaches – have led to an explosion in data availability, challenging existing methods and motivating novel approaches to sequence representation and similarity scoring, including adaptation of existing methods from other domains such as information retrieval.

In this work, we investigate locality-sensitive hashing of sequences through binary document signatures, applying the method to a bacterial protein classification task. Here, the goal is to predict the gene family to which a given query protein belongs. Experiments carried out on a pair of small but biologically realistic datasets (the full protein repertoires of families of *Chlamydia* and *Staphylococcus aureus* genomes respectively) show that a measure of similarity obtained by locality sensitive hashing gives highly accurate results while offering a number of avenues which will lead to substantial performance improvements over BLAST.

Keywords: bioinformatics; sequence comparison; alignment free.

1 Introduction

The determination of sequence similarity is a fundamental problem in bioinformatics, underpinning general query based search of public databases, the propagation of annotations based on the relatedness of genes, and the inference of phylogenetic relationships among organisms. While alternative and specialised algorithms may be used for particular tasks, most similarity determination in bioinformatics relies on weighted sequence alignment methods such as Clustal (Thompson et. al., 1994) and in particular on an heavily optimised heuristic known as the Basic Local Alignment Search Tool (BLAST; Altschul et al, 1990).

While these tools have been enormously successful, alignment based methods suffer from two key problems for large scale comparative genomics: alignment algorithms are based on dynamic programming and so quadratic in sequence length; and alignment methods are not robust in the presence of structural sequence rearrangements i.e. when closely related sequences vary not merely through small numbers of local substitutions, insertions and deletions, but rather through the insertion or deletion of a large contiguous fragment. This latter issue is particularly important in studies of the development of infectious diseases and antibiotic resistance, both of which frequently involve a process known as Lateral Genetic Transfer (Skippington and Ragan, 2011). These limitations have motivated work on alignment-free methods for sequence analysis (Reinart et. al., 2009) requiring alternative representations of the sequence. Alignment-free approaches usually employ some tokenization of the sequence content into short words of fixed length k known as k-mers, with scoring methods for similarity defined over the resulting (normalised) bag-of-words representation.

The selection of k is a crucial parameter, controlling the dimension of the vector space in which the sequence vectors are embedded. Choices of $k \in \{5,6,\ldots,25\}$ are typical for nucleotide (DNA) sequence analysis, dependent upon the application, while a choice of $k \in \{2,3,\ldots,7\}$ is appropriate for amino acid (protein) sequences, which are derived from nucleotide sequences via a triplet code. Approximate matching is handled through the tiling effect of overlapping k-mers, and by the use of a small mismatch neighbourhood for each k-mer – in essence the number of positions in the k-mer in which a variation is allowed while still being treated as a match. Such approaches usually rely on data structures such as the suffix trie for storage of these relationships, with the consequent computational burden when similarity is evaluated.

In the present work we consider an alternative sequence similarity measure based on methods developed in the information retrieval community for large scale document indexing and clustering. In common with other alignment-free sequence comparison methods, the sequence (or a window into the sequence) is tokenized to produce a list of overlapping k-mers which are analogous to the words appearing in a document. A pseudo-random projection is applied to the k-mer list, generating a binary string which acts as a signature for the sequence. The projection function employed is continuous, in the sense that two sequences having similar k-mer content will tend to have similar signatures – an aspect of locality sensitive hashing which stands in contrast to cryptographic hash functions, where it is desirable to minimise collisions.

Sequence similarity may then be computed rapidly through bit level determination of the Hamming distance between the signatures, without resort to dynamic programming or to traversals of complex data structures.

As a first step in this work, we investigate the use of these methods in a realistic bacterial protein classification task, and measure its effectiveness as a proxy for BLAST, the tool most widely used for such problems. Although we anticipate that an optimised signature based approach will yield significantly better performance than BLAST, we do not address that aspect in the current experiments; rather, we seek first to establish the extent to which a signature based similarity measure produces results that are consistent with those obtained via BLAST.

This paper is organised as follows. In the following section, we introduce the background on random signature methods essential for the remainder of the paper. In Section 3, we introduce the classification task and the adaptation of the random indexing methods described in Section 2 to a biological setting. Section 4 describes our experiments in detail and the results obtained, before concluding in Section 5 with a discussion and consideration of future work.

2 Random Indexing

Object signature approaches to information retrieval represent text documents, images and other searchable abstract objects, as binary strings of fixed length, called signatures. The starting point is invariably a vector-space object representation appropriate for the objects. For instance, a bag-of-words representation is most commonly used for text. A text document is then represented as a sparse vector of term frequencies or term-weights, whereby each vector component corresponds to a word in the vocabulary, but the only non-zero vector components correspond to terms that actually appear in the document. Images can be similarly treated – an image can be represented by a bag of visual features derived using various image feature extraction methods. Object signatures are then derived from the vector-space representation so that they preserve in signature space the mutual topological relationships that exist in the original representation of objects.

Locality-sensitive hashing (LSH) (Slaney, 2008) is a dimensionality reduction method, which transforms vector-space representations of objects into binary signatures. The original representation of an information object is typically a very sparse, high dimensional feature vector derived with some probabilistic language model, or other suitable feature extraction/definition approach. The binary signatures used in LSH methods offer compression of the original representation onto a dense, fixed, low-dimensional representation. LSH allows for comparison of two different objects for similarity far more efficiently than traditional methods such as cosine similarity, especially if the source objects are large. For instance, a relatively expensive cosine similarity computation between two large text documents may be replaced with a Hamming distance calculation over concise binary signatures. This efficiency motivates most applications of LSH and signatures, such as in information retrieval and near-duplicate detection. There are a number of different document signature models in use, including Minhash (Broder, 1997), Simhash (Sadowski and Levin, 2007), Topsig (Geva and De Vries, 2011) and Reflexive Random Indexing (Vasuki and Cohen, 2010), all of which use some variation of LSH.

Random projection is a simple and commonly used approach to LSH for achieving dimensionality reduction. Applications to image and text objects are described by (Bingham and Mannila, 2001). The basic approach is to project the original d-dimensional data to a reduced h-dimensional ($h \ll d$) subspace, through a random $h \times d$ matrix whose vectors have unit length. The Johnson-Lindenstrauss Lemma (Johnson and Lindenstrauss, 1984) assures us, in practical terms, that by selecting large enough h, the distance relationships between vectors will be preserved. In practice it turns out that h can be quite small relative to d – for instance, text documents are often projected with h of the order of 100 to 1000, while the cardinality of the vocabulary, d, may be many millions of terms.

Sahlgren (2005) describes an efficient implementation of random projection over highly sparse data, called random indexing. The approach is based on a random projection of objects from input space onto the $\{\pm 1\}^N$ hypercube. In the case of text, a simple random projection is achieved as follows. Each document is processed one term at a time. The term is used as a seed to a random number generator in order to generate a pseudo random sequence of values from the set $\{+1, 0, -1\}$. A common approach is to choose 1/6 of the values to be +1, 1/6 to be -1, and the rest to be 0, e.g. (Bingham and Mannila, 2001) and (Sahlgren, 2005), but other values may be used. Term signature values may be weighted at this point, for example TF-IDF in text applications.

Once all the term signatures are summed we have a real valued vector of dimensionality h. When terms are weighted this is a linear combination of all term signatures in the document. This vector is then converted into a binary string by applying the sign function. All non-negative values assume the value of 1, and all negative values assume the value of 0. The resulting document signature is then packed into a binary string.

With this representation it is now possible to perform efficient matching operations for a variety of applications. The collection of objects is first projected onto signature space and a database of object signatures is thus created. This is essentially a simple matrix where each row is a binary signature corresponding to one of the original objects. To find database objects similar to given search object, first we transform the search object representation into a binary signature using the same random projection process, and then perform Hamming distance calculation against the database to rank the objects by similarity to the search argument.

The following section sets out specific details of the *k*-mer based tokenization and random projection function employed to adapt the information retrieval techniques described above to a biological setting.

3 Methods

3.1 Overview

We use a protein classification task to assess the extent to which our signature based method yields results which

are consistent with those obtained via BLAST. Tasks of this nature arise routinely in functional annotation of sequence data and identification of families of orthologous genes, giving the results of the evaluation immediate practical relevance. We conducted two experiments using reference data representing distinct groups of bacteria obtained from the NCBI bacterial genome ftp site[1].

The experimental design consists of the following broad steps. First, a collection of protein sequences is obtained from a central repository and partitioned into mutually exclusive similarity groups of distinct, orthologous sequences. From each sequence we extract one or more overlapping component fragments, and each of these fragments is labelled with metadata which includes the identity of the originating sequence, fragment location and class label of the protein family to which the sequence belongs. A binary signature is then derived for each document fragment and these are saved with the associated fragment metadata to form the signature database.

To classify a query sequence, a set of component fragments is extracted from the query sequence and a binary signature is derived for each fragment. The database is then scanned to locate the $K \geq 1$ most similar sequences to the query. The similarity between the query and a candidate hit sequence is obtained by computing a pairwise similarity based on Hamming distance between binary signatures of all fragments in the query sequence and all fragments in the potential hit sequence and taking the maximum. A majority vote of the classes of the K most similar hit sequences is taken as the predicted class for the query. Tied votes are resolved in favour of the class to which the best matching hit sequence belongs.

The following subsections describe the preparation of baseline data and signature generation in greater detail.

3.2 Construction of base-line datasets

Our experiments rely on prior classification of protein sequences into similarity families; however this information is not readily obtained from the standard reference data. Although reference genomes are annotated to some extent with functional descriptions and symbolic names for many genes, these annotations are often inconsistent or incomplete, with many genes simply designated as "hypothetical protein". In the absence of definitive protein family classifications to support automated analysis, we employ a clustering algorithm to group protein sequences into putative families based on BLAST similarity.

Much earlier work has been devoted to clustering of protein families and protein domains, with the bulk centred on bottom-up agglomerative methods such as single linkage clustering, which proceeds by successively merging the most similar pairs. Harlow *et al* (2004) examined the problem of protein family clustering in detail, obtaining superior results by first applying the Markov Cluster algorithm (MCL) (van Dongen, 2000; Enright *et al*, 2002) to partition the protein sequences into equivalence classes based on a normalised BLAST similarity score before proceeding to use single linkage to map out fine-grained topology within clusters. With the findings of Harlow *et al* (2004) in mind, we use MCL to derive class labels in the present work[2].

Given a collection of reference sequences obtained from the genomes of one or more organisms, a non-redundant dataset is constructed in which duplicate sequences are replaced by a single representative instance. The non-redundant dataset is saved as a FASTA-formatted file which is used to construct a corresponding BLAST database.

NCBI BLAST is executed with the non-redundant dataset used as both query and database to obtain bidirectional BLAST bit scores and e-values (Ewens and Grant, 2001, p278) for each pair of potentially orthologous sequences in the dataset. Default settings are used for all parameters of the BLAST program. A sparse mutual similarity matrix is derived from the BLAST result dataset by filtering the results by e-value and normalising the bit scores to obtain a pairwise similarity score for each remaining pair. Given sequences (a, b) having bidirectional bit scores $S_{a,b}$ and $S_{b,a}$ and e-values $E_{a,b}$ and $E_{b,a}$, we select pairs satisfying $\min\{E_{a,b}, E_{b,a}\} < 10^{-3}$. The similarity for the pair is then obtained by dividing each of the bit scores by the corresponding self-hit bit score, giving $S'_{a,b} = S'_{b,a} = \max\{S_{a,b}/S_{a,a}, S_{b,a}/S_{b,b}\}$.

As noted above, MCL (Van Dongen, 2000) is applied to the mutual BLAST similarity matrix to partition the non-redundant sequences into families. The MCL inflation parameter is set to 1.1, a value experimentally determined by Harlow et al (2004) to yield protein clusters which are consistent with those obtained by manual curation via analysis of data from the Protein Data Bank (Berman et al, 2000). A unique identifier is defined for each equivalence class and this label is applied to the corresponding protein sequences.

3.3 Signature generation

The classification task is determined by a binary signature database which contains one or more entries for each sequence. In this work, sequences are the textual representations of amino acid chains which are represented by strings over an alphabet $\mathcal{A}$ made up of 20 characters. The signature of a sequence is a vector of $N > 0$ bits which is computed as a flattened linear combination of the signatures of the *k*-mers which appear in the sequence. A *k*-mer is a substring of the sequence having length $k \geq 2$. The number of non-zero bits set in an individual *k*-mer signature is controlled by a density parameter $0 < D < 1$.

To compute the signature of sequence

$$A = a_1 a_2 \dots a_M \in \mathcal{A}^M$$

we first obtain the *k*-mers appearing in A:

$$a_i a_{i+1} \dots a_{i+k-1}, i = 1 \dots M + 1 - k$$

and for each of these compute the corresponding *k*-mer signatures

$$t_i = (t_{i1}, t_{i2}, \dots, t_{iN})$$

[1] ftp://ftp.ncbi.nlm.nih.gov/genomes/Bacteria/

[2] Notwithstanding the effectiveness of MCL, we would emphasise that the approach is independent of the clustering method selected.

$$= \mathrm{Sig}(a_i a_{i+1} \ldots a_{i+k-1}; D, N), i = 1 \ldots M + 1 - k$$

As discussed earlier in Section 2,

$$\mathrm{Sig}(\cdot; D, N): \mathcal{A}^k \longrightarrow \{-1,0,1\}^N$$

is a function which maps a *k*-mer to a pseudo-random vector with N elements, of which $\left\lfloor \frac{(D+1)}{2} \right\rfloor$ are non-zero. Of the non-zero elements, half are assigned the value $+1$ and the others are set equal to -1. In the experiments described below an open source implementation of the RC4 stream cipher algorithm (Paul and Maitra, 2011) is seeded with *k*-mer $a_i a_{i+1} \ldots a_{i+k-1}$ to generate a sequence of pseudo-random indices in the range $1..N$. These determine the placement and sign of the non-zero elements in the resulting *k*-mer signature.

The *k*-mer signatures are then combined and flattened to produce the final signature:

$$S = (s_1, s_2, \ldots, s_N), \text{where}$$
$$s_i = \begin{cases} 1 & y_i > 0 \\ 0 & \text{otherwise} \end{cases}, \text{and}$$
$$y_i = \sum_{j=1}^{M+1-k} t_{ji}.$$

For small word length k, the relatively small *k*-mer vocabulary (20^k distinct words) creates a situation in which the probability that each bit in the resulting signature will be set increases with increasing sequence length. This occurs because the probability of any given *k*-mer appearing by chance in a sequence increases with the length of the sequence, becoming non-trivial for every *k*-mer when the length of the sequence becomes comparable to the size of the vocabulary. To mitigate this effect we partition each sequence into fragments and compute a separate signature for each fragment. For given fragment length F, a sliding window of length F is passed over each sequence at intervals of $\lfloor F/2 \rfloor$ characters, yielding a collection of overlapping subsequences which cover the original sequence. The signatures of all fragments are added to the signature database. The similarity of a pair of fragments is obtained by subtracting the normalised Hamming distance between their respective signatures from 1.

To determine the similarity of two sequences, *A* and *B*, each fragment of *A* is compared to each fragment of *B*, giving a set of pairwise fragment similarities. The similarity of *A* and *B* is then the maximum pairwise fragment similarity.

3.4 Evaluation of classification results

To evaluate classification results we take a sequence dataset and apply the process defined in Section 3.2 to generate class labels for each sequence. Then for a range of meta-parameters (N, D, F, k) (see Section 3.3) we construct a signature database which forms the basis of a nearest neighbour protein classifier.

The accuracy of the protein classifier is evaluated by carrying out 10 repeats of 10-fold cross validation as follows. The sequence dataset is partitioned into ten mutually exclusive subsets. In turn, each subset is held out for use as a test set while a reduced classifier is created by removing any signatures belonging to the test set from the signature database. Any isolated sequences – those that belong to a cluster with only one element – are removed from the test set as it is impossible to classify them correctly. We then use *K*-nearest neighbour search as described in Section 3.1 to classify each remaining member of the test set.

As each part is classified we maintain a confusion matrix to record the results. Initially for all classes c, c' we set $M_{c,c'} = 0$. When we test sequence i with expected class c_i and predicted class c_i', the matrix is updated by: $M_{c_i,c_i'} \leftarrow M_{c_i,c_i'} + 1$. From the confusion matrix we derive precision and recall scores for each non-trivial class. Here,

$$\text{Precision} = \frac{tp}{tp + fp}$$
$$\text{Recall} = \frac{tp}{tp + fn}$$

*w*here tp = true positives, fp = false positives and fn = false negatives. This process is repeated ten times, giving 100 independent test runs. From these we compute the median, 5% and 95% percentiles for both precision and recall.

4 Results

4.1 Data preparation

The section covers the results of two experiments which have been undertaken to gauge the effectiveness of signature-based similarity as an alternative to BLAST alignment for protein classification. A pilot study was carried out using data from the genomes of 8 members of the *Chlamydia* family in which we explore the effect of word length (k) and fragment length (F) on classification accuracy. A subsequent experiment seeks to verify the results of the pilot using the protein sequences of 49 strains of *Staphylococcus aureus*. The genomic datasets used are laid out in Table 1.

Case study	Genomes
Chlamydia	Chlamydiae: *C. muridarum* str Nigg; *C. pecorum* PV3056/3; *C. psittaci* 84/55; *C. trachomatis* A/HAR-13; *C. trachomatis* D/UW-3/CX. Chlamydophilae: *C. abortus* S26/3; *C. caviae* GPIC; *C. felis* Fe/C-56; *C. pneumoniae* CWL029.
S. aureus	NC_002745, NC_002758, NC_002774, NC_002951, NC_002952, NC_002953, NC_003140, NC_003923, NC_005951, NC_006629, NC_007622, NC_007790, NC_007791, NC_007792, NC_007793, NC_007795, NC_009477, NC_009487, NC_009619, NC_009632, NC_009641, NC_009782, NC_010063, NC_010079, NC_012417, NC_013450, NC_013451, NC_013452, NC_013453, NC_016912, NC_016928, NC_016941, NC_016942, NC_017331, NC_017332, NC_017333, NC_017334, NC_017335, NC_017336, NC_017337, NC_017338, NC_017339, NC_017340, NC_017341, NC_017342, NC_017343, NC_017344, NC_017345, NC_017346, NC_017347, NC_017348, NC_017349, NC_017350, NC_017351, NC_017352, NC_017673, NC_017763, NC_018608, NC_020529, NC_020530, NC_020531, NC_020532, NC_020533, NC_020534, NC_020535, NC_020536, NC_020537, NC_020538, NC_020539, NC_020564, NC_020565, NC_020566, NC_020567, NC_020568, NC_021059, NC_021060, NC_021552, NC_021554, NC_021657, NC_021670, NC_022113, NC_022126, NC_022222, NC_022226, NC_022227, NC_022228, NC_022442, NC_022443, NC_022604, NC_022605, NC_022610.

Table 1: Genomes used in experiments

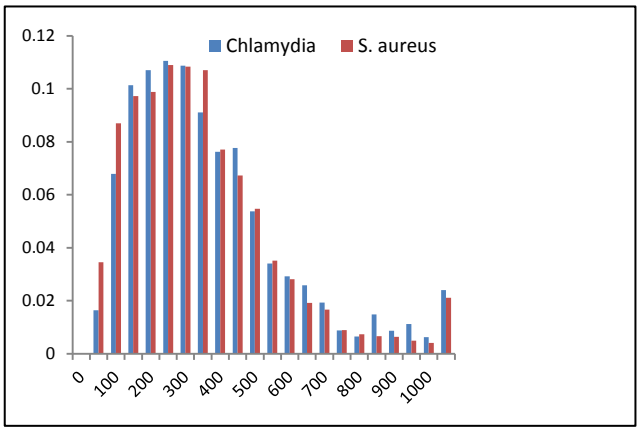


Figure 1: Sequence length

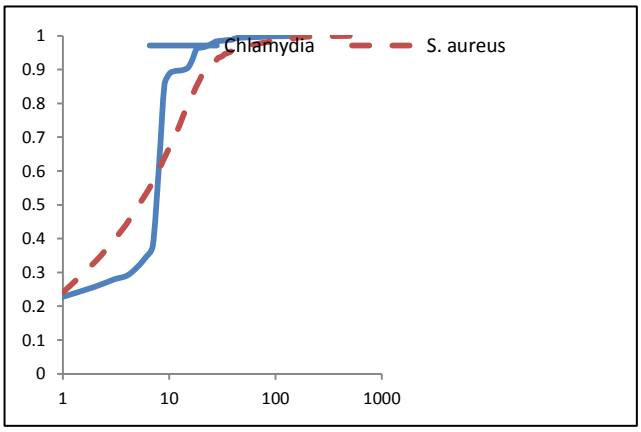


Figure 2: Cluster size (cumulative)

Baseline datasets were constructed via the method specified in Section 3.2, with results summarised in Table 2.

Case study	*Chlamydia*	*S. aureus*
Proteins per genome, avg	1,091	2,531
Proteins, all	8,732	124,019
Proteins, distinct	8,266	34,702
Seq. length, minimum	30	20
Seq. length, median	295	283
Seq. length, maximum	3,432	10,746
MCL clusters, all	1,036	2,882
MCL clusters, $N \geq 2$	800	2,188

Table 2: Protein sequence summary

Although *Staphylococcus* genomes are larger than those of *Chlamydiae*, the distribution of sequence lengths as shown in Figure 1 is comparable between the two datasets, so it is reasonable to apply findings from the pilot study to the larger classification task.

Cluster size distributions are displayed in Figure 2. Although a small number of clusters are quite large, with up to 150 members in the case of *Chlamydia* and 791 in the case of *S. aureus*, the majority of proteins belong to relatively small clusters, with 90% of proteins mapping to clusters with 14 or fewer members in the *Chlamydia* dataset and 20 or fewer members for *S. aureus*. Almost 25% of the proteins in both collections are unique in that BLAST identifies no closely related proteins; these proteins are excluded from the classification experiments reported below as a correct classification is impossible for an isolated protein.

4.2 *Chlamydia* pilot study results

The *Chlamydia* pilot study was conducted to obtain a sense of the effectiveness of signature-based protein classification and also determine the effects of varying window length F and word length k on classification accuracy. K-nearest neighbour classification tests were executed for each $K = 1, 3, 5, 7, 9$ and 11. F ranged over the values 30, 40, 50, 60, 70, 80, 90, 100, 200, 400 and "Full", the latter value representing the case where sequences were not partitioned at all. k ranged over the values 2, 3, 4, 5, 6 and 7. The signature length was held constant at 1024 bits and the k-mer signature bit density was fixed at 21%.

Of the range of K-NN classifiers tested, the best results were obtained with simple 1-nearest neighbour classifiers. Precision and recall degraded rapidly as K increased for all combinations of window length and word length. This may be due in part to the relatively high proportion of proteins which belong to small classes.

Figure 3 and Figure 4 show the respective precision and recall of 1-nearest neighbour classifiers on the *Chlamydia* dataset for a full parameter sweep over the range of k and F. Broadly speaking, the best classification accuracy is obtained from a combination of small word length and small window length. Precision and recall are better than 95% for all classifiers with $k \in \{2, \ldots, 6\}$ and $F = 30$. However, accuracy deteriorates fairly quickly with increasing window length when $k > 3$. While the best-performing combination is $(k = 3, F = 30)$, with precision and recall of 97.6% and 97.5% respectively, all combinations of $k \in \{2, 3\}$ and $F \in \{30, 40, \ldots, 100\}$ give precision and recall of approximately 96% or better.

In view of the small vocabulary generated by k-mers of length 2 or 3, the accuracy achieved with such short words warrants explanation. Consider an evolutionary setting in which proteins are derived from a common ancestor by a chain of random point mutations – insertions, deletions or substitutions. As long as the sequence length is substantially larger than k, the majority of point mutations will disrupt k words (because the location of each mutation is overlapped by that many whole words). Overall disruption caused by point mutations will therefore be minimised by the use of shorter words and this effect offsets the negative impact incurred via reduced vocabulary size.

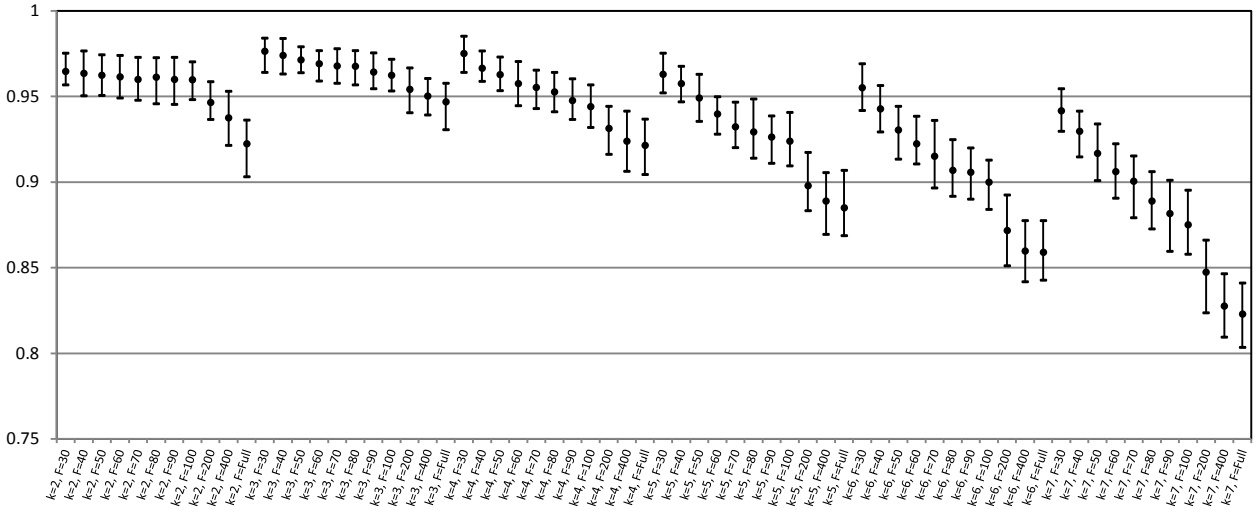


Figure 3: *Chlamydia* pilot study, 1-NN precision. Reading from left to right are groups for word length *k*=2, 3, 4, 5, 6 and 7. Within each group, results are shown for window length *F*=30, 40, 50, 60, 70, 80, 90, 100, 200, 400 and "Full".

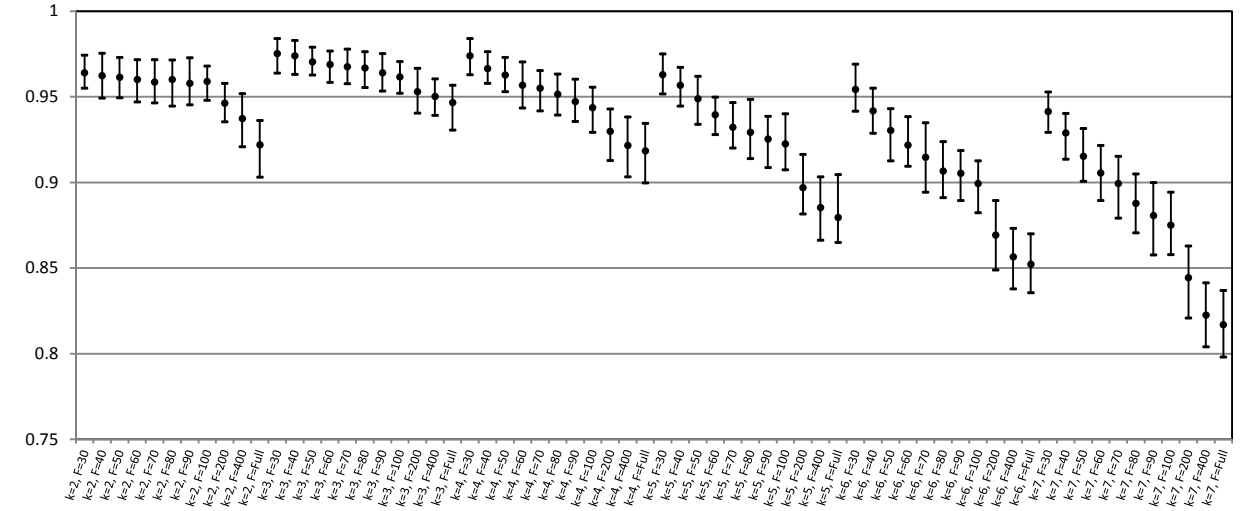


Figure 4: *Chlamydia* pilot study, 1-NN recall. Reading from left to right are groups for word length *k*=2, 3, 4, 5, 6 and 7. Within each group, results are shown for window length *F*=30, 40, 50, 60, 70, 80, 90, 100, 200, 400 and "Full".

In addition, partitioning provides two benefits. Firstly, the partitioning scheme introduces an element of locality by confining matching *k*-mers within a pair of similar fragments rather than allowing them to occupy arbitrary locations in the extended sequences from which fragments are derived. Secondly, since the similarity of a pair of sequences is equal to the best of the pairwise fragment similarities, partitioning makes it possible to recognise conserved sub-sequences embedded within otherwise dissimilar sequences, behaviour which is consistent with that of the traditional BLAST algorithm.

While partitioning delivers more accurate classification, it increases computational complexity as the number of signatures stored in the database is equal to the total number of fragments, which in turn is inversely proportional to the window length. The relatively high accuracy displayed with $k \in \{2,3\}$ and $F = 100$ suggests that good results may be achieved without aggressive partitioning.

4.3 *Staphylococcus* classification results

Informed by the outcomes of the pilot study, we conducted a second experiment using protein sequences obtained from *Staphylococcus aureus* genomes. In this experiment, window length *F* ranged over the values 30, 40, 50, 60, 70, 80, 90, 100, 200, 400 and "Full", while the word length *k* was held constant at 3. Once again, the signature length was set to 1024 bits and the *k*-mer signature bit density was held at 21%. Several K-NN classifiers were tested, with K taking the values 1, 3, 5, 7, 9 and 11.

As was the case with the *Chlamydia* dataset, the best accuracy (both precision and recall) were obtained using 1-NN classifiers, despite the tendency toward larger classes in the *S. aureus* dataset. The results are displayed

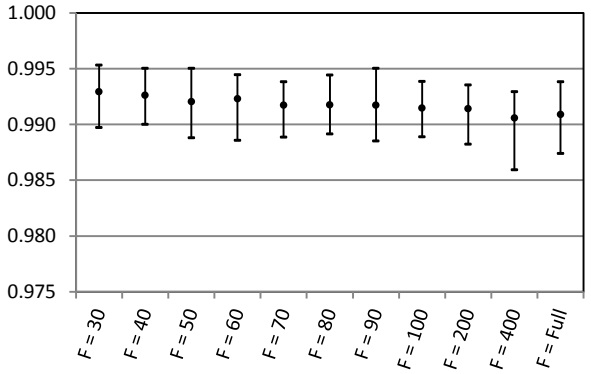


Figure 5: *Staphylococcus aureus* case study, 1-NN precision and recall.

in Figure 5. The precision and recall values differ only in the fourth decimal place, so a single series representing both is shown. With precision and recall in excess of 99% for all classifiers, the results are even better than those obtained in the pilot study. Very little deterioration is observed as window length increases.

5 Conclusions

We have developed a novel alignment-free technique for comparing protein sequences. In around 98% of the cases examined, the resulting classification is identical to that obtained via BLAST, the standard within the bioinformatics community for rapid determination of protein similarity. Moreover, some further investigation of the relationships among the 'erroneous' cases is warranted, as alignment based approaches may not prove robust in the presence of macro-scale structural rearrangements.

Our approach offers the promise of substantial improvements in performance over existing methods, especially given additional optimisation. We are able to compare our fixed length signatures in constant time, whereas alignment based methods take time at least proportional to the length of the sequences. Techniques exist to efficiently search a collection of millions of signatures in milliseconds (Chappell et. al. 2013), so formulating gene similarity in terms of signature similarity opens up a whole field of performance improvements over traditional alignment based methods. Careful tuning of these approaches – tailoring their parameters for large scale genomic datasets – is the subject of on-going work.

6 References

Altschul, S.F., Gish, W., Miller, W., Myers, E.W. & Lipman, D.J. (1990): Basic local alignment search tool. J. Mol. Biol. 215:403-410

Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., and Bourne, P.E. (2000): The Protein Data Bank. Nucleic Acids Research, 28:235-242.

Bingham, E. and Mannila, H. (2001): Random projection in dimensionality reduction: applications to image and text data. Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2001), August 26-29, San Francisco, CA, USA, pp. 245-250.

Broder, A. (1997): On the resemblance and containment of documents. Proceedings of the Compression and Complexity of Sequences, page 21.

Chappell, T., Geva, S., Nguyen, A. and Zuccon, G (2013): Efficient top-k retrieval with signatures. ADCS'13

Enright, A.J., Van Dongen, S. and Ouzounis, C.A. (2002): An efficient algorithm for large-scale detection of protein families. Nucleic Acids Research 30(7):1575-1584.

Ewens, W.J. and Grant, G. (2001): Statistical methods in bioinformatics: an introduction. New York, Springer-Verlag.

Geva, S., De Vries, C.M. (2011): TOPSIG : Topology Preserving Document Signatures. In Conference on Information and Knowledge Management 2011, 24-28 October 2011, Glasgow, Scotland.

Harlow, T.J., Gogarten, J.P. and Ragan, M.A. (2004): A hybrid clustering approach to recognition of protein families in 114 microbial genomes. BMC Bioinformatics, 5(45).

Johnson, W.B. and Lindenstrauss, J. (1984): Extensions of Lipschitz mappings into a Hilbert space. Contemporary mathematics, 26(189-206).

Paul, G. and Maitra, S. (2011): RC4 Stream Cipher and Its Variants. Taylor and Francis.

Reinert, G., Chew, D., Sun, F. and Waterman, M.S. (2009): Alignment-free sequence comparison (I): statistics and power. Journal of Computational Biology, 16(12): 1615-1634.

Sadowski, C. and Levin, G. (2007): SimHash: Hash-based Similarity Detection. http://simhash.googlecode.com/svn/trunk/paper/SimHashWithBib.pdf.

Sahlgren, M. (2005): An introduction to random indexing. In TKE 2005.

Skippington E, Ragan MA (2011) Lateral genetic transfer and the construction of genetic exchange communities. FEMS Microbiol Rev 35: 707-735

Slaney, M., Casey, M.(2008): Locality-Sensitive Hashing for Finding Nearest Neighbors Signal Processing Magazine. IEEE In Signal Processing Magazine, IEEE, Vol. 25, No. 2. (March 2008), pp. 128-131.

Thompson, JD., Higgins, DG and Gibson, TJ (1994): CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic Acids Res. 22(22):4673-80.

Van Dongen, S. (2000): Graph Clustering by Flow Simulation. PhD thesis. University of Utrecht.

Vasuki, V., Cohen, T. (2010): Reflective random indexing for semi-automatic indexing of the biomedical literature. J Biomed Inform. 2010 Oct;43(5):694-700.

A Case Study of Utilising Concept Knowledge in a Topic Specific Document Collection

Gavin Shaw & Richi Nayak
Data Sciences Discipline, Science & Engineering Faculty
Queensland University of Technology
2 George St, Brisbane, 4000, Queensland, Australia
gavin.shaw@qut.edu.au; r.nayak@qut.edu.au

Abstract

The use of 'topic' concepts has shown improved search performance, given a query, by bringing together relevant documents which use different terms to describe a higher level concept. In this paper, we propose a method for discovering and utilizing concepts in indexing and search for a domain specific document collection being utilized in industry. This approach differs from others in that we only collect focused concepts to build the concept space and that instead of turning a user's query into a concept based query, we experiment with different techniques of combining the original query with a concept query. We apply the proposed approach to a real-world document collection and the results show that in this scenario the use of concept knowledge at index and search can improve the relevancy of results.

Keywords: Text Mining, Document Concepts, Term to Concept, Concept Search, Case Study, Wikipedia.

1 Introduction

Text mining is a critical function used to discover documents that are related to a user's query of interest. This is usually facilitated by searching an index via a query and matching the terms from a query with those contained within documents. Such searching and mining activities can be enhanced through a variety of techniques at both ends of the process; document indexing time and/or query time. Given that documents can contain a large volume of text and many differing terms or be short with much fewer unique terms; all within the same collection; traditional term searching and matching can yield poor results [Fang, 2004; Lv, 2011]. The keyword-based text matching methods can miss documents that are relevant but use different terms.

One popular approach to improving performance of simple text matching approaches is to attempt to discover, add and utilize concept level knowledge within the document set and user queries. The identified concepts operate at a higher information space and reduce the number of 'terms' associated with a given document. A key advantage to the use of concepts, aside from providing a set vocabulary is that they can result in documents being brought together under one concept even when they use different terms/words to describe the same concept. Furthermore, if a 'concept only' search can be conducted, the costs of handling user queries can be reduced, although there may be a trade-off in the quality of the results due to the more 'generalised' nature of concepts.

This paper examines and evaluates the improving of document search retrieval for an industry dataset that has the potential to play an important role within a business unit of a much larger company. The document collection was gathered to help the unit undertake a foresight and future options development and help analysts looking for information relevant to strategic risks and/or structural changes. Thus ensuring highly relevant results are returned and ranked highly is key to supporting these activities.

There are several key points and contributions in our work;

1. Utilizing a publicly available, general purpose (eg. not domain specific) information source, Wikipedia, to build a domain topic specific concept space instead of purpose building a custom concept space; which can be resource demanding.
2. In using said general purpose knowledge source, instead of using all or a random selection of its content; to generate term to concept links, we use a specific portion that is identified as relevant to the particular application domain and control how far we crawl to build the concept space and associated term to concept links.
3. Instead of taking a user's query and converting it to a concept only query; via the built concept space model; such as that proposed in [Egozi, 2011]; we create a hybrid query by combining the traditional term/phrase based search with a concept based search. Similar to the traditional approaches the concepts can be discovered by mapping the user's query terms to concepts via the concept model, but can also be drawn from the documents returned as results to an initial query formed from the user query. Further, these two approaches to building the concept component of the hybrid query can be used together or separately.
4. For this work we test these ideas on a real world industry document collection gathered by a major business operating in the financial sector of Australia and evaluate the performance of such a concept space and hybrid query approach.

To our knowledge the specific combination of how the concept space was built and utilized is new, further its evaluation against industry data shows its potential for

use outside research, given that previous proposed approaches; such as those in [Milne, 2008; Medelyan, 2008]; often do not get evaluated on a dataset from the commercial world, should make the approach of interest to industry.

The rest of the paper is as follows; Section 2 reviews related background materials, while Section 3 outlines the proposed method for building a concept space and the utilizing it at index and query time. Section 4 presents our experiments, results and evaluations of the proposed approach. Section 5 concludes our work and the paper.

2 Background & Related Work

The key challenges of using concepts in text mining is identifying a list or vocabulary of concepts, determining which concepts a document is related/relevant to and then taking advantage of this extra information when a query is submitted. The first challenge, identifying concepts, has received much attention [Hou, 2013; Egozi, 2011, Huang, 2009; Medelyan, 2008; Mihalcea, 2007]. A common and popular source for determining concepts is Wikipedia (although WordNet and BabelNet have also been considered), where each article represents a concept [Hou, 2013; Huang, 2009; Medelyan, 2008; Mihalcea, 2007] and text that refers to a given article becomes terms/phrases that represent that concept. With approaches that use Wikipedia, either the entire articles collection is used to build a concept list, or a random subset is used. This approach is fine for document collections that cover multiple topic areas, but when you have a document collection focused on a much smaller topic area, making concepts from unrelated areas available for possible use is likely to lead to issues with performance at query time.

However, a purpose built concept space, for a specific domain/topic can be impractical depending on the scope and the level of detail desired and can involve the need for domain experts to build such a space. Thus it is of interest to determine and discover whether such a focused concept space can be built from an existing, much broader space without being 'polluted' by unrelated domains. There does not appear to be much existing work focusing on this.

For the second challenge, works such as [Medelyan, 2008; Mihalcea, 2007; Milne, 2008] outline approaches to determining how to relate documents to concepts. They include identification of term to concept mapping [Medelyan, 2008]; selection of key text to link [Mihalcea, 2007; Milne, 2008] and identification of the most relevant concept to a piece of key text when there are multiple possibilities [Medelyan, 2008; Milne, 2008].

The final challenge, using concepts at query time, usually involves some form of query enhancement or expansion. Recent work in [Carpineto, 2012] looks at many automatic query expansion applications; however none of them involved dealing with concepts. Other works identified that present concept oriented works do not explicitly state how such knowledge would be used at query time to improve result performance. Work in [Egozi, 2011] outlines how a query is converted into an Explicit Semantic Analysis concept vector which is used to find the best matching documents. This approach however results in the search becoming concept only, which can work in some applications, but not all domains, applications and queries. The weak point with an approach, such as that in [Egozi, 2011]; or [Huang, 2009; Medelyan, 2008; Mihalcea, 2007; Milne, 2008] if concept only querying is used; is that when a user's initial query terms cannot be successfully mapped to one or more concepts for search then the query will fail. Conversely if the initial terms never take advantage of being mapped to concepts, then this extra, higher level knowledge is unused. Further still, each approach with their limitations, are unlikely to always be able to substitute for the other.

Given these limitations and the importance of this document collection to the industry partner, it is essential to ensure that the proposed approach be able to deal with the widest possible range of queries and successfully return relevant results. Thus in order to achieve this we will combine the traditional term/phrase searching with a concept based searching to have a hybrid query search approach to finding relevant documents.

3 Proposed Approach of Concept Space Generation and Querying

In this section we outline our proposed approach to enhance text mining through concept discovery & mapping, and query searching. There are three main components; discovery of relevant concepts and their mapping to terms (eg. building the concept space); document concept discovery and selection; and finally, query concept discovery & utilization for search.

We start this section by first introducing important definitions related to our proposed methods, followed by presenting each of the three main components in turn.

3.1 Definitions

The following are key definitions used in our proposed approach.

Definition 1 – Important Term: An important term is a single word or n-gram (composed of successive words) found within a document, from the collection that matches exactly with one or more hyperlink text entries, obtained from the term to concept mappings discovered via Wikipedia. The important term thus maps to one or more Wikipedia articles, which using their titles, represent concepts potentially relevant to the document.

Definition 2 – Concept: A concept is generated by the presence of a Wikipedia article page (not counting disambiguation, red links, category, 'etc' pages) discovered during the crawl of Wikipedia. The concept is represented by the title of said Wikipedia article page. Thus, a Wikipedia article can be considered as a concept, identified by its title.

Definition 3 – Context Concepts: A given document usually contains many terms/n-grams that potentially link to multiple concepts. However, there are a handful of terms/n-grams that only link to one concept each. The concepts which have no competing candidates for the term/n-gram that mapped to them can be known as context concepts for the given document. These particular

concepts help to describe the document and set the context in which it is present; hence the name. Context concepts can be used to help determine which concept, from a list of candidates, is the most relevant/related to the given document for a given term/n-gram.

3.2 Concept Space Generation

In order to be able to enhance the document index and user queries with concepts, it is first necessary to discover concepts that would be relevant to the topic(s) covered by the document collection and the important terms that are used to refer to them.

One difference between our use of Wikipedia and that of other works is that we use a targeted subset of the Wikipedia collection as opposed to the entire collection or a random sample that has been used in previous works. In our case the targeted starting point for our subset of Wikipedia is the category page for 'Finance'. We have taken this approach due to the nature of our document collection and industry partner. They are interested in the financial sector and thus tagging documents with concepts from other categories/topic areas, such as Science Fiction, Cooking, Anime etc, is unlikely to be relevant. Thus using the whole Wikipedia collection for concept discovery would lead to the inclusion of concepts which have none or very little relevance to the area of finance and the expected user queries.

Similarly, random selection of Wikipedia articles is also likely to lead to a similar situation in which irrelevant concepts are linked to the documents. Further, the use of a set of random articles may be worse as the selected set may not include any articles and hence concepts, in the topic area(s) related to finance.

The following figures (1, 2 & 3) demonstrate at a high level the approach to build the topic specific concept space, with associated term/phrase to concept mappings, from the full Wikipedia resource.

Figure 1 outlines the overall Wikipedia crawling process, where we start at a specific page and determine whether it is a category page (Figure 2) or an article page (Figure 3) and process it accordingly. After all desired pages have been processed we have a full listing of term to concept mappings for which we calculate a commonness score, using Equation 1. Finally, the term list, concept list and term-concept list are brought together to build the concept space that will be used during indexing and querying/searching.

The commonness score is determined via the following:

$$Commonness(t,c) = \frac{count(t \mid c)}{count(t)} \qquad \text{Eq. 1}$$

where t is the term/n-gram, c is the concept, $count(t|c)$ is the number of times text t was discovered to link to concept c and $count(t)$ is the number of times term/n-gram t was discovered to link to a concept, including c.

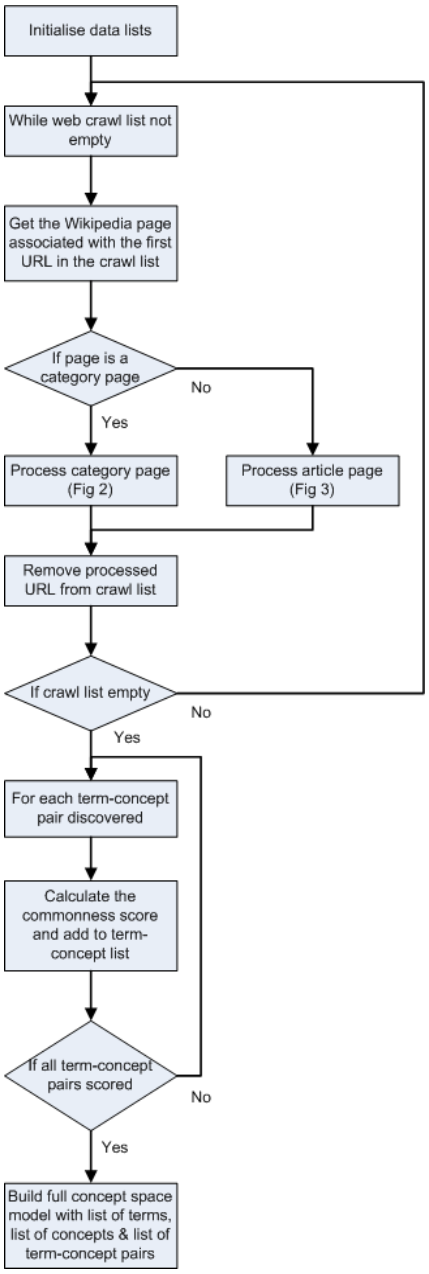


Figure 1. Overall concept space generation approach.

Figure 2 outlines the approach with processing category pages from Wikipedia. These pages do not form concepts, but instead provide a list of subcategories and articles (which are concepts) that fall under the category. For category pages the process is fairly straight forward, the current crawl depth of the page from the staring page in Wikipedia (eg. the number of hyperlink hopes) is compared against the maximum distance allowed. If it is less then all of the subcategories present are added to the crawl list with a crawl depth one greater than the crawl depth of the current category page. If the current crawl depth equals or is greater than the maximum allowed then the subcategories are not to be processed and the links to them are not added to the crawl list. Finally all of the links to article pages (eg. concepts) are added to the crawl list for processing.

We place a limit on the crawl depth to stop the process of generating the concept space from attempting to include a large portion of Wikipedia, which would then introduce unrelated concepts into the model. The

maximum crawl depth keeps the concepts collected more closely related to the original starting point and thus the specific topic of the domain remains relevant to this document collection.

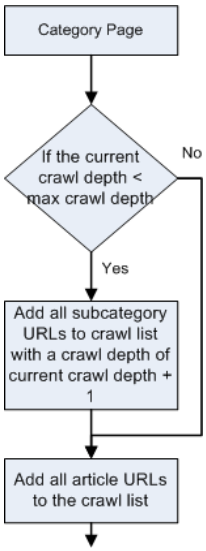


Figure 2. High level flow for processing category pages from Wikipedia.

Figure 3 outlines the approach for processing article pages from Wikipedia, which will form the basis for concepts in our concept space. For each article page all of the hyperlink text and their target URLs are extracted from the article's body. The last part of the target URL specifies the Wikipedia page and corresponds to the title of said page. This thus becomes the name of the concept that will be used within our concept space. With this we are able to build term/phrase to concept mappings. To support the concept space being built we maintain a list of terms, a list of concepts and a list of term-concept pairs, with every entry in these lists having an associated occurrence frequency. The text to URL titles extracted from each article page is added to these lists to build the concept space. Once all of the extracted hyperlink text and target URL pairs are processed, we have finished with the article page.

For the list of concepts it is important to note that it is not just the name of the concept (article linked to), but also a complete list of term/phrase to concept mappings of the hyperlinks from within this target article is included.

3.3 Document Concept Mapping

Once the concept generation via Wikipedia is complete, it is then necessary to index the document collection with the proposed enhancements. In our approach to this document collection, we implemented two main enhancements over the basic standard of indexing a document's title and textual contents. The enhanced index includes separate entries for a document's important terms and its associated concepts.

As the document concept mapping is not the primary focus of our proposed approach (rather the concept space generation and query enhancement are) and to save space, we do not go into great detail. Suffice to say, the approach involved takes each document, tokenises its content and then process each term in the following way. If the term is not a stopword and is found in the term to concept list previously developed then it becomes an important term and a set of candidate concepts for that term for the current document are identified. This is also done for n-grams up to the desired size. After a document's text has been processed and mappings to potential concepts identified, then the most relevant concept for each important term is identified. If an important term only maps to a single candidate concept then that concept is associated with the document and used as part of the context for the document.

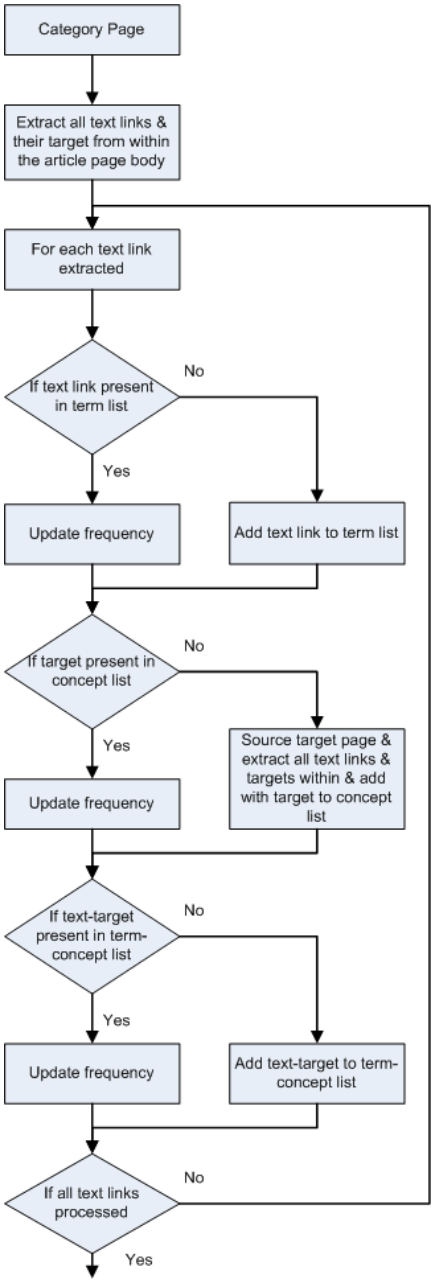


Figure 3. High level flow for processing article pages from Wikipedia.

For important terms that have more than one candidate concept we calculate the relevance scores of each candidate concept to the document for that term, using Equations 2 & 3 and the context concepts. The candidate concept with the highest relevance score for the important term is the one associated with the document. This

calculation and association happens for each important term instance.

Finally, once the most relevant concept for each important term is determined the top-n important terms and the top-n concepts; measured by frequency; are identified and added to an associated field for the document to be added to the index. The full listing of important terms and concepts are maintained in separate fields, giving the option of performing searches against the top terms or concepts or the full set of terms or concepts for each document. This supports greater searching options and allows us to test the performance of using both a document's full concept set and separately a document's set of top (most frequent) concepts.

As required by the approach taken to map documents to concepts prior to indexing, the following equations are used to determine the relevance of a candidate concept for a given term/n-gram for a given document.

$$SIM_{C1,C2} = 1 - \frac{(\max(\log(|C1|), \log(|C2|))) - (\log(|C1 \cap C2|))}{N - (\min(\log(|C1|), \log(|C2|)))} \quad \text{Eq. 2}$$

where *C1* and *C2* are concepts for which their similarity is being calculated, *N* is the total number of concepts extracted from Wikipedia (eg. present in the term to concept map) and $|C1|$ & $|C2|$ represents the number of links to other concepts from *C1* or *C2* and $|C1 \cap C2|$ represents the number of links to other concepts that *C1* & *C2* have in common.

$$RelevanceScore(n,t) = \frac{\sum_{c \in C} SIM_{t,c}}{|C|} \times Commonness(n,t) \quad \text{Eq. 3}$$

Where $c \in C$ are the context concepts for the current document, *T* is the candidate concept which the relevance score is being calculated and *n* is the important term/n-gram that the candidate concept is related to. Further information on these equations is available in [Medelyan, 2008].

3.4 Document Index Querying & Searching

At the other end of text mining is querying by end users. In order to make use of the enhancements introduced during the indexing of the document collection it is necessary to enhance the query generation process. In our proposed approach, instead of converting a user's query into a pure concept query, we build an enhanced query which contains the initial term/phrase based query for text matching, but also contains, where available concepts identified as being relevant (using approaches to be described in this section), for matching with the concept information stored within the index. While this approach is based on the idea of query expansion, it goes beyond simply trying to identifying extra suitable terms to add to the query. Firstly, both terms and concepts will be included in a single query through the use of sub-queries and each sub-query component can be targeted against different fields within the document index.

Figure 4 shows our proposed approach for building hybrid term-concept queries that will be used to search on the document index. We have two methods for enhancing the query with concepts; from the initial query (QC) and from the initial results (RC).

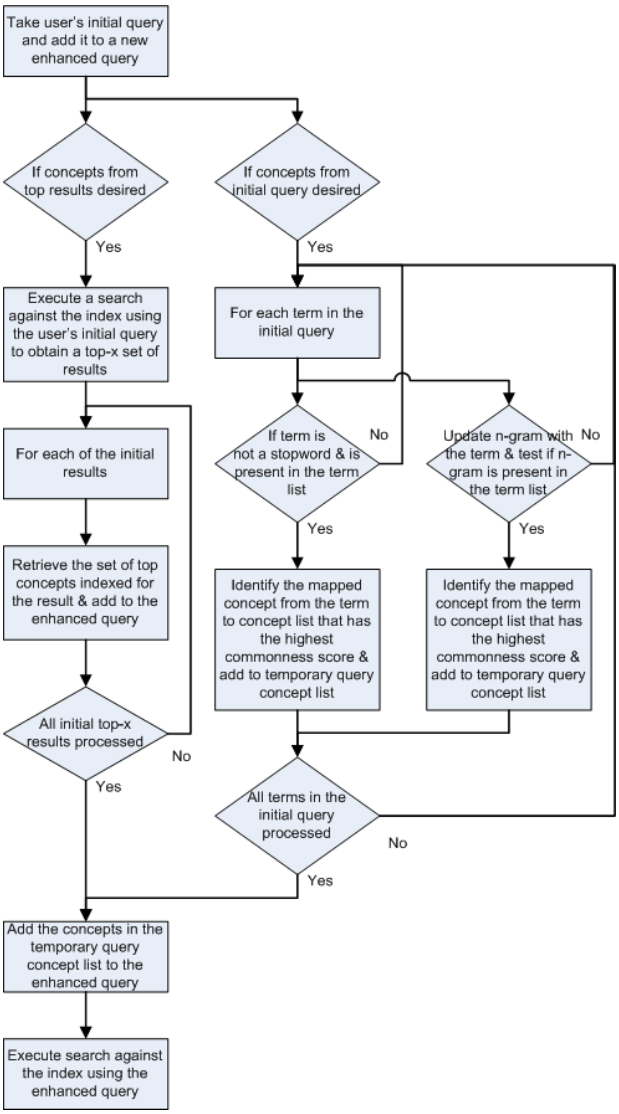


Figure 4. Overall query enhancement approach.

The first proposed method is to use the query provided by the user to obtain an initial set of search results. From these initial results we take the top-n and re-query the index to retrieve the top concepts associated with each returned result. The set of concepts from each result then forms a complete sub query of the final hybrid query. This method will work best when the initial term based query provided is of good quality and the top-n results returned from a search using this query are relevant. This helps to ensure that the top concepts utilised are relevant to the original initial query.

The second proposed method to enhancing the query with concepts is to take the initial query and in a manner similar to how we discovered and mapped concepts to a document, we discover concepts relevant to the terms and phrases present in the query. Thus for each term & n-gram we test to see whether it exists in the list of important terms that is within the concept space. If it is present, then that important term maps to at least one concept. If it maps to more than one concept we select the concept with the highest commonness score and these concepts are brought together to form a complete sub query of the final hybrid query.

The commonness score is chosen as the selector of which concept is relevant to the query because it would

be very unlikely that an accurate relevance score between a query and a concept could be calculated as there is a high probability that there would not be a suitable set of context concepts. The context concept set is required in order to be able to measure relevance if we treat the query like a document. The commonness score represents what proportion of the time; a given term or n-gram is related to a particular concept within Wikipedia.

These two approaches to creating a hybrid query can be used separately or together as each generates one or more sub components for the final enhanced query and do not depend on each other in any form. Thus there are three different combinations; 1) from the initial query, 2) from the initial results and 3) from both the initial query & initial results; that our hybrid query can obtain its concepts from. In the experiments that follow, we undertake tests on all three concept source configurations to get a measure of how well they perform in obtaining and adding relevant concepts to the hybrid query to improve search result relevance.

4 Experiments

In this section we describe the experiments undertaken to test the performance of our proposed document collection and query enhancement approach. We provide a brief summary of the real life industry dataset used along with key information, the evaluation measures used to assess performance and finally the actual results of our experiments.

4.1 Information on Datasets

The dataset used in this case study and experiments was provided by the project's industry partner, which we labelled IPDC randomly.

This dataset was built by the project's industry partner using an automated system to take in a list of starting website addresses and collect a set of pages reached via hyperlinks from the starting point. All together there were approximately 120 starting URL's supplied to the web crawler. At the end of the document collection process using these starting URL's a total of 467,070 documents had been gathered. The majority of these documents are web pages, but the collection also included pdf and word files.

Before we generated any indexes of this set of documents we undertook some basic pre-processing to identify and remove as much web page mark-up as possible. This included the identification and removal of contents such as HTML tags and Java script.

For the IPDC set, we created a small set of finance oriented queries which were used against this set. Due to the nature of this document collection we had no relevance information to draw upon to judge the performance of said queries. Thus it was necessary to manually review the results and make relevance judgments. Due to limitations in time and resources, we could only perform this manual judgment for a small number of queries and only on the results returned.

4.2 Evaluation Measures

Ideally, in such experiments we would measure at a minimum the precision and recall performance of each of the queries at various ranks. From these measurements we would then expand to other measures such as average precision and mean average precision (MAP) for each query and then an overall value for the set of queries.

For the IPDC dataset, we are only able to calculate the precision of each query and an overall average precision for the set of queries. We are unable to determine the recall result of our queries as we did not have the resources to thoroughly assess the dataset to find 'all' relevant documents to a given query. Further, the industry partner did not have the resources to undertake activity to build such a baseline to identify documents relevant to a base set of queries (in the manner of TREC for example).

We also calculated the overall average result overlap between the baseline and each experimental enhancement configuration, allowing us to discover how many results each enhanced approach had in common with the un-enhanced approach. Along with the overlap, we also measured the Kendall correlation between each enhanced approach and the baseline to gain an idea of whether there was strong, or any consistency in the actual ranking order of the documents returned in response to queries.

For the precision calculations, we manually assessed the top-20 documents returned for each query in each experimental configuration. This limit was chosen due to the cost in resources for performing manual assessment and that the top-20 results often correspond to the first page from search engines. For the overlap and correlation calculations we measured this using the top-100 documents returned for each query in each experimental configuration.

Our baseline query approach that we compare our hybrid queries against is the straight forward, simple term based query. The query terms that form each complete query in the baseline, also serve as the initial query from which the hybrid queries are built. All baseline queries search against the document's textual content and do not take advantage of any extra knowledge available by having concepts mapped to the documents.

4.3 Experimental Results

For the results in the following tables the following descriptions apply, indicating what data within the document index is being searched against for the concept component of the hybrid query;

- QC – query concepts, where concepts are discovered from the initial query and used to build the hybrid query
- RC – result concepts, where the initial query is executed and the top-x results have their top concepts extracted and used to build the hybrid query
- T3 – top-3, the top-n number of initial resulting documents whose top concepts are extracted to build the hybrid query
- B – body, the document's textual contents
- FC – full concepts, the document's complete set of concepts
- TC – top concepts, the document's set of top 10 concepts

Table 1. Overall average precision and '%' difference against baseline.

Query Exp. Config.	Overall Average Precision @ Top-20	% Difference with Baseline
Baseline	0.583	
QC_B	0.7	20.07
QC_FC	0.467	-19.9
QC_TC	0.533	-8.58
RC_T3_B	0.5	-14.24
RC_T3_FC	0.667	14.41
RC_T3_TC	**0.783**	**34.31**
RC_T3_QC_B	0.717	22.98
RC_T3_QC_FC	0.65	11.49
RC_T3_QC_TC	0.767	31.56

Table 1shows the results of our experimentation on the IPDC corpus where we executed a set of queries and manually judged the relevancy of the results returned, due to the lack of known ground truth for the corpus. The baseline query configuration; which is term based; achieved an overall average precision of 0.583 at the top-20 rank. Measured against this, were 9 experimental approaches which obtained/discovered concepts and utilized said concept knowledge in different ways.

The poor performance of two of the QC based approaches; where the concepts used in the hybrid query come from the initial query itself; comes down to two possibilities, or a combination of. First that there could have been a mismatch between the concepts selected at query time (via the commonness score) and those selected at index time (via relevance score). The second possibility is that our approach was unable to find mappings to concepts for the terms in the initial query; eg. they were not 'important terms'.

The two configurations with the greatest improvement; of over 30%; both utilise concepts extracted from an initial set of results discovered via the initial term query and then utilise said concepts against the top concepts field in the index. Thus when the initial query returns relevant results, there is a high probability that relevant concepts can be extracted from these results and added to the hybrid query, yielding further relevant results. Further, the best performance is obtained when we take the concept component of our hybrid query and use it to search against the indexed top concepts field, rather than the indexed full concepts field. This demonstrates the need for ensuring that the concepts associated with documents are strongly relevant to the document and that keeping a full list of concepts introduces weakly relevant concepts that have the potential to limit the quality of results. Further, it also demonstrates that the use of frequency; how many times a concept is mapped to a document through its various important terms; is a viable method for determining the most relevant, and hence the top concepts for a document.

We also tested using the concepts in the hybrid query and searching against the textual contents of the documents; eg. like we do with terms. The results do show an improvement is possible, indicating that in some cases the concepts themselves are present within the document's textual content; body; as terms and thus the hybrid query acts more like a term expanded query, rather than a term-concept query. However, a decrease in performance is also possible if the concepts are not present in the document's content then their similarity score would tend to decrease. Further, should the concepts be present in documents that are not relevant and do not feature the initial terms, they may be promoted up the result rankings allowing them to contend for being included in the top-20.

Table 2. Overall average result overlap and Kendall correlation against baseline.

Query Exp. Config.	Overall Average Overlap with Baseline @Top-100	Overall Average Kendall Correlation with Baseline @Top-100
QC_B	90.67	0.609
QC_FC	32.67	-0.103
QC_TC	41	-0.014
RC_T3_B	40.67	-0.083
RC_T3_FC	25.33	-0.069
RC_T3_TC	26.33	-0.079
RC_T3_QC_B	34.33	-0.135
RC_T3_QC_FC	**17**	-0.059
RC_T3_QC_TC	18.67	0.008

As can be seen in Table 2 the different methods obtaining concepts for inclusion into the hybrid query and their utilisation have differing extents of overlap with the baseline method; straightforward term based querying. The overlap ranges from a high of 90.67 to a low of 17 results in common. The fact that the overlap is not 100% makes it clear that the hybrid queries are making the determination that other documents; that the baseline discards; are relevant and should be returned. Thus the hybrid queries are not simply reordering the top-100 results. The implication of this is that a term based query here finds one set of documents, while the hybrid query finds a different set from the document collection, offering an approach for a user to find potentially relevant results that their original query would not find.

Result overlap with the baseline set is minimised when the concepts in the hybrid query are utilised against the full concepts of the documents. This happens as this field maximises the probability of matching the query concepts to document concepts; as all concepts deemed relevant to the document are present. The overlap when using the top concepts field is generally only marginally higher, indicating that even the shorter list of document concepts can be enough to allow the hybrid query to have a high probability of matching concepts. Thus both of these offer a good method for finding alternative results. The experimental approach in Table 1 that had the highest overall average precision; RC_T3_TC; only has

approximately a quarter of its results in common with the baseline. Thus not only is it ranking a larger number of relevant documents in the top-20; it is also finding documents that the baseline does not discover, opening up new sources to the user. The experiments in which the document's textual content is used for concept searching have the highest result overlap. Again, this can come down to two possibilities; first that the concepts are present in the textual content of the same documents that the terms appear in (thus reinforcing their perceived relevance) or second, that the concepts cannot be found in the textual content of most documents in the collection and the hybrid query essentially defaults to a term based query, identical to that used in the baseline.

The correlation scores also show the difference between the baseline and enhanced methods. With the exception of one enhanced method, all of the correlation scores are closer to zero, indicating a degree of independence between the baseline and enhanced methods. This indicates that the ranking order of the results from the hybrid query is different and that the concept component plays an important role in the ranking order. The exception to this is the hybrid query configuration where the concepts are obtained from the initial query terms and searched for against the document's textual content; body. Here a high correlation score showing strong agreement was obtained. This would be caused by this particular approach having difficulties in obtaining concepts from the initial terms; eg. a lack of important terms; and then locating them within the body of a document; either they are present in the same documents as the terms, or are not present at all in the majority of the documents in the collection.

5 Conclusions

In this paper we have applied the idea of using concepts, derived from a general source to build a topic specific concept space and a hybrid based querying method to then search upon an associated document index to take advantage of any topic specific concepts identified as relevant to documents within. We then applied our proposed approach to a real world industry document collection that the project's industry partner built to assist their business operations.

In our experiments we tested three methods for identifying concepts to add to the query and then tested these combinations by using the query concepts against three different fields (content, full concepts and top concepts) within the index. This helped us to discover which method of obtaining concepts and how we should use them for searching on the index. We achieved a 34% improvement in result relevancy through the use of extracting concepts from the top-3 results returned by the user's initial term-based query and using those concepts to search against the top-10 concepts associated with the documents.

Our application of the proposed methods and experiments demonstrate that there is potential for a topic specific concept space to be built from a general knowledge source like Wikipedia and that hybrid queries, composed of terms & concepts can return more relevant results than just term based query. We also demonstrate that this method can work on industry document collections via our experiments which focused on a real world collection.

6 Acknowledgement

The content presented in this paper is part of an ongoing cooperative study between Queensland University of Technology and an Industry Partner, with sponsorship from the Cooperative Research Centre for Smart Services (CRC-SS). The authors wish to acknowledge CRC-SS for funding this work and the Industry Partner for providing data. The views presented in this paper are of the authors and not necessarily the views of the organizations.

7 References

Carpineto, C. & Romano, G. (2012): A Survey of Automatic Query Expansion in Information Retrieval. *ACM Comput. Surv.,* 44**,** 1-50.

Egozi, O., Markovitch, S. & Gabrilovich, E. (2011): Concept-Based Information Retrieval Using Explicit Semantic Analysis. *ACM Transactions on Information Systems (TOIS),* 29**,** 1-34.

Fang, H., Tao, T. & Zhai, C. (2004): A formal study of information retrieval heuristics. *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval.* Sheffield, United Kingdom: ACM.

Hou, J. & Nayak, R. (2013): A concept-based retrieval method for entity-oriented search. *Proceedings of the 11th Australasian Data Mining Conference (AusDM 2013)*. Canberra, Australia.

Huang, A., Milne, D., Frank, E. & Witten, I. (2009): Clustering Documents Using a Wikipedia-Based Concept Representation. *In:* Theeramunkong, T., Kijsirikul, B., Cercone, N. & Ho, T.-B. (eds.) *Advances in Knowledge Discovery and Data Mining.* Springer Berlin Heidelberg.

Lv, Y. & Zhai, C. (2011): Lower-bounding term frequency normalization. *Proceedings of the 20th ACM international conference on Information and knowledge management.* Glasgow, Scotland, UK: ACM.

Markó, K., Hahn, U., Schulz, S., Daumke, P. & Nohama, P. (2004): Interlingual Indexing across Different Languages. *7th International Conference on Computer-Assisted Information Retrieval (Recherche d'Information et ses Applications) - RIAO*. France.

Medelyan, O., Witten, I. H. & Milne, D. (2008): Topic Indexing with Wikipedia. *Proceedings of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy*. AAAI Press.

Mihalcea, R. & Csomai, A. (2007): Wikify!: linking documents to encyclopedic knowledge. *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management.* Lisbon, Portugal: ACM.

Milne, D. & Witten, I. H. (2008): Learning to link with wikipedia. *Proceedings of the 17th ACM conference on Information and knowledge management.* Napa Valley, California, USA: ACM.

Decreasing Uncertainty for Improvement of Relevancy Prediction

Libiao Zhang[1,2] **Yuefeng Li**[1,3] **Moch Arif Bijaksana**[1,4]

[1]School of Electrical Engineering and Computer Science,
Queensland University of Technology, Brisbane, QLD 4001, Australia.

[2]L39.zhang@hdr.qut.edu.au [3]y2.li@qut.edu.au [4]arifbijaksana@gmail.com

Abstract

As one of the key techniques of Information Retrieval (IR) and Information Filtering (IF), Text Classification focuses on classifying textual documents into predefined categories through relative classifiers learned from labelled or unlabelled training samples. Binary text classifiers is the main branch of Text Classification, involving the relevance prediction of documents to users or categories. However, the current binary text classifiers cannot clearly describe the difference between relevant and irrelevant information because of knowledge uncertainty owing to the imperfection of the knowledge mining techniques and the limitation of feature selection methods. This paper proposes a relevance prediction model by decreasing the relative uncertainty to improve the performance of binary text classification. It tries to form and train the decision boundary through partitioning the training samples into three regions (the positive, boundary and negative regions) to assure the discrimination of extracted knowledge for describing relevant and irrelevant information. It then produces six decision rules corresponding with six different situations of the related objects to help make relevance predications for those objects. A large number of experiments have been conducted on two standard datasets including RCV1 and Reuters21578. The experiment results show that the proposed model has significantly improved the performance of binary text classification, thus proved to be effective and promising.

Keywords: Relevance prediction, Text classification, Uncertainty, Decision boundary, Decision rule.

1 Introduction

With the explosive growth of electronic textual documents, text analysis and classification is getting increasingly important and attracting extensive attention in the similar research fields in recent years. Relevance prediction is a big research issue [17, 21] for text analysis and classification, which focuses on predicting a document's relevance to a query, a category that a user concerns. Text classification is the process of classifying an incoming stream of textual documents into predefined categories through the classifiers learned from the training samples, labelled or unlabelled. Different kinds of text classification technologies have been invented and developed in different level and utilized to automatically classify the textual documents, such as k-Nearest Neighbors [7], Support Vector Machines [30], Naive Bayes [14], Rocchio Similarity [27] and rule-based methods. With the continuous improvement of text classification technology, its application has been prevalent in the real world and many applications of text classification have been developed in recent years such as the classification of news stories, e-mail message, customer reviews, academic papers or medical records, filtering of spam and porn, and the application in Bioinformatics and customer service automation [30]. A binary text classifier can be used to help gain relevant information to a category or a user's interest, which assigns one of two predefined classes (e.g., relevant category or irrelevant category) to incoming documents since relevance is a single class problem [12]. The most common solution to the multi-class problem is to decompose it into several independent binary classifiers.

A binary classifier usually defines a decision boundary to group documents into two categories: the relevant and irrelevant categories. However, the decision boundary contains a lot of uncertain information because of a number of reasons such as noise of knowledge mining and deficient strategy of feature extraction for text classification.

Text feature selection is the essential step to decrease computational complexity by eliminating noises for building a satisfactory classifier [2]. Over the years, a variety of text feature selection methods have been proposed [21]. The effective way of feature selection for relevance prediction is based on a feature weighting function which indicates the critical degree of information represented by the feature occurrences in a document and reflects the relevance of the feature to the related topic or category.

For many years, we have observed that after a set of features are selected and weighted, documents can be easily grouped into three regions rather than two categories by using a binary classifier. Even training documents previously labelled as relevant or irrelevant can not be reclassified into their original categories when applied any binary classifier [38]. Therefore, it is hard to find a clear boundary by any classic text classifier, which can be accurately described by means of mathematics, between relevant and irrelevant groups of documents as shown in Figure2, in which the "+" denotes the relevant documents and the "-" denotes the irrelevant ones, because it is almost impossible to define a curve for relevancy separation with any exact math equation as there are always many strange cases of unexpected or irregular data points. Even existing similar boundary, it is still not easy to be applied to the prediction of the incoming testing documents because of the different

conditions of different types of testing document sets [38]. In order to deal with the probable uncertainty which is difficult to be solved through traditional text classification way, the proposed relevancy prediction model tries to indirectly achieve the final purpose by partitioning the result list into different three regions for further processing and refinement by stepwise. In this paper, we propose the model for dealing with the uncertain boundary to improve the performance of binary text classification. It aims to form and train the decision boundary through partitioning the training samples into three regions (the positive region (POS), boundary region (BND), and negative region (NEG)) in order to assure the discrimination of extracted knowledge for describing relevant and irrelevant information (see Section 3). The proposed approach iteratively enhances the certainty of the two regions representing relevant and irrelevant objects, and absorbing and resolving the uncertain objects in the third region BND so as to make the knowledge on document relevancy and irrelevancy more precise and unambiguous. It starts from calculation of two main centroid vectors C_P and C_N by clustering the relevant and irrelevant training subsets, and further regroups the training samples into three regions using the two centroid vectors at basal level, with all the indeterminate objects collected into a boundary region BND, the objects with most relevant possibility to the topic stored into the POS region, and those with most irrelevant possibility to the topic collected into the NEG region.

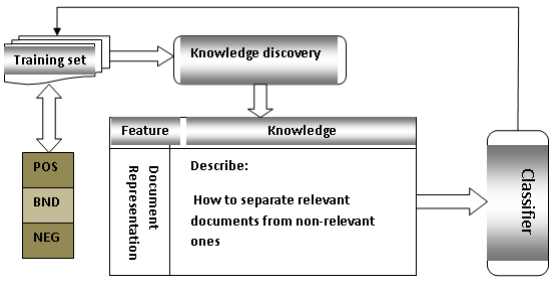


Figure 1: Schematic diagram of classifier training process

However, there must be some documents which are close to both of the two centroid vectors so as not easy to be separated clearly, and the pair of centroid vectors are not accurately located and usually closely spaced at beginning of training process. It is the key issue to approximately optimize the two pairs of centroid vectors gradually and use them for more precise text classification further, thus how to realize and improve the proposed strategy to reach the purpose is the main problem. Under such situation, one clear decision boundary is not easy to be gained for separating the documents as expected. Therefore, it is more practical to find uncertain boundary enclosed by two lines that can at least separate most of the relevant documents from the irrelevant ones during the training process, during which the uncertain or indeterminate documents are gradually absorbed into BND regions and the other two regions including POS and NEG will be enhanced the certainty, as shown in Figure2. Through the above training process, it filters as many uncertain objects gradually and save them into BND region, and makes the other two regions POS and NEG of greater certainty. During the training process the two main centroid vectors C_P and C_N and two other auxiliary centroid vectors B_P and B_N formed from the BND region are expected to be trained and optimized successively in the multi-learning process to reach the optimal condition. Figure 1 demonstrates the overall process of the classifier training. Simultaneously, the knowledge is also proposed to be updated and ultimately used for predicting the relevancy of each incoming document to the same topic so that the polarity prediction accuracy of the incoming documents will be improved. Development of vector space theory make it possible to represent and operate the documents in the type of vectors[5]. Although Rocchio classification[27] also involves the operation of centroids, the centroids have not been optimized through further learning process. We also analyze six situations based on which six decision rules are generated correspondingly to help make polarity predictions for incoming documents (see Section 4).

We have completed two series of tests based on different features including TF*IDF [28] and BM25 [31] respectively. A large number of experiments have been conducted based on the proposed approach for text classification using two standard datasets: R-CV1 [22] and Reuters21578, including the comparison analysis among the proposed model and seven other state-of-the-art baseline models (see Section 5). The experimental results show that the proposed model can significantly improve the performance of text classification in the measures of F_1 and *Accuracy*.

The evaluation of the text classification is another key issue that the paper addresses. We have chosen F_1 and *Accuracy* as the key evaluation measures. *Accuracy* reflects the accuracy degree of relevancy prediction for both relevant and irrelevant documents, and can be very high even when the number of relevant documents is zero as the number of relevant documents is usually quite low because the datasets with imbalanced class structure are used in most cases, but F_1 is an integrated, comprehensive assessment measure so as to be able to better reflect the real improvement situation of the classifier than *Accuracy*. Therefore, compared with *Accuracy*, the F_1 measure is emphasized and used for both the performance assessment of the proposed model and comparison analysis with the baseline models. Therefore, the proposed model aims to pursue substantial improvement on F_1 with the *Accuracy* guaranteed not to be reduced. Suppose we do the testing of binary text classification based on the usual datasets with imbalanced number of relevant and irrelevant targeted documents, the *Accuracy* measure may produce misleading results and is not able to reflect the real improvement degree, because even all the relevant documents are wrongly predicted as irrelevant, the *Accuracy* value will not be subjected to a big negative effect and can still be very high because of the quite low proportion of the true relevant documents. However, the calculation of F_1 depends on two factors, the *Precision* and *Recall* which can together reflect the real situation of relevant and irrelevant ratio and their improvement degrees in the testing process.

In this paper, section 2 introduces the related technologies and algorithms in text classification area; there is a detailed description of the general idea of the proposed successive approximation approach and its implementation process along with different algorithms for each step including centroid generation and training in section 3, centroid optimization in section 4, feature updating and performance improvement by Cosines laws and statistical method derived from Standard Deviation theory in section 5. The related evaluation metrics, datasets, baseline models and the experiment results are introduced to help show the

effectiveness of the model development and improvement process in section 5. Section 6 concludes the whole paper.

2 Related Work

Relevance prediction is a big research issue [17, 22] for text analysis and classification, which mainly discusses how to predict a document's relevance to a user or a category. However, the knowledge uncertainty caused by the usual knowledge mining techniques, document representation through traditional feature selection ways and traditional classification algorithms are not effective for solving relevance prediction issue because relevance is a single class problem [12].

Text classification is the process of classifying textual data into predefined categories by using classifiers learned from training samples. Text classification involves many key technologies which have certain relations with the topic and most possibly contribute to the core issues discussed in this paper. As one crucial technique of text classification, feature selection and its related methods are reviewed firstly. Then comes the analysis of some popular text classification technologies, especially some major algorithms related to this project. To date, many text classifiers such as AdaBoostM1, J48, Instance-Based Learning, kNN, Naive Bayes, SVM and Rocchio have been developed.

Document representation is one of the most important steps for text classification, in which related documents are represented by single or multiple informative features to ease the automatic operation of the documents in the subsequent steps. Feature selection can increase the performance of text classification and decrease computational complexity by eliminating noise features [2]. Feature selection is one of the important steps for text classification [30] which is the task of assigning documents to predefined classes. Feature selection plays a significant role in document representation for the purpose of text classification because a document vector is composed of a set of weighted features, and the feature number and feature quality affect the performance of text classification. The features can be simple structures (words), complex linguistic structures, statistical structures, supported information, named entities, etc. in the document. Feature selection aims to help build up the documents' vectors by selecting a subset of the key features for describing all the related documents, and remove irrespective or noise features according to corpus statistics to increase the scalability, efficiency and accuracy of a text classifier. The process of feature selection is based on a feature weighting function. A feature weighting function indicates the correlation degree of the features represented by the feature occurrences in a document and reflects the importance of the features to the document. A number of popular term weighting functions have been developed and used such as $tf*idf$ (term frequency inverse document frequency) [28], Latent Semantic Analysis (LSA) [9], Probabilistic LSA (pLSA) [13], Latent Dirichlet Allocation (LDA) [3], Chi-Square [35, 34], Information Gain [19, 35, 34], Mutual Information [11, 20], semantic structure [29], NGL coefficient [23], belief revision method [18], relevance frequency (RF) [16], pattern deploying method [32], Rocchio algorithm, Okapi BM25[26], and distributional feature [33].

Vector space model is an algebraic model for representing text documents as vectors of identifiers. Under such model, the documents are required to be represented in vectors, for example:

$$d_n = (t_{1n}, t_{2n}, \ldots, t_{mn})$$

Where d_n refers to the name of any text document, and t_{in} refers to the feature weight of any selected feature for document representation. **TF*IDF** is the basic and most effective way to calculate the feature weights. TF means the term frequency in the document, and IDF means the inverse document frequency. This method is commonly applied to weight each term in the document, which means it captures the relevancy among terms, documents and certain categories [1]. The classic formula of **TF*IDF** used for term weighting is described by the following equation:

$$w_{ij} = tf_{ij} \times log(\frac{N}{df_i}) \quad (1)$$

where w_{ij} denotes the weight of term i in document j, N denotes the total number of documents in the document set, tf_{ij} means the occurrence frequency of term i in document j, and df_i means the document frequency of term i in the document set, which represents the number of documents where a term occurs in the whole document set. It has been proven that the TF*IDF scheme is extraordinarily robust and difficult to be beaten, even by much more models and theories worked out carefully [25].

BM25 [31] is a well-known probabilistic scoring function for feature selection. From the experiments completed on the proposed model in the paper, it is found that the BM25 performs better than TF*IDF. We use the scoring function to estimate the weight of term t extracted from relevant documents as follows:

$$W(t) = \frac{tf \cdot (k_1+1)}{k_1 \cdot ((1-b) + b\frac{DL}{AVDL}) + tf} \cdot \log \frac{\frac{(r+0.5)}{(n-r+0.5)}}{\frac{(R-r+0.5)}{(N-n-R+r+0.5)}} \quad (2)$$

where N is the total number of training documents; R is the number of relevant documents; n is the number of documents which contain term t; r is the number of relevant documents which contain term t; tf is the term frequency; DL and $AVDL$ are the document length and average document length, respectively; and k_1 and b are the experimental parameters. We also use the BM25 with the parameters tuned in [39] (i.e.,$k_1 = 1.2$ and $b = 0.75$).

Classification algorithm is another key component of a text classifier. The document classification can usually be categorized in three ways including unsupervised, supervised and semi supervised methods. In the past few years, lots of classification algorithms have been developed for classifying electronic documents. We main focus on the supervised classification methods such as Naive Bayes, Support Vector Machines (SVM), Rocchio and k-Nearest Neighbour (kNN). Support Vector Machine (SVM) can be applied to classify both linear and nonlinear data. The algorithm of SVM transforms the training samples to a higher dimensional feature space through a nonlinear mapping process. SVMs is relatively successful, but the complexity of the training and categorizing algorithm cause high time and memory consumption during the training and classifying stages. [15]. Also, SVMs is highly dependent on the size of the training samples, they are not the best practice in large-scale data mining such as pattern recognition and machine learning.[36]

Bayesian classifiers can be regarded as probabilistic models. Bayesian approaches to supervised learning generally utilize Bayes law to calculate the reverse probability of the model parameters given function

input-output examples, which known as training samples [6]. Naive Bayes is similar to independent events in mathematics. It assume that all the features in a certain class are irrelevant to each other and one feature does not affect other features. These assumption bring the computation of Bayesian classifiers more efficiency but with the cost of limited applicability.

Rocchio algorithm of classification is a vector space model for text classification presented by Rocchio in 1971 [37]. This method merges relevance feedback information into the vector space model in information retrieval by building prototype vector for each class with training samples. This method is easy to implement as well as efficient in computation, but it has a potential disadvantage that the performance will be reduced when the documents belonging to a category naturally form separate clusters.

The k-Nearest Neighbour algorithm is a statistical approach to realize text classification. It ranks a document's nearest neighbours by calculating the degree of similarity between the documents and uses the top k ranked neighbours to predict the polarity of a new document. Generally, this method is efficient, but in [10] it points out that the *Accuracy* of kNN classifier depends on the value of k in turn affected by the training samples.

3 Theoretical Basis and Model Construction

Let CF be a binary text classifier, D be a training set in which all documents are labelled as either relevant D^+ or irrelevant D^-, and $F = \{f_1, f_2, \ldots, f_n\}$ is a set of terms (e.g., keywords) extracted from D. For each document $d \in D$, it can be represented as a vector $\vec{d} = (w(f_1), w(f_2), \ldots, w(f_n))$ by using the terms of F and their weights expressed by the term weighting function w.

Based on the above definitions, classifier $CF : D \longrightarrow \{R, iR\}$ will partition D into two groups: the possible relevant group R and possible irrelevant group iR for a given decision boundary or a threshold. However, it is hard to find a clear boundary by any text classifier, between relevant and irrelevant documents. Therefore, normally we have $D^+ \neq R$ and $D^- \neq iR$.

For modelling the uncertainty between relevant and irrelevant documents, we extend the classifier $CF \Longrightarrow CF'$, where $CF' : D \longrightarrow \{POS, NEG, BND\}$ is called an extended classifier, which is able to classify $d \in D$ into three regions: positive (POS, possible relevant), negative (NEG, possible irrelevant) and boundary (BND, uncertain) regions by the following definitions:

Definition 1. *If ($CF(d)$ = "R" and $d \in D^+$) Then $CF'(d)$ = "POS"; Else, If ($CF(d)$ = "iR" and $d \in D^-$) Then $CF'(d)$ = "NEG"; otherwise, $CF'(d)$ = "BND".*

Based on the above definitions, some properties about the three regions can be derived as follows:

Property 1. *If $d \in POS$ then $d \in D^+$.*

Property 2. *If $d \in NEG$ then $d \in D^-$.*

Property 3. *If $d \in D^+$ and $d \in BND$ then $CF(d)$ = "iR".*

Property 4. *If $d \in D^-$ and $d \in BND$ then $CF(d)$ = "R".*

The boundary region BND includes the uncertain decisions for relevant documents and irrelevant documents, which can be further divided into two groups: $B^+ = BND \cap D^+$ and $B^- = BND \cap D^-$.

If every document $d \in D$ is represented as a vector of term-weights, the four groups (POS, NEG, B^+ and B^-) can generate 4 centroid vectors. Let C'_P be the centroid vector of POS and C'_N be the centroid vector of NEG, B_P be the centroid vector of B^+ and B_N be the centroid vector of B^-. We also assume there is a central line (a decision boundary) between R and iR. Theorem 1 indicates the relations between them.

Theorem 1. *Let $B^+ = BND \cap D^+$ and $B^- = BND \cap D^-$, all the documents in B^+ must be below the central line, whereas all the documents in B^- must be above the central line.*

Proof. **If** there is a document $d \in B^+$, **then** according to the definition of B^+, it should be $d \in D^+$, **suppose** it is above the central line, i.e., $CF(d)$ = "R"; then it must be $d \in POS$ by Definition 1, that is against the property of B^+: $d \in BND$, **therefore** d is below the central line. **In the same way,** we can prove that any document $d \in B^-$ must be above the central line. □

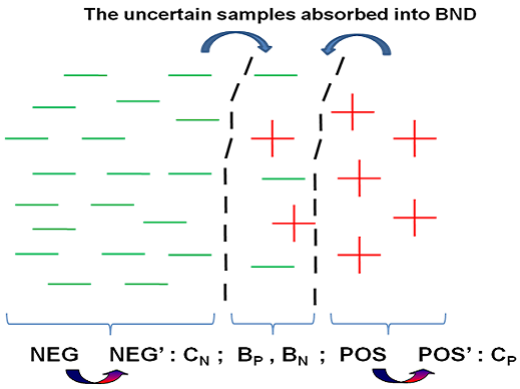


Figure 2: Training process for modelling uncertain boundary

4 Decision Rules Generation

The extended classifier CF' firstly generates two basic centroid vectors (C'_P and C'_N) to represent relevant and irrelevant information; however, there is a uncertain boundary (B^+ and B^-) between C'_P and C'_N. To assure the discrimination of C'_P and C'_N for describing relevant and irrelevant information, we propose a optimization process to iteratively update two basic centroid vectors. The process make the boundary region gradually absorb as many uncertain training documents as possible so that the two basic centroid vectors are moving away from each other accordingly until the distance between them no longer changes.

When the extended classifier CF' is produced, it is then used back to classify the training set again to update POS, NEG and BND. More uncertain documents will possibly be found and put into BND. Figure 4 shows the result example in which we can clearly see that C'_P and C'_N have been changed to C_P and C_N. The larger the gap between C_P and C_N is, the easier it would be made to separate documents apart into binary categories by comparing their distances to

the centroid vectors. In the process of this case, the size of the boundary region keeps growing and the distance between the centroid vectors C'_P and C'_N keeps increasing synchronously to reach the maximum when the training process ends.

A schematic diagram on the training process is given in Figure 3 which roughly demonstrates all the steps of the whole training and optimization process.

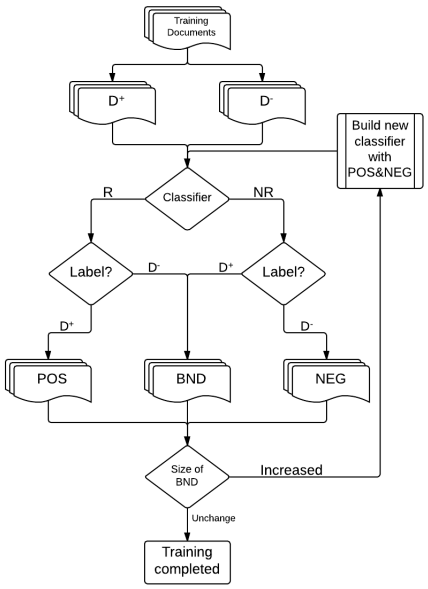


Figure 3: Centroid training and optimization process

Figure 4 shows the relations between centroid vectors, where C_P and C_N are the optimization of vectors C'_P and C'_N. We will discuss them in next section.

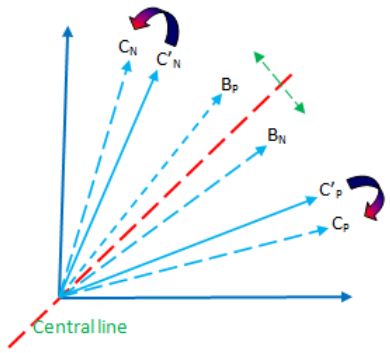


Figure 4: Four kinds of centroid vectors

For a given incoming document u, it will be compared with the two centroid vectors C_P and C_N in order to decide its relevance by using the central line and the Euclidean distance. However, the performance is poor because of the uncertain boundary. In this section, we present six decision rules to improve the performance.

Let F be the selected feature set, and $\vec{u} = (w_1, w_2, ..., w_{|F|})$ be the vector of document u and $\vec{v} = (w'_1, w'_2, ..., w'_{|F|})$ be a centroid vector. We use the following definitions to measure the distance between documents and centroid vector.

$$dis(\vec{u}, \vec{v}) = \sqrt{\sum_{j=1}^{|F|}(w_j - w'_j)^2} \quad (3)$$

$$meanDis(v) = \frac{k}{|D_v|}\sum_{d \in D_v} dis(\vec{d}, \vec{v}) \quad (4)$$

where $D_v = D^+$ if $\vec{v} = C_P$, else $D_v = D^-$ if $\vec{v} = C_N$, and k is an experiment parameter.

To predict the polarity of each incoming document $\vec{u}$, we need to understand the possible relationship between u and centroid vectors. We describe the relationship in trigonometry and use the law of cosines to display the relations. Figure 5 shows the relationship, where the round dot denotes u, "+" denotes C_P and "-" denotes C_N.

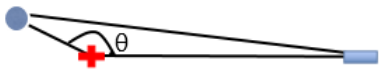


Figure 5: Example of cosines law

Below is the formula of the law of cosines:

$$\cos\theta = \frac{dis(u, C_P)^2 + dis(C_P, C_N)^2 - dis(u, C_N)^2}{2 \times dis(u, C_P) \times dis(C_P, C_N)} \quad (5)$$

Based on the law of cosines and the positions of C_P, B_N, B_P and C_N, we have six scenarios (rules) that cover all typical spatial location of the incoming document vectors for relevant analysis and decision-making of polarity prediction, as illustrated in Figure 6, where the dotted line refers to the central line; and u_1, u_2, u_3, u_4, u_5 and u_6 denote the six types of incoming document vectors in different six situations corresponding with different orientation and distance to centroid vectors, three of which are located at the left side of the central line and closer to C_P, and others are closer to C_N.

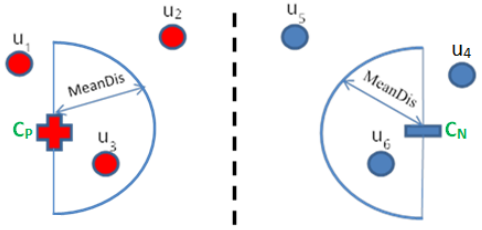


Figure 6: Six scenarios for polarity prediction

The following are the six decision rules (scenarios) for predicating the polarity (relevant or irrelevant) of each incoming document.

Rule 1 - for document u_1:

u_1 is on the left side of positive centroid, it means that u_1 is close to positive centroid and far away from the negative centroid ($dis(u_1, C_P) << dis(u_1, C_N)$). If $\cos\theta \leq 0$ ($\theta \geq \frac{\pi}{2}$, an obtuse triangle) document u_1 is predicted as relevant.

Rule 2 - for document u_2:
u_2 locates between the centroid vectors C_P and C_N but around the centroid vector B_N, specifically between B_N and the central line. Under such circumstance, we can also know that $dis(u_2, C_P) < dis(u_2, C_N)$ and θ is smaller than $\frac{\pi}{2}$, but the $dis(u_2, C_P)$ is greater than $meanDis$. Then u_2 is predicted as irrelevant.

Rule 3 - for document u_3:
u_3 is similar to u_2, but it actually locates between C_P and B_N, and the distance $dis(u_3, C_P)$ is not greater than Dis. In this case, it has a greater chance that u_3 is relevant. Therefore, u_3 is predicted as relevant.

Rule 4 - for document u_4:
u_4 is quite similar with u_1, however, it is on the right side of the negative centroid, showing that u_4 is close to C_N and far away from C_P ($dis(u_4, C_N) << dis(u_4, C_P)$). Therefore, it is predicted as irrelevant.

Rule 5 - for document u_5:
u_5 is quite similar with u_2, so the similar decision making can also be applied for it. So, it is predicted as relevant.

Rule 6 - for document u_6:
u_6 is quite similar with u_3, but the document u_6 locates between C_N instead of C_P, and B_P instead of B_N, and the distance $dis(u_6, C_P)$ is not greater than meanDis. Therefore, u_6 is predicted as irrelevant.

5 Experiments and Evaluations

5.1 Data Set

We used two popular datasets to test the proposed model: RCV1 (Reuters Corpus Volume 1), a very large data collection; and Reuters-21578, a relatively small one. RCV1 consists of all and only English language stories produced by Reuter's journalists between August 20, 1996, and August 19, 1997. RCV1 includes 806,791 documents that cover a broad spectrum of issues or topics. TREC (2002) has developed and provided 50 assessor topics for RCV1. These topics were evaluated by human assessors at the National Institute of Standards and Technology (NIST). The relevance judgements of these topics on RCV1 have also been made by the NIST assessors. For each topic, a subset of RCV1 documents is divided into a training set and a testing set. RCV1 and TREC assessor topics are standard data collections [22].

Reuters-21578 (R21578) corpus is a widely used test collection for text mining and information retrieval researches. The data was originally collected and labelled by Carnegie Group, Inc. and Reuters, Ltd. in the course of developing the CONSTRUE text categorization system[1]. In this experiment, we picked up the set of 10 classes for testing since the class distribution for documents is too skewed. According to Sebastiani's convention [8], it was called the set R8 because two classes *corn* and *wheat* are intimately related to the class *grain*, and they were appended to class *grain*. In our experiments, each class is paired with other seven classes to get more testing cases (in total, we have 56 cases). For each case, documents in the class are relevant and in another class are irrelevant.

[1] Reuters-21578, http://www.daviddlewis.com/resources/

Table 1: The algorithms of the baseline models

No	Algorithm type	Classifier
1	Function based	SVM
2	Classifiers committee based	AdaBoost
3	Decision tree based	J48 ; Random Forest
4	Probabilistic based	Naive Bayes
5	Instance-based (lazy learner)	IBk (KNN)
6	Representative based	Rocchio

To avoid bias in experiments, all of the meta-data information has been ignored. Documents are treated as plain text documents. The preprocessing tasks include removing stop words from each document according to a given list of the predefined stop words, and stemming all the terms by applying the Porter Stemming algorithm.

5.2 Baseline Models

In order to make a comprehensive evaluation, we have chosen seven types of classifiers with different algorithms from total 22 models and determined them as the baseline models (see Table 1). The selected baseline models (also see Section 2) are the state of art influential ones including Support vector machine (SVM), AdaBoostM1, J48 [24], Naive Bayes [14], Random forest [4], IBk (Instance-Based Learning), Rocchio.

Precision (p), Recall(r) are two basic parameters for evaluation of the proposed model. In the paper, the effectiveness of text classification is measured by two key measures: F_1 measure and *Accuracy* (*Acc*). F_1 is stressed as it is one of the most important metrics of comprehensive assessment [30].

$$F_1 = \frac{2PR}{P+R}\ , \quad F_1^M = \frac{\sum_{i=1}^{|\mathcal{C}|} F_{1,i}}{|\mathcal{C}|}$$

where F_1^M is the macro average of F_1 for all the tested topics, and $F_{1,i}$ is the F_1 of topic i. For the calculation of *Accuracy*,

$$Acc = \frac{TP+TN}{TP+FP+TN+FN}\ , \quad Acc^M = \frac{\sum_{i=1}^{|\mathcal{C}|} Acc_i}{|\mathcal{C}|}$$

where Acc^M is the macro average of *Accuracy* for all the the tested topics, and Acc_i is the *Accuracy* of topic i.

5.3 Experiment Results

The comparison between the proposed model (UBD) and the baseline models has been completed mainly by the two measures of F_1 and *Accuracy*. UBD is compared with seven baseline models as shown in Table 2 based on RCV1 Dataset and Table 3 based on R21578 Dataset. In Table 2, we found that the proposed model has got an average increase of 5.48% for *Accuracy* and 43.36% for F_1 compared with the other seven baseline models. The *Accuracy* value got by the proposed model exceeds SVM model which has the highest *Accuracy* value in all the baseline models, and the F_1 value has also been extremely improved by the proposed model at 116.70% compared with SVM model. In Table 3, we found that the proposed model has gained an average increase of 5.82% for *Accuracy* and 21.85% for F_1 compared with the other seven baseline models.

Table 2: The results of experiments on RCV1

No	Models	F_1	Accuracy
1	SVM	19.39%	85.45%
2	AdaBoostM1	35.46%	84.54%
3	J48	34.25%	82.85%
4	NaiveBayes	26.87%	81.62%
5	RandomForest	27.60%	84.79%
6	IBk	37.22%	82.26%
7	Rocchio	33.86%	70.13%
8	CVTO-SD-BM25-TF	**42.02%**	**85.79%**
9	Average %chg	**43.36%**	**5.48%**

Table 3: The results of experiments on R21578

No	Models	F_1	Accuracy
1	SVM	60.96%	85.46%
2	AdaBoostM1	56.79%	81.26%
3	J48	64.12%	85.39%
4	NaiveBayes	79.54%	82.49%
5	RandomForest	66.01%	85.25%
6	IBk	75.10%	86.45%
7	Rocchio	69.39%	71.16%
8	CVTO-SD-BM25-TF	**81.20%**	**86.95%**
9	Average %chg	**21.85%**	**5.82%**

Table 2 and Table 3 indicate that the proposed model has the highest score in both F_1 and *Accuracy* on two datasets, especially in F_1 that best reflects the real situation of text classification performance. Therefore, the proposed partitioning approximation approach has gained the best performance on RCV1 and R21578 compared with all the collected seven influential baseline models.

6 Conclusion

This paper proposed the method for dealing with uncertain decision boundary for finding relevant information. The experimental results show that the proposed model can significantly improve the performance of binary text classification in both F_1 and *Accuracy* compared with seven other influential baseline models. The proposed model is promising and has the following contributions:

- Developed a model to understand the difference between relevant and irrelevant information by dividing training documents into three regions to reduce the impact of the uncertain information for text classification.
- Presented six decision rules to improve the performance of binary text classification on F_1 measure and *Accuracy*.

References

[1] B. Baharudin, L. H. Lee, and K. Khan. A review of machine learning algorithms for text-documents classification. *Journal of advances in information technology*, 1(1):4–20, 2010.

[2] R. Bekkerman and M. Gavish. High-precision phrase-based document classification on a modern scale. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 231–239. ACM, 2011.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[4] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[5] C. J. Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

[6] J. L. Carroll. *A bayesian decision theoretical approach to supervised learning, selective sampling, and empirical function optimization*. PhD thesis, Brigham Young University, 2010.

[7] T. Cover and P. Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, 1967.

[8] F. Debole and F. Sebastiani. An analysis of the relative hardness of reuters-21578 subsets. *Journal of the American Society for Information Science and technology*, 56(6):584–596, 2005.

[9] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, 41(6):391–407, 1990.

[10] H. Doshi and M. Zalte. Comparison of supervised learning techniques for binary text classification. *IJCSIS) International Journal of Computer Science and Information Security*, 10(9), 2012.

[11] S. T. Dumais, J. C. Platt, D. Hecherman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *CIKM*, pages 148–155, 1998.

[12] G. Forman. An extensive empirical study of feature selection metrics for text classification. *The Journal of machine learning research*, 3:1289–1305, 2003.

[13] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of SIGIR*, pages 50–57. ACM, 1999.

[14] G. H. John and P. Langley. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 338–345. Morgan Kaufmann Publishers Inc., 1995.

[15] L. Khan, M. Awad, and B. Thuraisingham. A new intrusion detection system using support vector machines and hierarchical clustering. *The VLDB Journal—The International Journal on Very Large Data Bases*, 16(4):507–521, 2007.

[16] M. Lan, C. L. Tan, J. Su, and Y. Lu. Supervised and traditional term weighting methods for automatic text categorization. *IEEE Transations on Pattern Anaalysis and Machine Intelligence*, 31:721–735, April 2009.

[17] R. Y. Lau, P. D. Bruza, and D. Song. Towards a belief-revision-based adaptive and context-sensitive information retrieval system. *ACM Transactions on Information Systems (TOIS)*, 26(2):8, 2008.

[18] R. Y. K. Lau, P. Bruza, and D. Song. Belief revision for adaptive information retrieval. In *SIGIR*, pages 130–137, 2004.

[19] D. D. Lewis. Feature selection and feature extraction for text categorization. In *Speech and Natural Language Workshop*, pages 212–217, San Francisco, 1992.

[20] D. D. Lewis and M. Ringuette. A comparison of two learning algorithms for text categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 81–93, Las Vegas, US, 1994.

[21] Y. Li, A. Algarni, and N. Zhong. Mining positive and negative patterns for relevance feature discovery. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 753–762. ACM, 2010.

[22] Y. Li, A. Algarni, and N. Zhong. Mining positive and negative patterns for relevance feature discovery. In *Proceedings of SIGKDD*, pages 753–762. ACM, 2010.

[23] H. T. Ng, W. B. Goh, and K. L. Low. Feature selection, perceptron learning, and a usability case study for text categorization. In *SIGIR*, pages 67–73, 1997.

[24] J. R. Quinlan. *C4.5: programs for machine learning*, volume 1. Morgan kaufmann, 1993.

[25] S. Robertson. Understanding inverse document frequency: on theoretical arguments for idf. *Journal of documentation*, 60(5):503–520, 2004.

[26] S. E. Robertson and I. Soboroff. The trec 2002 filtering track report. In *TREC*, volume 2002, page 5, 2002.

[27] J. Rocchio. Relevance feedback in information retrieval. *SMART Retrieval System Experimens in Automatic Document Processing*, pages 313–323, 1971.

[28] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24(5):513–523, 1988.

[29] H. Schütze, D. A. Hull, and J. O. Pedersen. A comparison of classifiers and document representations for the routing problem. In *SIGIR*, pages 229–237, 1995.

[30] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[31] J. S. Whissell and C. L. Clarke. Improving document clustering using okapi bm25 feature weighting. *Information retrieval*, 14(5):466–487, 2011.

[32] S. Wu, Y. Li, and Y. Xu. Deploying approaches for pattern refinement in text mining. In *Proceedings of ICDM'06. Sixth International Conference on Data Mining.*, pages 1157–1161, 2006.

[33] X.-B. Xue and Z.-H. Zhou. Distributional features for text categorization. *IEEE Transactions on K*, 21(3):428 – 442, 2009.

[34] Y. Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1:69–90, 1999.

[35] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *ICML*, pages 412–420, 1997.

[36] H. Yu, J. Yang, J. Han, and X. Li. Making svms scalable to large data sets using hierarchical cluster indexing. *Data Mining and Knowledge Discovery*, 11(3):295–321, 2005.

[37] A. Zeng and Y. Huang. A text classification algorithm based on rocchio and hierarchical clustering. In *Advanced Intelligent Computing*, pages 432–439. Springer, 2012.

[38] L. Zhang, Y. Li, C. Sun, and W. Nadee. Rough set based approach to text classification. In *2013 WI/IAT and IEEE/WIC/ACM International Joint Conferences*, volume 3, pages 245–252. IEEE, 2013.

[39] N. Zhong, Y. Li, and S. Wu. Effective pattern discovery for text mining. *Knowledge and Data Engineering, IEEE Transactions on*, 24(1):30–44, 2012.

Pattern-based Topic Modelling for Query Expansion

Yang Gao **Yue Xu** **Yuefeng Li**

School of Electrical Electronics and Computer Science
Queensland University of Technology
Brisbane, Queensland, 4000
Email: [y10.gao@connect,yue.xu@,y2.li@]qut.edu.au

Abstract

One big problem with information retrieval (IR) is that the size of queries is usually short and the keywords in a query are very often ambiguous or inconsistent. Automatic query expansion is a widely recognized technique which is effective to deal with this problem. However, many query expansions methods require extra information such as explicit relevance feedback from users or pseudo relevance feedback from retrieval results. In this paper, we propose an unsupervised query expansion method, called Topical Query Expansion (TQE), which does not require extra information. The proposed TQE method expands a given query based on the topical patterns which can create links among those more associated and semantic words in each topic. This model also discovers related topics that are related to the original query. Based on the expanded terms and related topics, we propose to rank the document relevance with different ranking strategies. We conduct experiments on popularly used datasets, TREC datasets, to evaluate the proposed methods. The results demonstrate outstanding results against several state-of-the-art models.

Keywords: Topical Pattern, Information Retrieval, Query Expansion

1 Introduction

Standard bag-of-words retrieval models, such as BM25 (Lv & Zhai 2011, Robertson et al. 2004), have the benefits of mature statistical theories and their efficient computational performance which produce reasonable good results. However, these models are restricted on limited number of features that are from query terms. Besides, the words in a query may not be consistent and can be ambiguously understood or interpreted.

Automatic Query Expansion (AQE) is considered as an promising technique in IR to improve the effectiveness of document retrieval especially for short queries. Most researches of AQE techniques involve relevance feedback (Andrzejewski & Buttler 2011, Maxwell & Croft 2013) and are always combined with smoothing technique to improve the effectiveness(Yi & Allan 2009, Zhai & Lafferty 2001), etc. But one limitation of these methods in real Web search is their high cost in computation because fast response time required by Web search applications.

Targeting on the problems mentioned above, in this paper, we will propose a new IR model which includes a novel query expansion method and a novel document ranking method, both based on topical patterns generated from the document collection. The proposed query expansion method can precisely interpret the user interests even though only limited features are provided and also can precisely balance the query drift during the process of the query expansion. The proposed IR model can efficiently retrieve relevant documents because no online re-training is needed. In the real scenario, if users are not very familiar with the content related to their formulated queries, they will not be able to provide interactive refinement within a limited response time, therefore the relevance feedback models are not suitable under this situation. In contrast, in this case, the proposed IR model will be much more useful.

The new proposed Topical Query Expansion (TQE) model, can effectively expand queries with semantic topical patterns. In the model, the discovered relevant topical patterns are used to determine the certain topics and uncertain topics for a specific query. For document relevance ranking, this distinguish for the related topics can be reflected by different weighting mechanisms. The rest of this paper will be presented as follows. In Section 2, we review topic models and related state-of-the-art IR tehniques. Section 3 introduces the method of generating pattern-based topic model. Then we propose a new method for query expansion based on the pattern-based topic representation and the topic-based weighting system, which are described in Section 4. In Section 5 we describe the ad hoc retrieval experiments and prove the proposed model is significantly improve the effectiveness. According to the experimental results, we discuss the strengths of the proposed model from different perspectives in Section 6. At last, Section 7 concludes the whole work and presents the future direction.

2 Related Work

Topic modelling algorithms are used to discover a set of hidden topics from collections of documents, where a topic is represented as a distribution over words. Topic models provide an interpretable low-dimensional representation of documents (i.e. with a limited and manageable number of topics). Latent Dirichlet Allocation (LDA)(Blei et al. 2003) is a typical statistical topic modelling technique and the most common topic modelling tool currently in use. It can discover the hidden topics in collections of documents using the words that appear in the documents. Let

Table 1: Example results of LDA: word-topic assignments

Topic		Z_1		Z_2		Z_3
d	$\vartheta_{d,1}$	Words	$\vartheta_{d,2}$	Words	$\vartheta_{d,3}$	Words
d_1	0.6	w_1, w_2, w_3, w_2, w_1	0.2	w_1, w_9, w_8	0.2	w_7, w_{10}, w_{10}
d_2	0.2	w_2, w_4, w_4	0.5	w_7, w_8, w_1, w_8, w_8	0.3	w_1, w_{11}, w_{12}
d_3	0.3	w_2, w_1, w_7, w_5	0.3	w_7, w_3, w_3, w_2	0.4	w_4, w_7, w_{10}, w_{11}
d_4	0.3	w_2, w_7, w_6	0.4	w_9, w_8, w_1	0.3	w_1, w_{11}, w_{10}

$D = \{d_1, d_2, \cdots, d_M\}$ be a collection of documents. The total number of documents in the collection is M. The idea behind LDA is that every document is considered to contain multiple topics and each topic can be defined as a distribution over a fixed vocabulary of words that appear in the documents. For the ith word in document d, denoted as $w_{d,i}$, the probability of $w_{d,i}$, $P(w_{d,i})$ is defined as:

$$P(w_{d,i}) = \sum_{j=1}^{V} P\left(w_{d,i}|z_{d,i} = Z_j\right) \times P\left(z_{d,i} = Z_j\right) \quad (1)$$

$z_{d,i}$ is the topic assignment for $w_{d,i}$, $z_{d,i} = Z_j$ means that the word $w_{d,i}$ is assigned to topic j and the V represents the total number of topics. Let ϕ_j be the multinomial distribution over the words for Z_j, $\phi_j = (\varphi_{j,1}, \varphi_{j,2}, \cdots, \varphi_{j,n})$, $\sum_{k=1}^{n} \varphi_{j,k} = 1$. θ_d refers to multinomial distribution of the topics in document d. $\theta_d = (\vartheta_{d,1}, \vartheta_{d,2}, \cdots, \vartheta_{d,V})$, $\sum_{j=1}^{V} \vartheta_{d,j} = 1$. $\vartheta_{d,j}$ indicates the proportion of topic j in document d. LDA is a generative model in which the only observed variable is $w_{d,i}$, while the others are all latent variables that need to be estimated. Gibbs sampling method is an effective strategy for hidden parameters estimation(Steyvers & Griffiths 2007) that is used in this paper.

The resulting representations of the LDA model are at two levels, document level and collection level. Apart from these, the LDA model also generates word-topic assignments, that is, the word occurrence is considered related to the topics by LDA. Take a simple example and let $D = \{d_1, d_2, d_3, d_4\}$ be a small collection of four documents with 12 words appearing in the documents. Assuming the documents in D involve 3 topics, Z_1, Z_2 and Z_3. Table 1 illustrates the topic distribution over documents and word-topic assignments in this small collection. From the outcomes of the LDA model, the topic distribution over the whole collection D can be calculated, $\theta_D = (\vartheta_{D,1}, \vartheta_{D,2}, \cdots, \vartheta_{D,V})$, where $\vartheta_{D,j}$ indicates the importance degree of the topic Z_j in the collection D.

LDA has been widely accepted by IR community. It was combined with language model for document smoothing (Mei et al. 2008, Yi & Allan 2009) (typically like LBDM (Wei & Croft 2006)) and used for query expansion (i.e., model-based feedback (Zhai & Lafferty 2001) and relevance model with Markov random fields (Lavrenko & Croft 2001, Metzler & Croft 2007)) techniques. The combination works mainly because that it takes the advantages from LDA's multiple topic representation for document modelling and relevance feedback viewing each document as topics to discover better query-specific topics. However, most of these works assume that query terms are independent given documents, which makes query expansion can not always keep good performance.

The topical n-Gram model (TNG) proposed in (Wang et al. 2007) automatically discovers term relationships within topics and extracts topically relevant and flexible phrases. Also topical PageRank can extract keyphrases in (Liu et al. 2010). But syntactically valid phrases often share low frequency in documents which cause poor performance for some queries. In (Bai et al. 2005), dependence models have been incorporated to extract term relationships for query expansion. This combination of terms are more flexible than phrases and the expanded terms are dependent to query and document. However, the risk of query drift is still a problem in query expanding area.

To solve the problem, the concept of optimization is prevalent for choosing relevant information. For example, optimization is treat as a classification task (Cao et al. 2008) that discriminates relevant from irrelevant expansion terms depending on whether they improve the performance. The approach proposed in (Maxwell & Croft 2013) focus on in-query terms selection by defining informative words and incorporating global statistics and local syntactic phrase to improve the performance. Optimised smoothing (Mei et al. 2008) technique is a general unified optimization framework for smoothing language models on graph structures. Collins-Thompson (Collins-Thompson 2009) defines a uncertainty set and models constraints to minimise the optimal loss over this set. Our approach partially inherits the idea of "mitigate risk-reward tradeoff", but we additionally provide more concrete and meaningful categories to estimate the expanded query and the methods are proposed from different perspectives.

In this paper, we propose a new approach that incorporates multiple topics from topic model and association rule mining techniques, in the sense that discovers semantic meaning of topics and inferentially exploits related terms for original query. And we also present a new ranking method to systematically weight the different importances for all expanded and original queries.

3 Pattern-based Topic Representation

Pattern-based representations are considered more meaningful and more accurate to represent topics than word-based representations. Moreover, pattern-based representations contain structural information which can reveal the association between words. In order to discover semantically meaningful patterns to represent topics and documents, two steps are proposed: firstly, construct a new transactional dataset from the LDA model results of the document collection D; secondly, generate pattern-based representations from the transactional dataset to represent user needs of the collection D.

3.1 Construct Transactional Dataset

Let R_{d_i,Z_j} represent the word-topic assignment to topic Z_j in document d_i. R_{d_i,Z_j} is a sequence of

words assigned to topic Z_j. For the example illustrated in Table 1, for topic Z_1 in document d_1, $R_{d_1,Z_1} = \langle w_1, w_2, w_3, w_2, w_1 \rangle$. We construct a set of words from each word-topic assignment R_{d_i,Z_j} instead of using the sequence of words in R_{d_i,Z_j}, because for pattern mining, the frequency of a word within a transaction is insignificant. Let I_{ij} be a set of words which occur in R_{d_i,Z_j}, $I_{ij} = \{w | w \in R_{d_i,Z_j}\}$, i.e. I_{ij} contains the words which are in document d_i and assigned to topic Z_j by LDA. I_{ij}, called a *topical document transaction*, is a set of words without any duplicates. From all the word-topic assignments R_{d_i,Z_j} to Z_j, we can construct a transactional dataset Γ_j. Let $D = \{d_1, \cdots, d_M\}$ be the original document collection, the transactional dataset Γ_j for topic Z_j is defined as $\Gamma_j = \{I_{1j}, I_{2j}, \cdots, I_{Mj}\}$. For the topics in D, we can construct V transactional datasets $(\Gamma_1, \Gamma_2, \cdots, \Gamma_V)$. An example of transactional datasets is illustrated in Table 2, which is generated from the example in Table 1.

3.2 Generate Pattern Enhanced Representation

The basic idea of the proposed pattern-based method is to use frequent patterns generated from each transactional dataset Γ_j to represent Z_j. In the two-stage topic model (Gao, Xu, Li & Liu 2013), frequent patterns are generated in this step. For a given minimal support threshold σ, an itemset X in Γ_j is frequent if $supp(X) >= \sigma$, where $supp(X)$ is the support of X which is the number of transactions in Γ_j that contain X. The frequency (also called relative support) of the itemset X is defined $\frac{supp(X)}{|\Gamma_j|}$. Topic Z_i can be represented by a set of all frequent patterns, denoted as $\mathbf{X}_{Z_i} = \{X_{i1}, X_{i2}, \cdots, X_{im_i}\}$, where m_i is the total number of patterns in $\mathbf{X}_{Z_i}$. For all topics, we have $\mathbf{X}_{Z_1}, \mathbf{X}_{Z_2}, \cdots, \mathbf{X}_{Z_V}$ and V is the total number of topics. $\mathbf{U} = \{\mathbf{X}_{Z_1}, \mathbf{X}_{Z_2}, ..., \mathbf{X}_{Z_V}\}$ is the pattern-based topic model generated from the given collection of documents. Take Γ_2 as an example, which is the transactional dataset for Z_2. For a minimal support threshold $\sigma = 2$, all frequent patterns generated from Γ_2 are given in Table 3 ('itemset' and 'pattern' are interchangeable in this paper).

Table 2: Transactional datasets generated from Table 1(topical document transaction(TDT))

T	TDT	TDT	TDT
1	$\{w_1, w_2, w_3\}$	$\{w_1, w_8, w_9\}$	$\{w_7, w_{10}\}$
2	$\{w_2, w_4\}$	$\{w_1, w_7, w_8\}$	$\{w_1, w_{11}, w_{12}\}$
3	$\{w_1, w_2, w_5, w_7\}$	$\{w_2, w_3, w_7\}$	$\{w_4, w_7, w_{10}, w_{11}\}$
4	$\{w_2, w_6, w_7\}$	$\{w_1, w_8, w_9\}$	$\{w_1, w_{11}, w_{10}\}$
	Γ_1	Γ_2	Γ_3

Table 3: The frequent patterns for Z_2, $\sigma = 2$

Patterns	*supp*
$\{w_1\}, \{w_8\}, \{w_1, w_8\}$	3
$\{w_9\}, \{w_7\} \{w_8, w_9\}, \{w_1, w_9\}, \{w_1, w_8, w_9\}$	2

4 Topical Query Expansion (TQE)

The representations generated by the pattern-based LDA model, discussed in Section 3, carry more concrete and identifiable meaning than the word-based representations generated using the original LDA model. Based on the pattern-based topic model with layered structure and interpretable representation, we will present the method of expanding queries by our proposed novel Topical Query Expansion (TQE) model for IR. The details are described in the following subsections.

4.1 Related Topics Selection and Query Expansion

Given a collection of documents D, V pre-specified latent topics can be generated and represented by patterns according to the approach described in Section 3. From the results of LDA to D, V transactional datasets, $\Gamma_1, \cdots, \Gamma_V$ can be generated from which the pattern-based topic representations for the collection, $\mathbf{U} = \{\mathbf{X}_{Z_1}, \mathbf{X}_{Z_2}, \cdots, \mathbf{X}_{Z_V}\}$, can be generated, where each $\mathbf{X}_{Z_i} = \{X_{i1}, X_{i2}, \cdots, X_{im_i}\}$ is a set of frequent patterns generated from the transactional dataset Γ_i, where m_i is the total number of patterns in topic Z_i and each X_{ij} in $\mathbf{X}_{Z_i}$ is a unique pattern with corresponding weight f_{ij}. This pattern-based representation enhances the semantic meaning of topics, which can also be useful for selecting right topics that a query may involve.

Normally, the number of frequent patterns is considerably large and many of them are not necessarily useful. Several concise patterns have been proposed to represent useful patterns generated from a large dataset instead of frequent patterns, such as maximal patterns (Bayardo Jr 1998) and closed patterns. The number of these concise patterns is significantly smaller than the number of frequent patterns for a dataset. Based on the results on information filtering (Gao, Xu & Li 2013, Gao et al. 2014), the topics that are discovered from the collection of documents can be better represented by closed patterns.

For a transactional dataset, an itemset X is a *closed itemset* if there exists no itemset X' such that (1) $X \subset X'$, (2) $supp(X) = supp(X')$. A closed pattern reveals the largest range of the associated terms. It covers all information that its subsets describe. Closed patterns are more effective and efficient to represent topics than frequent patterns.

Association rule mining is to find associations between itemsets, called *association rules*. An *association rule* is an implication in the form of $X \Rightarrow Y$, where X and Y are disjoint itemsets, i.e., $X \cap Y = \emptyset$. The strength of an association rule can be measured in terms of its *support* and *confidence*. *Support* determines how often a rule is applicable to a given dataset, while *confidence* determines how interesting or strong the correlations of this rule. X and Y are itemsets, X is called antecedent and Y is called consequent, the *rule* means that X implies Y. The *relative support* of a rule is the percentage of transactions that contain X and Y, the *confidence* of a rule is the ratio between the support of the rule and the support of X.

Confidence, on the other hand, measures the reliability of the inference made by a rule. For a given $X \Rightarrow Y$, the higher the confidence, the more likely it is for Y to be present in transactions that contain X. The association rule suggests co-occurrence relationship between items in the antecedent and consequent of the rule.

For the pattern-based topic model described in

Section 3, from each transactional dataset 'Γ_i' for topic i, we can generate a set of association rules which satisfy the predefined minimum support σ and confidence η from a given transactional dataset, and denoted as $\mathbb{R}_i$. Based on the discovered rules from pattern-based topic $\{\mathbf{X}_{Z_1}, \mathbf{X}_{Z_2}, \cdots, \mathbf{X}_{Z_V}\}$, for a given query $Q = \{q_1, q_2, \cdots, q_n\}$, where q_w is one of the terms in the query Q, $w = 1, 2, \cdots, n$, we can discover which topics that the query is related to, as well as the expanded terms of original queries.

The rational behind using pattern based topic models to expand queries is topical patterns contain more strong relations among terms and these associations create a reliable links between original query terms and their related topics from which the best terms can be determined to expand the query. The detailed process is described as follows.

As mentioned before, the pattern-based topic representation is $\mathbf{X}_{Z_i} = \{X_{i1}, X_{i2}, \cdots, X_{im_i}\}$ for topic Z_i in which the pattern $X_{ij} = \left\{x_{ij}^1, x_{ij}^2, \cdots, x_{ij}^{l_{ij}}\right\}$ is a set of terms, l_{ij} is the length of this pattern X_{ij}, $supp(X_{ij})$ is the support of X_{ij}. If there is a term $x_{ij}^k \in X_{ij}, k \in \{1, \cdots, l_{ij}\}$, $q_w = x_{ij}^k$, i.e., $q_w \in X_{ij}$, and $q_w \Rightarrow X_{ij} \backslash q_w$ is a rule in $\mathbb{R}_i$, topic Z_i is considered as a related topic of q_w. The pattern X_{ij} is called a topic-related patten of q_w. The set of related topics of the query word q_w, denoted as RT_{q_w} can be defined below:

$$RT_{q_w} = \{Z_i | \exists (q_w \Rightarrow X_{ij} \backslash q_w) \in \mathbb{R}_i, q_w \in X_{ij}\} \quad (2)$$

The set of related topics for a query Q is defined as:

$$RT_Q = \bigcup_{w=1}^{n} RT_{q_w} \quad (3)$$

For a query word q_w, there could be multiple topic-related patterns in X_{Z_i} that make the topic Z_i a related topic of q_w. Let $X_{q_w}^i$ be a set of topic-related patterns in X_{Z_i} for q_w, $X_{q_w}^i$ can be used to represent topic Z_i in terms of q_w. $X_{q_w}^i$ is defined below:

$$X_{q_w}^i = \{X | X \in X_{Z_i}, \exists (q_w \Rightarrow X \backslash q_w) \in \mathbb{R}_i, q_w \in X\} \quad (4)$$

For each pattern $X \in X_{q_w}^i$, the expanded query is $X \backslash q_w$ and the relevance of X to q_w with respect to the topic Z_i is defined as:

$$f_{q_w}^i(X) = \frac{supp(X)}{\sum_{Y \in X_{q_w}^i} supp(Y)} \quad (5)$$

The relevance $f_{q_w}^i$ will be used to determine the relevance of a document to a query in the retrieval stage which will be discussed in Section 4.3.

A simple example is given in Figure 1 where we describe how to find related topics for word "trade" in query 55. The minimum confidence in our experiment $\eta = 0.25$. The related patterns in topic 59 are "trade stock", "trade market" and "trade exchange" in Figure 1. The relative support of the pattern "trade stock" is 0.44, and the relative support of "trade" is 0.59 in topic 59, and $0.44/0.59 = 0.75$ is the confidence of the rule "trade⇒stock" which is larger than η. As a result, topic 59 is one of the related topics for the keyword "trade". Similarly, Z_{28} is another related topic and thus $RT_q = \{Z_{59}, Z_{28}\}$ is the set of related topics of keyword "trade".

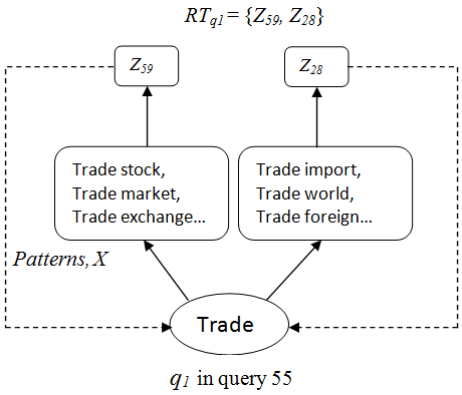


Figure 1: An example of finding related topics given a keyword "trade" from query 55

As results, the corresponding expanded queries contain "stock, market, exchange, import, world, foreign". But among all the related topics, the association degree with one word in the query is different. Some related topics are highly related to the user's interests while some other discovered related topics are actually rarely associated. So we need topic optimization process to differentiate the various importance of all the related topics to the user's real interests.

4.2 Topic Optimization

In traditional topic models, topics are represented by single words. One word could appear in different topics since words suffer from the polysemy problem. In our pattern-based topic model, the topic representation is more discriminative by patterns. Topical patterns can capture more stable and meaningful associations between words. However, different combination with the same word can often represent different topics. For example, "south" in "south Africa" is country name but it in "south west" is a direction. As a result, not all the related topics generated using Equation (3) can 'truly' represent the user's interests. Topic optimization is such a process that helps to choose those related topics which are more closed to the user's interests. In this section, we will optimise the selected related topics, RT_Q, for the query, and define a level of certainty for these related topics.

A topic Z_i is considered as the user's certain topic if it meets the following conditions:

- Z_i is a related topic of at least keyword in Q, i.e., $\exists q_k \in Q, Z_i \in RT_{q_k}$;
- Z_i is a common related topic of at least two different keywords in Q, i.e., $\exists q_h \in Q, Z_i \in RT_{q_h}$

Formally, the *certain topics* of the query Q , denoted as T_Q^c, is defined by the equation (6).

$$T_Q^c = \left\{Z_i | Z_i \in RT_{q_k} \bigcap RT_{q_h}, \exists k, h \in \{1, \cdots, n\}, k \neq h\right\} \quad (6)$$

The set of *certain topics* can be considered as the closest topics to the user's interest because they are related to at least two query words in the user' query. This feature is very important because two or more words actually form a pattern. The pattern consisting of query words can be considered a 'user-specific pattern'. It is because of the 'user-specific pattern' that the topics in T_Q^p are more certain to represent the user's interest.

The other related topics other than the certain topics in T_Q^p are considered as a set of *uncertain topics*, T_Q^u:

$$T_Q^u = RT_Q \backslash T_Q^p \tag{7}$$

A related topic in the set of *uncertain topics* contains only one original query keyword and it only satisfies the first condition.

Let $RT_{q_w}^p$ be the set of certain topics of q_w and $RT_{q_w}^u$ be the set of uncertain topics of q_w:

$$RT_{q_w}^p = \{Z | Z \in RT_{q_w}, Z \in T_Q^p\} \tag{8}$$

$$RT_{q_w}^u = \{Z | Z \in RT_{q_w}, Z \in T_Q^u\} \tag{9}$$

4.3 Document Ranking

Original query Q can be expanded by patterns from the representation of related topics RT_Q based on their property (i.e., certain topic or uncertain topic). Expanded patterns of related topic (in section 4.1) which belongs to the certain topics are called certain expansion, and patterns of related topic which belongs to uncertain topics are uncertain expansion. For different expansions, we assign different weights to indicate their importance using a trade-off parameter $\lambda \in [0, 1]$. A constant $\lambda_s \in [0, 1]$ controls the relative proportion between original query and the extended terms, and a constant $\lambda_p \in [0, 1]$ controls the weighting trade-off between extended terms in T_Q^c and the ones in T_Q^u. Also, the expanded terms have their topical frequencies that can represent the their specificities in describing the meaning of the related topic, which is calculated by Equation (5). The higher relevance of the pattern indicates the more specific and important of this pattern in this topic.

For example, query 58 is "rail striker", the related topic for "rail" is topic66 in which the expanded patterns are "transport rail" with relative support 0.05918 and "rail train" with relative support 0.05477; the related topic for "striker" is topic84 in which the expanded patters are "striker worker" with relative support 0.06155 and "striker union" with relative support 0.05352. According to Equation (5), the relevance of the expanded patterns will be "transport rail"(0.5194) and "rail train"(0.4806) in topic66, "striker worker"(0.5349) and "striker union"(0.4651) in topic84. However, queries are always related to multiple topics. The more related topics a query has, the more diverse the query is, thus more uncertain for this query. The number of related topics in RT_{q_w}, is defined as word *diversity* of q_w, denoted as $div(q_w)$. For the set of certain topics, $div_{q_w}^c = |RT_{q_w}^c|$ and for the set of uncertain topic, $div_{q_w}^u = |RT_{q_w}^u|$. If a word has high diversity, it will not be discriminative in delivering the user interest therefore the importance weight should be lower than the word with low diversity.

The principles of document relevance ranking are: 1) increase weights of certain related topics to user's interests; 2) balance the weights of uncertain related topics to user's interests. The expanded terms in certain topics are assigned higher weight compared with expanded terms in uncertain topics in which the proportion is controlled by λ_f. At more specific level, diversity (div)of the keyword is used to balance every possible meaning of it; the relevance of the pattern $f(X)$ indicates its importance in one topic. $\#div \times f(X)\#$ together presents the specificity of the expanded terms.

Table 4: Important Notations

Q	the original query
T_Q^c	set of certain topics for the query Q
T_Q^u	set of uncertain topics for the query Q
$RT_{q_w}^p$	set of certain topics of q_w
$RT_{q_w}^u$	set of uncertain topics of q_w
$X_{q_w}^i$	set of topic-related patterns in X_{Z_i} for q_w
$f_{q_w}^i(X)$	the relevance of pattern X to q_w with respect to the topic Z_i
λ_p, λ_s	trade-off parameters
$bm(q)$	the BM25 score of term q

One unique characteristic for topical pattern expansion is that the expanded terms particularly for an original query term are derived from one pattern, and they only make sense when the original query exists in a document since the original keyword and expanded terms together represent a completely meaningful pattern. Another reason is the original query is the antecedent of a rule which is a dominant element. Therefore, our proposed document ranking requires co-occurrence of the original query and its expanded terms in a document d. The ranking score of document d will linearly combine all these elements introduced above. The document ranking is calculated by Equation (10), the relevant notations in the equations are given in Table 4.

$$\begin{aligned} score(d|Q) = \sum_{\substack{q_w \in Q \\ q_w \in d}} \{\lambda_s bm(q_w) + (1-\lambda_s)\{ \\ \frac{\lambda_p}{|RT^c(q_w)|} \sum_{Z_i \in T_Q^c} \sum_{X \in X_{q_w}^i} f_{q_w}^i(X) bm(X \backslash q_w) + \\ \frac{1-\lambda_p}{|RT^u(q_w)|} \sum_{Z_i \in T_Q^u} \sum_{X \in X_{q_w}^i} f_{q_w}^i(X) bm(X \backslash q_w))\}\} \end{aligned} \tag{10}$$

4.4 Algorithm

To understand this process clearly, we formally describe the process in two algorithms: Offline training (i.e., generating pattern-based topic model for the collection) Algorithm and Online Retrieval (i.e., expanding queries and document relevance ranking) Algorithm. The former generates pattern-based topic representations and paves the way for the latter expansion system. Given a query online, the latter TQE model expands the query with related topics and computes the document ranking with topic-related patterns.

5 Evaluation

In order to emphasize the effectiveness of our proposed weighted query expansion model, in the experiments, we only use query itself as the input without user interactive information such as relevance feedbacks. The hypothesis to be verified in the experiments to be discussed in the following sections is that the proposed query expansion based on the related topical patterns generated from the collection is effective. This section discusses the experiments and evaluation in terms of data collection, baseline models, measures and results. The results show that the

Algorithm 1 *Offline Training*

Input: a collection of training documents D;
minimum support σ_j as threshold for topic Z_j;
number of topics V
Output: pattern-based topic representations for the collection, $\mathbf{X}_{Z_1}, \mathbf{X}_{Z_2}, \cdots, \mathbf{X}_{Z_V}$

```
Generate topic representation φ and word-topic assignment z_{d,i} by applying LDA to D
for each topic Z_i ∈ [Z_1, Z_V] do
    Construct transactional dataset Γ_i based on φ and z_{d,i}
    Construct user interest model X_{Z_i} for topic Z_i using a pattern mining technique so that for each pattern X in X_{Z_i}, supp(X) > σ_i
end for
```

Algorithm 2 *Online Retrieval*

Input: topic representations $\mathbf{X}_{Z_1}, \mathbf{X}_{Z_2}, \cdots, \mathbf{X}_{Z_V}$;
a query $Q = \{q_1, q_2, \cdots, q_n\}$;
a list of documents D';
a minimum confidence η;
Output: $score(d|Q), d \in D'$

```
for each q_w ∈ Q do
    RT_{q_w} := ∅
    for each topic Z_i ∈ [Z_1, Z_V] do
        X^i_{q_w} := ∅
        for each pattern X_{ij} ∈ X_{Z_i} do
            Scan patterns and find q_w = x_{ij}, x_{ij} ∈ X_{ij}
            if q_w ⇒ X_{ij}\q_w ∈ R_i then
                RT_{q_w} = Z_i ∪ RT_{q_w}
                X^i_{q_w} = X_{ij}\q_w ∪ X^i_{q_w}
            end if
        end for
    end for
end for
for all q_w ∈ Q do
    Find T^c_Q by equation (6)
end for
for each d ∈ D' do
    score(Q|d) := 0
    score(Q|d) is calculated by equation (10)
end for
```

proposed TQE model significantly outperforms the baseline models in terms of electiveness.

5.1 Data

The benchmark datasets from Text REtrieval Conference (TREC)[1] (Voorhees et al. 2005) are used to evaluate our proposed model for IR. For each dataset, the queries are taken from the "title" field of TREC topics. A collection of text documents and relevance judgements for each query are involved in each dataset. Table 5 shows the datasets (AP, SJMN, WSJ) details.

5.2 Measures

The effectiveness is assessed by five different measures: average precision of the top K ($K = 5$, and $K = 10$) documents, $F_\beta(\beta = 1)$ measure, Mean Average Precision (MAP). F_1 is a criterion that assesses the effect involving both precision (p) and recall (r),which is defined as $F_1 = \frac{2pr}{p+r}$. The larger the

[1] http://trec.nist.gov/

Table 5: The Statistics of TREC corpora and topics. The number of documents D is given in thousands

Collection	Abbrev	#D	TREC topics
Associated Press	AP	243	51-150
San Jose Mercury News	SJMN	90	51-150
Wall Street Journal	WSJ	173	51-100 151-200

top5, *top10*, MAP or F_1 score, the better the system performs.

5.3 Settings

Firstly, we apply the LDA model to construct topic models with $V = 100$ latent topics for SJMN, 200 topics for WSJ and 300 topics for AP according to the size of each data collection, using the MALLET topic modelling toolkit[2]. Our experiments show that insufficient number of topics will largely bring in abundant expanded patterns into the topic model. We run collapsed Gibbs inference for 1000 samplings, the hyper-parameters of the LDA model are $\alpha = 50/V$ and $\beta = 0.01$. We also pre-process all documents by removing standard stopping words, removing numbers and punctuation.

In the process of generating pattern-based topic representations, the minimum support σ for every topic in each collection is set to 0.05.

For selecting the related topics, the minimum confidence of a topical association rule is set to $\eta = 0.25$.

The trade-off parameters in document relevance ranking are set as $\lambda_p = 0.9$, $\lambda_s = 0.7$ in experiments for all the three datasets.

5.4 Baseline IR models

Three existing IR models are chosen as baseline models in the experiments, including one term-based ranking model BM25 and two state-of-the-art topic-based IR models, which integrate topic model with language model and have achieved relatively successful performance.

5.4.1 BM25

BM25 (Robertson et al. 2004) is one of the state-of-the-art term-based document ranking approaches. In this paper, the original and expanded queries are all scored by BM25 weights. The term weights are estimated using the following equation:

$$W(t) = \frac{tf \times (k+1)}{k_1 \times ((1-b) + b\frac{DL}{AVDL}) + tf} \times \log(\delta + \frac{N - n + 0.5}{n + 0.5}) \quad (11)$$

where N is total number of documents in the collection; n is the number of documents that contain term t; tf is the term frequency; DL and $AVDL$ are the document length and average document length, receptively; and k_1 and b are the parameters, which are set as 1.2 and 0.75 in this paper. Notice that, in this paper, we use the modified BM25 in the part

[2] http://mallet.cs.umass.edu/

of inverse document frequency (idf). The constant δ is added to avoid the negative value of idf for the common terms in the collection, we set $\delta = 1$.

5.4.2 Topical n-gram model

The topical n-Gram model (TNG) proposed in (Wang et al. 2007) automatically and simultaneously discovers topics and extracts topically relevant phrases. It has been seamlessly integrated into the language modelling based IR task (Wang et al. 2007). The generative process can be described as follow:
1) draw discrete ϕ_z from Dirichlet β for each topic z;
2) draw discrete θ_d from Dirichlet α;
3) the difference of TNG from normal LDA model is to draw Bernoulli φ_{zw} from Beta γ for each topic z and each word w; and
4) draw discrete σ_{zw} from Dirichlet δ for each topic z and each word w;
Bernoulli chooses the assignment of the word $w_{d,i}$ to a topic ϕ_j, which is used to determine whether nearby content can be composed as phrases. Readers can refer to (Wang et al. 2007) for more details.

5.4.3 LDA-based Document Model (LBDM)

LDA-based document model smoothing technique (Wei & Croft 2006) utilizes Dirichlet smoothing to smooth $P_{ML}(w|d)$ with $P(w|coll)$, then further smooth the result with $P_{LDA}(w|d)$:

$$P(w|d) = \lambda(\frac{N_d}{N_d+\mu}P_{ML}(w|d) + (1-\frac{N_d}{N_d+\mu}) \\ P_{ML}(w|coll)) + (1-\lambda)P_{LDA}(w|d) \quad (12)$$

where $P(w|d)$ is the maximum likelihood estimate of word w in the document d, and $P(w|coll)$ is the maximum likelihood estimate of the word w in the whole collection. μ is the Dirichlet prior, which is set to 1000 , and $\lambda = 0.7$ in the experiment.

5.5 Results

Table 7: Comparison of the TQE model with TNG and LDBM models. The evaluation measure is average precision. *impr* indicates the percentage of improvement of TQE over best performance of TNG and LBDM

Collection	TNG	LDBM	TQE	*impr*
AP	0.2423	0.2651	0.3390	27.8%
SJMN	0.2122	0.2307	0.2375	3.0%
WSJ	0.2958	0.3253	0.3502	7.7%

We can see that TQE model outperforms the other two baseline models on MAP value, achieves dramatical increase for AP collection which is 27.8% and minimum increase of 3.0% for SJMN. The significant improvement is solid evidence that supports the hypothesis.

From Table 6, we can see that the number of net queries in AP, SJMN and WSJ are 99, 94 and 100, respectively. Among the valid queries, 80.8% (80/99) queries in AP, 90.4%(85/94) queries in SJMN and 83.0%(83/100) in WSJ can be expanded by our proposed approach. This demonstrates that topical patterns that are discovered by our model are the interpretable representations and also effective at selecting related topics which can be further used in query expansion. This results also support the hypothesis.

In the same table, we can find that in the expanded queries across the three collections, 23.5%-37.5% queries get improved under evaluation measure *top*5 ,25.9%-42.4% queries performs better under evaluation *top*10, 53.0%-66.3% queries under MAP and 51.8%-66.3% under F1 get improved. Although the improving number of queries under *top*5 is smallest but its average gain over all improved queries are the highest, while MAP and F1 evaluation have large number of improved queries but the improvement value is relatively the lowest. To summarise the performance, our proposed TQE consistently performs excellently across all evaluation measures, which proves the hypothesis that it is effective at query expansion.

6 Discussion

The results on benchmark TREC datasets show that this technique can result in major improvements for a large proportion of queries. The reason behind is that using topical patterns are to expand the query can obtain more accurate and semantically related terms. The significant improvements are also contributed by the divisions of focused queries and scattered queries at a topic level and different ranking mechanisms based on the different queries for ranking documents.

6.1 Topical Patterns for Query Expansion

Instead of using individual words as representation of topics, we use stronger terms relationships by topical patterns to select related topics as well as expanded terms from the related patterns. Pattern-based representations are effective at interpreting the semantic meaning, thus the topical pattern based expansion is more trustful in terms of semantics . At ranking stage, the topical patterns are also effective because they can deliver specific weightings with patterns (X) and their relevance ($f^i_{q_w}(X)$).

But based on our experimental experience, if the number of trained topics is too small (i.e., $V = 100$ for the larger collection AP) which can't fit the topic partitions within a collection, the performance will drop sometimes even worse than original BM25. The main reason is that very low dimension of topics leads to abundant patterns, which causes the related patterns contain many noises. Therefore, the number of topics directly affects the correctness of query expansion by topical patterns.

The query expansion approach TQE is quite flexible indeed. If the query itself is short, the related topics will complement the features; and if they query is verbose, the optimization of related topics will converge to focused topics and decrease the effects of noisy terms. Consequently, topical pattern for query expansion is a good strategy for IR and it also solve the mentioned problems.

6.2 Complexity

Many topical IR models require feedbacks which is less efficient in retrieving documents after user's inputs. Since our proposed model only need offline training once which requires no extra online processing time, it performs significantly outstanding on retrieval effectiveness and efficiency.

As discussed in Section 4.4, there are two algorithms in the proposed model, i.e. offline training and Online retrieval. The complexity of the online retrieival is related to the number of terms in a query

Table 6: Improvements of using TQE model compared with only using original queries. "Expanded Queries" column indicates number of the queries can be expanded . For each evaluation, "impr." shows the number of queries get improved, and "avg gain" means the average improvement over the improved queries.

Collection	Net Queries	Expanded Queries	*top*5		*top*10		MAP		F1	
			impr.	avg gain	impr.	avg gain	impr.	avg gain	impr.	avg gain
AP	99	80	30	0.238	36	0.196	53	0.099	53	0.070
SJMN	94	85	20	0.217	22	0.10	50	0.031	48	0.037
WSJ	100	83	23	0.236	28	0.123	44	0.072	43	0.044

and the expanded terms. However, the number is always small and the calculation time is often acceptable.

For offline training, the proposed pattern-based topic modelling methods consist of two parts, topic modelling and pattern mining. For the topic modelling part, the initial user interest models are generated using the LDA model, and the complexity of each iteration of Gibbs sampling for the LDA is linear with the number of topics (V) and the number of documents (N), i.e. $O(V * N)$ (Wei & Croft 2006). For pattern mining, there is no specific quantitative measure for the complexity of pattern mining reported in relevant literature. But the efficiency of the FP-Tree algorithm (Han et al. 2007) for generating frequent patterns has been widely accepted in the field of data mining and text mining. The transactional datasets used in the TQE model are generated from the topic representations produced by the LDA model rather than the original document collections. The patterns used to represent topics are generated from the words which are considered to represent the document topics by the LDA model. These words are part of the original documents, whereas other pattern mining models generate patterns from the whole collection of documents.

Moreover, the TQE model combines the topic modelling and pattern mining linearly. Thus, in summary, the complexity of the TQE model can be determined by topic modelling or pattern mining. In most cases, the complexity of the TQE model would be the same as pattern mining since, in general, the complexity of pattern mining is greater than that of topic modelling. As them name indicates, offline training can be conducted off-line which means that the complexity of the offline training part will not affect the efficiency of the proposed IR model.

7 Conclusion and Future Work

This paper presents an innovative weighted query expansion for information retrieval including topical-pattern based query expansion and document relevance ranking. The TQE generates pattern-based topic representations for every topic in a collection. With these semantic topical patterns, a query belongs to certain topics or uncertain topics can be expanded with topic-related patterns. For the document ranking, the TQE estimates the relevance from the aspects of determining certainty of topics at general level, additionally analysing related patterns with more specific features. The proposed model has been evaluated by using the TREC collections for IR task. Compared with the state-of-the-art models, the TQE demonstrates excellent strengths both on query expansion and document relevance ranking.

The proposed model automatically generates discriminative and semantic rich representations for modelling topics and queries by combining topic modelling techniques and data mining techniques. The following work in future could take relevance feedback into consideration, the related topics and patterns discovered by the TQE will be assigned with more precise weights.

References

Andrzejewski, D. & Buttler, D. (2011), Latent topic feedback for information retrieval, *in* 'Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 600–608.

Bai, J., Song, D., Bruza, P., Nie, J.-Y. & Cao, G. (2005), Query expansion using term relationships in language models for information retrieval, *in* 'Proceedings of the 14th ACM International Conference on Information and Knowledge Management', CIKM '05, ACM, New York, USA, pp. 688–695.
URL: *http://doi.acm.org/10.1145/1099554.1099725*

Bayardo Jr, R. J. (1998), Efficiently mining long patterns from databases, *in* 'ACM Sigmod Record', Vol. 27, ACM, pp. 85–93.

Blei, D. M., Ng, A. Y. & Jordan, M. I. (2003), 'Latent dirichlet allocation', *the Journal of machine Learning research* **3**, 993–1022.

Cao, G., Nie, J.-Y., Gao, J. & Robertson, S. (2008), Selecting good expansion terms for pseudo-relevance feedback, *in* 'Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval', ACM, pp. 243–250.

Collins-Thompson, K. (2009), Reducing the risk of query expansion via robust constrained optimization, *in* 'Proceedings of the 18th ACM conference on Information and knowledge management', ACM, pp. 837–846.

Gao, Y., Xu, Y. & Li, Y. (2013), Pattern-based topic models for information filtering, *in* 'Data Mining Workshops (ICDMW), 2013 IEEE 13th International Conference on', pp. 921–928.

Gao, Y., Xu, Y. & Li, Y. (2014), A topic based document relevance ranking model, *in* 'Proceedings of the companion publication of the 23rd international conference on World wide web companion', International World Wide Web Conferences Steering Committee, pp. 271–272.

Gao, Y., Xu, Y., Li, Y. & Liu, B. (2013), A two-stage approach for generating topic models, *in* 'Advances in Knowledge Discovery and Data Mining', Springer, pp. 221–232.

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.

Kumaran, G. & Carvalho, V. R. (2009), Reducing long queries using query quality predictors, *in* 'Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval', ACM, pp. 564–571.

Lavrenko, V. & Croft, W. B. (2001), Relevance based language models, *in* 'Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval', ACM, pp. 120–127.

Liu, Z., Huang, W., Zheng, Y. & Sun, M. (2010), Automatic keyphrase extraction via topic decomposition, *in* 'Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing', Association for Computational Linguistics, pp. 366–376.

Lv, Y. & Zhai, C. (2011), Lower-bounding term frequency normalization, *in* 'Proceedings of the 20th ACM international conference on Information and knowledge management', ACM, pp. 7–16.

Maxwell, K. T. & Croft, W. B. (2013), Compact query term selection using topically related text, *in* 'Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval', ACM, pp. 583–592.

Mei, Q., Zhang, D. & Zhai, C. (2008), A general optimization framework for smoothing language models on graph structures, *in* 'Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval', ACM, pp. 611–618.

Metzler, D. & Croft, W. B. (2007), Latent concept expansion using markov random fields, *in* 'Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval', ACM, pp. 311–318.

Robertson, S., Zaragoza, H. & Taylor, M. (2004), Simple bm25 extension to multiple weighted fields, *in* 'Proceedings of the thirteenth ACM international conference on Information and knowledge management', ACM, pp. 42–49.

Steyvers, M. & Griffiths, T. (2007), 'Probabilistic topic models', *Handbook of latent semantic analysis* **427**(7), 424–440.

Voorhees, E. M., Harman, D. K. et al. (2005), *TREC: Experiment and evaluation in information retrieval*, Vol. 63, MIT press Cambridge.

Wang, X., McCallum, A. & Wei, X. (2007), Topical n-grams: Phrase and topic discovery, with an application to information retrieval, *in* 'Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on', IEEE, pp. 697–702.

Wei, X. & Croft, W. B. (2006), Lda-based document models for ad-hoc retrieval, *in* 'Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval', ACM, pp. 178–185.

Yi, X. & Allan, J. (2009), A comparative study of utilizing topic models for information retrieval, *in* 'Advances in Information Retrieval', Springer, pp. 29–41.

Zhai, C. & Lafferty, J. (2001), Model-based feedback in the language modeling approach to information retrieval, *in* 'Proceedings of the tenth international conference on Information and knowledge management', ACM, pp. 403–410.

Content Based Image Retrieval Using Signature Representation

Chathurani N.W.U.D., Geva S., Chandran V., Chappell T.
School of Electrical Engineering and Computer Science
Queensland University of Technology
Brisbane, Australia

(n.chathu@qut.edu.au, s.geva@qut.edu.au, v.chandran@qut.edu.au,
t.chappell@connect.qut.edu.au

Abstract

Retrieving relevant images from a large, diversified collection using visual queries (image content) as search argument is a challenging and important open problem. It requires an efficient and effective content-based image retrieval (CBIR) system. Image representation has a profound effect on the performance of CBIR. This paper presents a CBIR system based on a novel image representation using a new approach to the generation of image signatures (CBIR-ISIG). Image signatures are generated by applying random indexing (RI) to a Bag-of-visual Words (BoW) representation of the images. RI is an efficient and scalable approach to dimensionality reduction, based on random projection which avoids the computational cost of matrix factorization. Most importantly, it can be performed incrementally as new data arrives, as is crucial for online systems. The retrieval quality of the proposed approach is evaluated using a benchmark dataset for image classification (a subset of the Corel dataset). The proposed approach shows promising results with comparable retrieval quality to state of the art approaches while retaining the benefits of the highly efficient representational scheme.

Keywords: Content based image retrieval, Image signature, Random Indexing, Invariance, Bag-of-Words.

1 Introduction

The development of the Internet and increased availability of image capturing devices have enabled collections of digital images to grow at a fast pace in recent years and to become more diverse. This created an ever growing need for efficient and effective image browsing, searching, and retrieval tools. Retrieving relevant images accurately to satisfy an information need from a large, diversified collection using visual queries is a challenging and important problem to address. Despite many years of research in this area, an effective general solution still eludes researchers. The most familiar CBIR system in wide use today is offered by Google Images, where a user can upload an image as a query, and the system responds with similar images, based on content. The performance of this system can only be described as less than satisfactory for all but a very small fraction of images – when it works well, it is almost invariably for very famous images, but even then it often fails. Figure 1 shows an example of a failed search for a query image (Eiffel tower) in Google Images search. All the while Google Images are quite effective for text based search using accompanied image text annotations, it is not effective for (visual) content-based image retrieval.

Selecting an appropriate image representation is the most important factor in implementation of an effective CBIR system. Among the many image representations that have been studied, the BoW approach is one of the most promising (Yuan, et al. 2011 and Mansoori, et al. 2013). The BoW approach is flexible with respect to geometry, deformations and viewpoint and it provides vector representation for sets. Finally it gives a compact summary of image content. This research adapts the BoW approach and presents an efficient content based image retrieval system using image signatures (CBIR-ISIG) and, introduces a new approach to image signature definition. Image signatures are generated by applying random indexing on a BoW representation of the image, using Topsig (Geva and De Vries 2011). This improves the computational efficiency of conventional BoW CBIR approaches. The system performance is evaluated by the use of a well categorized subset of the standard Corel dataset and is shown to produce superior results when compared with other independently developed and tested approaches.

There are numerous examples in the literature where signatures are used to detect near-duplicates in image processing and text processing at high speed. Topsig is a tool that is used to generate and search signatures, with response time at the millisecond scale to search millions of documents (Chappell, et al. 2013). Signature representation is a concise representation of real vectors, which help to reduce the storage space (eg: in this research around 5840 bytes of a real valued image feature vectors are represented by 1024 bytes). It can be defined as a representation of documents, images and other searchable abstract objects as binary strings of fixed length (Chappell, et al. 2013). The motivation behind this research is to find a representation that is efficient to search while retaining retrieval quality by feature fusion and signatures.

In image retrieval there are essentially two approaches. The first is the text based (TBIR) approach. Here the user describes the image content with a textual specification of the image content, such as tags. However the representation of images using text was found to be too difficult: requiring excessive human effort to support, time consuming and expensive, incapable of describing

Figure 1: Retrieval results for an example query image (Eiffel tower) in Google image search

rich image features, and very much dependent on human perception. To overcome the limitations of TBIR, content based image retrieval (CBIR) was introduced, allowing queries to be specified visually. In CBIR image features are extracted and indexed automatically to support storage and retrieval of images with visual queries.

Over the last 30 years significant attention has been paid to CBIR. Extensive research has been conducted (Liua, et al. 2007) to develop sophisticated algorithms to extract low-level image features such as colour, shape, texture, edges, point of interest, and spatial relationships. These algorithms then measure the similarity between pairs of images based on image feature vectors. Much work had been dedicated to exploring and provision of solutions to the problems of image rotation, translation and scale invariance (RTSI). However, these algorithms cannot adequately model image semantics and have many limitations when it comes to dealing with broad content image databases, especially with regards to response time and retrieval accuracy. CBIR has been assessed comprehensively in (Liua, et al. 2007).

In this paper authors offer a different approach to the construction of image signatures. This starts from a set of standard image features, use vector quantisation to generate a BoW representation. Sub-image feature sets as well as full image feature set are further used. Then random indexing is applying to fuse the various feature sets and then the representation used to store and search images. This approach is described in more detail later in this paper.

The rest of the paper is organized as follows. Section 2 outlines the previous work done on CBIR. Section 3 describes an overview of the system and the proposed method. Section 4 provides details of the experiment and the results obtained. Conclusions drawn from the research findings are included in section 5.

2 Previous Work

CBIR systems measure visual similarity between a query image and multiple database images and then retrieve the top ranked images using various similarity measures. These similarity measures are computed on extracted features (Colour, Texture, and Shape) from images. In (Liua, et al. 2007) a comprehensive assessment of prior work with CBIR was performed.

Most of the systems (Saad, et al. 2011) have used global feature representation whereas some other systems (Hiremath and Pujari 2007, Mansoori, et al. 2013, Takala et al. 2005) have used local feature representation. In global representation, features are extracted from the whole image while in local representation features are extracted from either, segmented regions, or from a regular grid or points of interest. Local representation is applied to a wide range of CBIR systems and applications to achieve robustness. However a precise image segmentation method that can be applied to general image collections has not yet been found. In the absence of an accurate segmentation approach, a sliding window approach over location and scale has shown to be quite effective (Hiremath and Pujari 2007).

Feature extraction is an important step largely responsible for the performance of CBIR. Colour is one of the most cognitively significant features in an image and it is the simplest and most extensively used feature in image retrieval (Hiremath and Pujari 2007, Mansoori, et al. 2013, Saad, et al. 2011) (notwithstanding human ability to work well with Greyscale images). The texture feature has been used in CBIR systems in different ways (Hiremath and Pujari 2007, Takala et al. 2005, Yuan, et al. 2011), and these feature extraction methods can be classified as statistical, spectral or structural. Texture describes the content of many real world images and provides important characteristics for surface and object identification. Similar to the aforementioned features, shape is also an important feature in CBIR (Hiremath and Pujari 2007, Saad, et al. 2011), especially when dealing with objects that have clear shapes. Shape features are classified as both boundary-based and region-based methods. Different systems use different features and combinations of these features. In this approach BoW approach uses all these extracted image features to generate visual vocabularies.

During the past decade, the BoW approach has achieved popularity in the fields of classification and retrieval in CBIR (Yuan, et al. 2011, Mansoori, et al. 2013) because of its simplicity and good relative performance. It is also suitable for large databases as it scales efficiently to large collections. This approach was introduced by Sivic and Zisserman (Sivic and Zisserman 2003) to the computer vision community and it was inspired by the BoW model in text document retrieval. The visual vocabulary or

visual codebook is formed by clustering image features that are extracted from images in the database. Firstly, similar features are gathered together where each cluster centre stands for a visual word. After that, feature vectors are mapped to those visual words and each image is represented as a histogram of visual words. Spatial information has been introduced to the BoW approach (Lazebnik et al.2006) to improve the results. However, like the other representations, the BoW approach still requires dealing with high-dimensionality data, which presents scalability challenges.

High dimensional indexing is one of the prevailing challenges in CBIR. Quite a few systems have addressed the problem of high dimensionality features. The well-known techniques used in CBIR include wavelet transform (Elharar et al. 2007), discrete cosine transform (DCT) (Elharar et al. 2007), latent semantic analysis (LSA) (Gorman et al. 2006), principal component analysis (PCA) (Banda et al. 2013), singular value decomposition (SVD) (Banda et al. 2013) and locality sensitive hashing (LSH) (Gorman et al. 2006). They are designed to reduce the dimensionality of feature vectors while maintaining the information in the descriptors as much as possible. Features encoded in a lower dimensional space must contain enough information to usefully distinguish between classes of images and perform well. Random indexing (RI) has been used and has shown great promise as a dimensionality-reduction technique in text retrieval (De Vries et al. 2009, Gorman et al. 2006). Compared to other methodologies RI has low computational cost, lower complexity, competitive accuracy, and most importantly it is an incremental approach (Magnus 2005, De Vries et al. 2009). Reducing the dimensionality of features has a considerable effect on the way that feature vectors are stored and retrieved. The reduced dimensional space provides a compressed representation of the original feature space.

The BoW representation as used in the literature is generally a histogram of visual words (Mansoori, et al. 2013). The BoW approach that is presented in this study is different however, in that it literally considers images as text documents and represents them as sets of visual-words (represented as symbolic tokens). The process developed in this research moves image retrieval further into the text retrieval domain. In the literature different techniques are adapted to reduce the dimensionality of images and this paper introduces RI as a dimensionality reduction technique used in text retrieval, to CBIR.

The objective of this research is to develop a novel CBIR system in order to achieve faster and adequately precise image classification and retrieval. An extended BoW feature representation method is developed, therefore utilizing subdivisions of each image into equally sized non overlapping tiles to generate a codebook. In order to achieve this objective, RI is applied to the BoW representation of images, introducing a new approach to the definition of image signatures. This BoW representation is also novel in the way that it translates an image into a bag of visual words document. Empirical evaluation is performed on a standard Corel dataset to validate the performance of this method against other independently evaluated methods.

3 Overview of CBIR-ISIG System

Feature extraction plays a major role in CBIR. The extracted features are used to index the images in CBIR. Comparisons of defined techniques indicate that a single feature for image retrieval is not an adequate solution. General images may have some colour images, images with texture, images of objects, and so on. Therefore it is concluded that multiple feature representation for image retrieval is necessary and this study proposes a novel approach that uses a combination of low-level features including colour, texture, shape, and GIST. The main reason for the selection of these descriptors is to address retrieval in a heterogeneous database of images. The following section defines features with the feature descriptors that are critical for accurate retrieval. These particular features are selected because of their reported performance and variation of feature description as described in the literature.

3.1 Colour Feature

Colour plays an important role in image retrieval and has been widely considered in feature extraction in the literature (Liua et al. 2007). There are a number of colour descriptors and the three most popular colour descriptors; the colour histogram, colour moments and colour coherence vector are selected for this study. Colour histograms are efficient and insensitive to small changes in camera view point. Colour histogram was adapted from (Qiu 2002) as it achieved better retrieval results using the YCbCr colour space, providing a closer match with human perception. The process of generating the histogram is as described in (Qiu 2002), and is most comparable to the colour set approach. Colour moments overcome the quantization effects in histograms and gives colour distribution. First order original moment, second order central moment, and third order central moment are calculated. Colour coherence vector (Pass 1996) includes spatial information; it classifies each pixel in a colour bucket as coherent or incoherent. Eight colour components are used in this approach.

3.2 Texture Feature

Notwithstanding the fact that texture is not well defined, it is very helpful to describe real world images. The well-known Gabor wavelet, Wavelet transforms and Edge histogram descriptor are selected as texture descriptors in this study. The Gabor features are widely adapted and have performed well in CBIR and they are also used in this proposed system. In this paper five scales and eight orientations are used. The rotation and scale invariance property is achieved by simple circular shift operation proposed in (Rahmanan, et al. 2011). Mean and standard deviation of each filter are used as a feature vector. The Wavelets transform provides a good multi resolution tool for texture description and allows the representation of a texture having various spatial resolution. It effectively describes both global and local information. Daubechies wavelets are chosen because they are better for general-purpose images search (Manthalkar, et al. 2003). Here decomposition is done up to three levels and each time the low frequency sub band is decomposed. The mean and standard deviation of each sub band are computed. 2003) is used to achieve the rotation and scale invariance

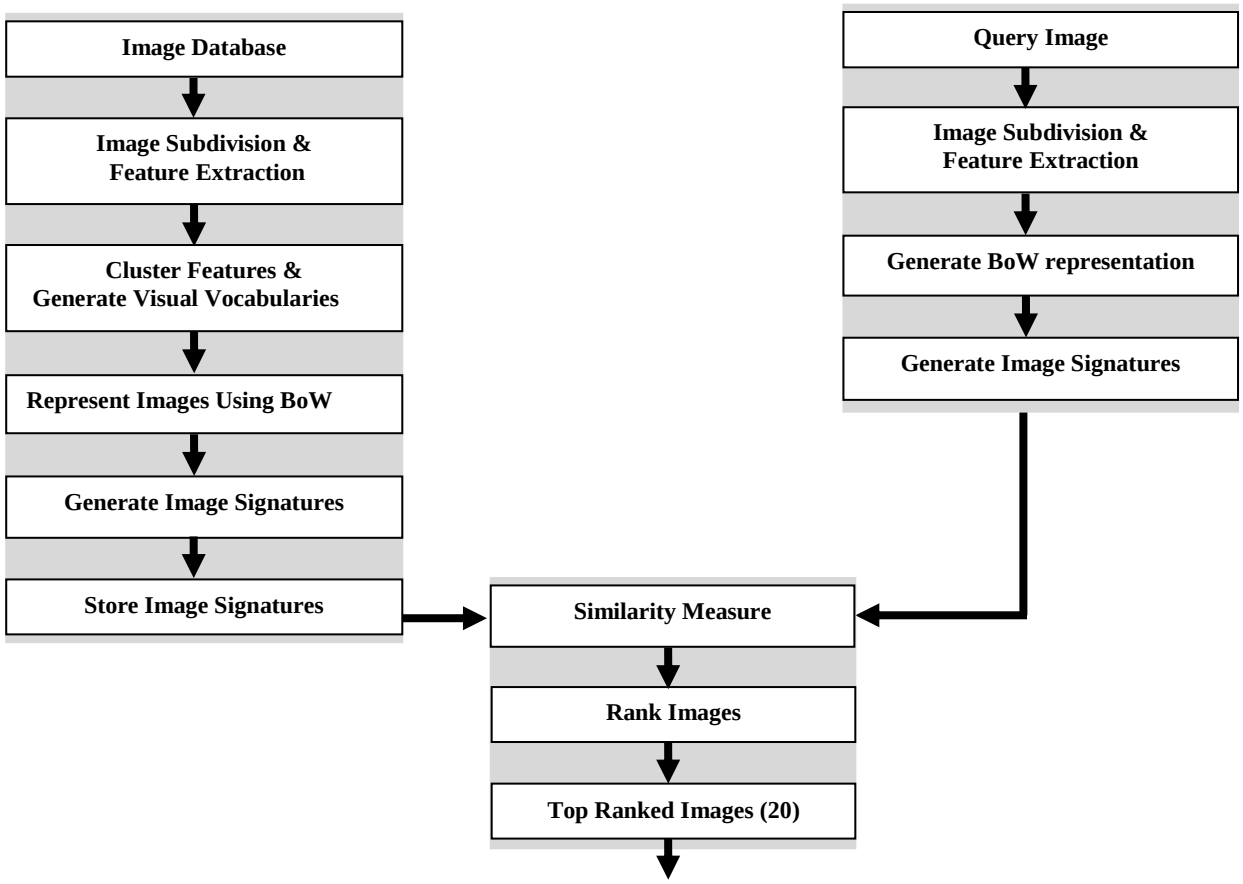


Figure 2: An Overview of the CBIR-ISIG System

The simple operation, proposed in (Manthalkar, et al. property. The Edge histogram descriptor has shown reliable performance and it effectively describes heterogeneous textures (Agarwal, et al. 2013). It captures the spatial distribution of edges and helps to extract different textures using five filters.

3.3 Shape Feature

The most common Generic Fourier descriptor and the invariant moments are used for shape description. The Generic Fourier descriptor is a region based method and suitable for general image retrieval. It is translational, rotational, scale invariant and robust to noise and occlusion too (Minaqiang et al. 2008). Four radial frequencies and 15 angular frequencies are used here. Although this is computationally expensive, it generates excellent retrieval performance. The invariant moment is an invariant feature and widely used for shape retrieval task. It gives compact representation on pixel distribution of a shape image. Moments are limited to seven by the calculation that use of higher order moments result in being sensitive to noise thus cause hindrance to accuracy.

3.4 GIST

The GIST descriptor describes the spatial envelope of the image and has shown good retrieval performance (Torralba, et al. 2008) in the literature. So the GIST feature used in this approach as well. This system uses code developed for the SUN database (Jianxiong, et al. 2010) where greyscale with eight orientations and three scales are used.

All the above features are adapted in CBIR-ISIG with the aim of improving retrieval performance. MATLAB is used for feature extraction. Each image is partitioned into 9 equal sized sub-images, by dividing each image into 3 by 3 grid. A family of image features, like shape, texture, and colour, characterizes every sub-image. In this research it was intended to provide some degree of spatial invariance through these sub-images. Even though only one signature is generated at the end, it is derived from a combination of sub-image signatures.

Different researchers try to achieve rotation translation scale invariance (RTSI) in different ways. Some low level features already have these properties and adapted existing techniques, but these are still low-level features. Rather than relying exclusively on explicit RTSI features, this proposed approach also relies on the BOW invariance property – sub-images are not indexed by position (in the same manner that the positions of words in text are not used in BoW text indexing.) Underlying features are the basic features like Colour histogram, Wavelet transform, GIST.

It is necessary to adapt the BoW approach used in document retrieval in order to apply it in an analogous way to images. After extracting low level features it is necessary to select an appropriate multidimensional indexing algorithm to index them. Clustering is a promising technique among indexing techniques. It is

first necessary to cluster the image features in order to obtain discrete representation of feature sets. K-means is one of the simplest and best-known unsupervised clustering algorithms that can be easily implemented for feature vocabulary generation.

After extracting features, independent visual vocabularies are generated. In order to achieve this, all the features are clustered, separately into groups where similar feature vectors are placed together. Here the size of the vocabulary is the number of clusters generated and each cluster centre is considered as a visual word in the vocabulary. Then each sub-image and the full image are represented by visual words from these vocabularies through codebook lookup of each raw image feature. The images are represented symbolically, just like text, by using the codebook label of each cluster as a visual word to encode the feature. Other approaches which use BoW to represent images use histogram representation, by counting how many times each word appears in an image. However in our approach an image is regarded as analogous to a document and sub-image is regarded as analogous to a paragraph in a document.

If a BoW representation of a full image is denoted as Z it can be defined as,

$Z = X_a$, $a \in \{1,....,M\}$,Where X_a is a sub-image representation and M is the number of sub-images in an image;

$X = \{f_i \_ c_j\}$, $i \in \{1,....,N\}$ and $j \in \{1,.....,K\}$,

Where f_i is a local feature, N is the number of features used in the system and each feature is given a number, c_j is a cluster number (to which cluster that word belongs) and K is the vocabulary size. Each feature is given a number to denote it within the representation.

Image signatures are generated for sub-images as well as the full image. This is done to significantly reduce the dimensionality of the representation. The descriptors' dimensionality is important and heavily influences the complexity of the similarity measure in retrieval, and the memory requirements for storing the descriptors. There are several approaches for dimensionality reduction including RI (Magnus 2005), which is an efficient, scalable and incremental approach, based on random projection to avoid the computational cost for matrix factorization (Geva and De Vries 2011). One prime advantage of RI is that it can work directly with symbolic features, for instance, words in documents. RI is used effectively in text retrieval applications to reduce the dimensionality of documents without significant degradation in retrieval quality. RI can produce binary object signatures. The representation of objects as bit vectors lends itself to efficient processing, with low level bitwise operations supported on all conventional processor architectures. Most importantly, RI can be performed incrementally aligning with the new data arrival, as is crucial for online systems. Therefore in our approach RI is used for dimensionality reduction and to create image signatures. This allows the feature vector space to be reduced in dimensionality without expensive factorization such as, for instance latent semantic analysis (LSA) techniques. Seeding a pseudorandom number generator with the feature hash, and then generating a feature signature is used to create pseudo-random sparse ternary feature vector having values from {-1,0,+1}. A common choice with RI is to assign the proportion of vector elements with each value {-1,0,+1} to be respectively 1/6, 2/3 and 1/6.

All feature vectors in an image, of the entire image and of each sub-image, are then summed to produce a single image index vector. The image index vector is then squashed into a binary signature by assigning 1 bits to positive values and 0 bits to negative values. Similar images that share similar features will have similar signatures. Image signatures can then be compared for similarity by taking the bitwise (Hamming) distance between them. This technique can be used as a highly efficient replacement for a cosine similarity calculation in the original feature vector space. This approach uses a signature search-engine for searching. The motivation for using signatures to represent images comes from the fact that computation time quickly becomes a bottleneck when dealing with large databases and signature search engines can retrieve results from web-scale collections in milliseconds (Chappell, et al. 2013). Topsig (Geva and De Vries 2011) which is available in open source, is used to generate and search signatures in our CBIR system. This paper is concerned with identifying the ability of our approach to represent and then find images, rather than the signature matching and searching mechanism itself, so in fact it allows us to use any signature search engine regardless of specifications. Our concern is with how well the signatures would represent the images. The proposed signature based approach was evaluated using a gold standard benchmark i.e. subset of Corel dataset which is described in section 4.

An overview of the proposed CBIR system is depicted in Figure 2.

4 Evaluation

The Wang dataset of 1000 images was used for both evaluation of the system and comparison with the other systems. The Wang dataset of 1000 images is a subset of manually selected images from Corel image database and it was previously used in CBIR as a standard dataset for evaluation purposes; hence, it is convenient to re-use here since it provides a baseline for comparison with other independently developed and tested approaches. It consists of 10 classes with 100 images in each and they are African people, Beaches, Buildings, Buses, Dinosaurs, Elephants, Flowers, Horses, Mountains, and Food. These images are JPEG with the resolution of 384x256 or 256x384.

During the evaluation features are extracted from the query images and represented using nearest cluster centres as in section 3. Then an image signature is generated as described in section 3. Finally the query signature is compared with that of the image database and top k images are retrieved from the database. Hamming distance is used for the similarity measure.

The most common evaluation measure in information retrieval is precision and it is used to evaluate the CBIR-ISIG system. Precision is the fraction of retrieved images that are relevant to the query and it is defined as;

(a) (b)

Figure 3: Search Results of CBIR-ISIG system for two queries (Query image is the top left most one)

Class	2005 [1]	2007 [2]	2011 [3]	2011 [4]	2013 [5]	CBIR-ISIG
Africans	0.23	0.48	0.57	**0.90**	0.70	0.75
Beach	0.23	0.34	0.58	0.38	0.28	**0.64**
Building	0.23	0.36	0.43	**0.72**	0.56	0.50
Bus	0.23	0.61	0.93	0.49	0.84	**0.85**
Dinosaur	0.23	0.95	0.98	1.00	0.81	**1.00**
Elephant	0.23	0.48	0.58	0.39	0.58	**0.70**
Flower	0.23	0.61	0.83	0.56	0.55	**0.95**
Horse	0.23	0.74	0.68	0.87	0.87	**0.94**
Mountain	0.23	0.42	0.46	0.45	0.48	**0.58**
Food	0.23	0.50	0.53	**0.87**	0.66	0.69
Average	0.23	0.55	0.66	0.66	0.63	**0.76**

Table 1: Average Precision (AP) of each class along with whole dataset with performance in the literature (AP for the top 20 images)

Class	2000 [6]	2002 [7]	2008 [8]	2009 [9]	2012 [10]	CBIR-ISIG
Africans	0.48	0.47	0.48	0.45	0.49	**0.50**
Beach	0.33	0.33	0.34	0.35	0.40	**0.45**
Building	0.33	0.33	0.33	0.35	**0.39**	0.33
Bus	0.36	0.60	0.52	0.60	0.58	**0.62**
Dinosaur	0.98	0.95	0.95	0.95	0.96	**0.98**
Elephant	0.40	0.25	0.40	**0.60**	0.50	0.44
Flower	0.40	0.63	0.60	0.65	0.75	**0.75**
Horse	0.72	0.63	0.70	0.70	**0.80**	0.68
Mountain	0.34	0.25	0.36	0.40	**0.40**	0.36
Food	0.34	0.49	0.46	0.40	**0.51**	0.41
Average	0.47	0.49	0.51	0.55	0.55	0.55

Table 2: Average Precision (AP) of each class along with whole dataset with performance in the literature (AP for the top 100 images)

$$\text{Precision} = \frac{|\{\text{Relevant images}\} \cap \{\text{Retrieved images}\}|}{|\{\text{Retrieved images}\}|}$$

Average Precision $P(c)$ $(1 \le c \le 10)$ is taken for each class as follows.

$$P(c) = \frac{1}{N}\sum_{i=1}^{N} p(i)$$

Where $p(i)$ is the average precision of i^{th} query image and N is the number of images used for evaluation.

When N=20, each class has achieved more than 50% accuracy in the dataset and most of the classes have achieved the highest average precision among the compared systems. When N=100 performance of this system is relatively reduced, but still performs well by comparison with other systems. The systems which are compared are represented by year of publication and reference. Maximum average precision values of each class are highlighted Table 1 shows the results of CBIR-ISIG compared with the other systems (average precision for the top 20) and it shows the CBIR-ISIG system generates better results using this approach. They are not statistically significantly better (at 95% significance level), but averages are higher. In addition Table 2 shows the results of CBIR-ISIG compared with other systems (average precision for the top 100). Here total average precision reaches a highest in the CBIR-ISIG system for the top 20 and average performance for the top 100. Bold values show the highest among the compared systems.

References for the compared systems as given bellow for Table 1 and Table 2.

[1]- (Takala, et al. 2005), [2]- (Hiremath and Pujari 2007), [3]- (Yuan, et al. 2011), [4]- (Saad. et al. 2011), [5]- (Mansoori, et al. 2013) [6]- (Li, et al. 2000), [7]- (Chen 2002), [8]- (Hiremath and Pujari 2007), [9]- (Banerjee, et al. 2009), [10]- (Chowdhury 2012).

Figure 3 shows the top 20 images for a given query. The query image is at the top left. (a) sample from flowers class (19 out of 20) and (b) horses class (20 out of 20).

Our approach inherits the scalability of the particular signature search engine which was used here . This signature search engine (Geva and De Vries 2011) was reported to be capable of searching millions of signatures in milliseconds and so our approach can operate at that speed when the same signature size is used. However as this approach inherits the properties of signatures it will improve retrieval speed and reduce the memory size required to store features. But there is a trade-off between efficiency and accuracy. Signature size can be selected according to our need. Table 3 depicts the changes in AP with signature size. In this research 8192 bits (1024 bytes) signature size is chosen as it is

Class	Signature Size (in bits)										
	64	**128**	**256**	**512**	**1024**	**2048**	**4096**	**8192**	**16384**	**32768**	**65536**
Africans	0.40	0.55	0.64	0.70	0.70	0.73	0.73	0.75	0.75	0.74	0.74
Beach	0.31	0.47	0.51	0.59	0.61	0.63	0.65	0.64	0.64	0.64	0.64
Building	0.35	0.38	0.40	0.40	0.46	0.48	0.47	0.50	0.48	0.48	0.48
Bus	0.55	0.76	0.77	0.80	0.81	0.84	0.84	0.85	0.84	0.84	0.84
Dinosaur	0.88	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Elephant	0.31	0.59	0.57	0.62	0.65	0.69	0.69	0.70	0.70	0.70	0.70
Flower	0.71	0.90	0.95	0.94	0.94	0.94	0.94	0.95	0.95	0.95	0.95
Horse	0.65	0.83	0.88	0.91	0.91	0.92	0.93	0.94	0.93	0.93	0.93
Mountain	0.29	0.46	0.40	0.46	0.58	0.53	0.56	0.58	0.58	0.58	0.58
Food	0.32	0.55	0.54	0.64	0.65	0.69	0.68	0.69	0.69	0.69	0.68
Average	0.477	0.658	0.666	0.706	0.731	0.745	0.749	**0.76**	0.756	0.755	0.754
	0.48	0.66	0.67	0.71	0.73	0.74	0.75	**0.76**	0.76	0.76	0.75

Table 3: Average Precision (AP) of each class along with whole dataset with different signature size (AP for the top 20 images)

sufficient to demonstrate improvement. It only accommodates around 20% of the real valued feature vector in size, but even if use 128 bits (16 bytes) the results are respectable. Signature size can be chosen according to the requirement. There is a significant computational cost in generating the raw image features Feature extraction involves relatively heavy image processing, which is time consuming. This cost is not unique to our system and is incurred by any system that uses the same feature set. The cost of image signature generation comes from clustering features in order to generate a BoW representation, and the cost of generating signatures from the BoW representation. This entire process is done only once during indexing and can be trivially parallelised since there is no dependence between images. At search time the process is performed only on the search argument (a single image). and it is fast enough to support immediate response to the user.

The speed at which a signature can be generated is limited by the complexity of feature extraction, essentially conventional image processing and not by random indexing which consumes negligible amount of time, by comparison. In our computational configuration the average time needed to extract the above features from a 64 by 64 image is 0.1347 seconds, using Windows Core i5, 1.8GHz, and using MATLAB for image processing.

5 Conclusion

This paper presented a novel approach to represent images using image signatures which are derived by applying RI to a BoW representation of images. These image signatures improve the retrieval speed and reduce the requirement of memory for storage. The CBIR-ISIG system shows more than 50% average precision for the top 20 images in each class and achieved 75% average precision which is best among all the compared system against a standard subset of the Corel dataset. The results however are still limited and the approach will be tested with larger datasets in future work. The most obvious applications of this approach are large scale CBIR for heterogeneous collections and near-duplicate detection.

6 Reference

Yuan, X. Yu, J. Qin, Z. and Wan, T. (2011): A SIFT-LBP Image Retrieval Model Based on Bag-of-Features. *Proc. IEEE ICIP 18th International Conference on Image Processing*, Brussels, Belgium, 1061-1064, Brussels, Belgium.

Mansoori, N.S. Nejati, M. Razzaghi, P. and Samavi, S. (2013): Bag of visual words approach for image retrieval using colour information. *Proc. ICEE 21st Iranian Conference on Electrical Engineering*, Mashhad, 1-6, IEEE.

De Vries C. M., V. Lance D. V. and Shlomo G. (2009): Random Indexing K-tree. *Proc. ADCS of the 14th Australian Document Computing Symposium*, Sydney, Australia, 1-8, University of Sydney.

Liua, Y. Zhanga, D. Lua, G. and Mab, W.Y. (2007): A Survey of Content-based Image Retrieval with High-Level Semantics. *Journal of Pattern Recognition*, **40**(1):262-282.

Saad, M. H. Saleh, H. I. Konbor, H. and Ashour, M. (2011), Image retrieval based on integration between YCbCr colour histogram and shape feature, *Proc. ICENCO 7th International Computer Engineering Conference*, Giza, 97-102, IEEE.

Hiremath, P. S. and Pujari, J. (2007): Content Based Image Retrieval based on Colour, Texture and Shape features. *Proc. 15th International Conference on Advance Computing and Communications*, Guwahati, Assam, 780-784, IEEE.

Takala, V. Ahoen, T. and Pietikainen, M. (2005): Block-Based Methods for Image Retrieval Using Local Binary Patterns. *Proc. of the 14th Scandinavian Conference on Image Analysis*, Joensuu, Finland, **3540**: 882-891, Springer Berlin Heidelberg.

Sivic, J. and Zisserman, A. (2003): Video Google: a text retrieval approach to object matching in videos. *Proc.*

9th IEEE International Conference on Computer Vision, Nice, France, **2**:1470-1477, IEEE.

Chen, Y. and Wang, J. (2002): A region-based fuzzy feature matching approach to content-based image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**(9):1252-1267.

Elharar, E. Stern, A. Hadar, O. and Javidi, B. (2007): A hybrid compression method for integral images using discrete wavelet transform and discrete cosine transform. *Journal of display technology*, **3**(3):321--325.

Gorman, J. Curran, J. R. (2006): Scaling Distributional Similarity to Large Corpora. *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia, 361-368, Association for Computational Linguistics, Stroudsburg, PA, USA.

Banda, J.M. Angryk, R.A. and Martens, P.C.H. (2013): On Dimensionality Reduction for Indexing and Retrieval of Large-Scale Solar Image Data, Journal of Solar Physics, **283**:113-141.

Magnus, S. (2005): An introduction to random indexing. *Proc. 7th International Conference on Terminology and Knowledge Engineering- Methods and Applications of Semantic Indexing Workshop*, TKE, 1-6, IEEE.

Geva, S. and De Vries. C. M. (2011): TOPSIG : Topology Preserving Document Signatures, *Proc. CIKM '11 20th ACM international conference on Information and knowledge management*, New York, NY, USA, 333-338, ACM Press.

Li, J. Wang and J. Z. Wiederhold, G. (2000): IRM: Integrated Region Matching for Image Retrieval. *Proc. ACM 8th international conference on Multimedia*, New York, NY, USA, 147-156, ACM Press.

Qiu, G. (2002): Indexing chromatic and achromatic patterns for content-based. *Journal of pattern recognition*, **35**:1675—1686.

Pass, G. Zabih, R. and Miller, J. Comparing Images Using Colour Coherence Vectors. (1996), *Proc. ACM fourth international conference on Multimedia*, New York, NY, USA, 65-73, ACM Press.

Rahmana, M.H. Pickering, M.R. Frater, M.R. (2011) "Scale and Rotation Invariant Gabor Features for Texture Retrieval," *Proc. DICTA International Conference on Digital Image Computing Techniques and Applications*, Noosa, QLD, 602-607, IEEE.

Manthalkar, R. Biswas, P. K. Chatterji, B.N. Rotation (2003) Scale invariant Texture Features Using Discrete Wavelet Packet Transform. *Journal of Pattern Recognition*, **24**(14):2455-2462.

Agarwal, S. Verma, AK. and Singh, P. (2013): Content Based Image Retrieval using Discrete Wavelet Transform and Edge Histogram Descriptor. *Proc. ISCON International Conference on Information Systems and Computer Networks*, Mashhad, 19-23, IEEE.

Minaqiang, Y. Kidiyo, K. and Joseph, R. (2008): A survey of shape feature extraction techniques. *Journal of Pattern Recognition*, 43-90.

Torralba, A. Fergus, R. and Freeman, W.T. (2008): 80 Million Tiny Images: A Large Data Set for Nonparametric Object and Scene Recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **30**(11):1958-1970.

Jianxiong, X. Hays, J. Ehinger. and K.A, Oliva. A, Torralba. (2010): A, SUN database: Large-scale scene recognition from abbey to zoo. *Proc. IEEE CVPR Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, 3485-3492, IEEE.

Chappell, T. Geva, S. Nguyen, A. and Zuccon, G. (2013): Efficient Top-k Retrieval with Signatures. *Proc. ADCS 18th Australasian Document Computing Symposium*, Brisbane, Australia, 10-17, ACM Press.

Hiremath, P. S. and Pujari, J. (2008): Content Based Image Retrieval Using Colour Boosted Salient Points and Shape Features of an Image. *Proc. International Journal of Image Processing*, **2**(1):10-17.

Banerjee, M. Kundu, M. and Maji, P. (2009): Content-based image retrieval using visually significant point features. *Journal of Fuzzy Sets and Systems*, 160(23):3323--3341.

Chowdhury, M. Das, S. and Kundu, M. (2012): Interactive content based image retrieval using Ripplet transform and fuzzy relevance feedback. *Proc. 1st Indo-Japan Conference*, Kolkata, India, 243-251, Springer.

Lazebnik, S., Schmid, C and Ponce, J. (2006): Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, **2**:2169-2178

Towards Social Media as a Data Source for Opportunistic Sensor Networking

James Meneghello[1] **Kevin Lee**[1] **Nik Thompson**[1]

[1] School of Engineering and Information Technology
Murdoch University,
Perth, Western Australia 6150,
Email: j.meneghello@murdoch.edu.au, k.lee@murdoch.edu.au, n.thompson@murdoch.edu.au

Abstract

The quality and diversity of available data sources has a large impact on the potential for sensor networks to support rich applications. The high cost and narrow focus of new sensor network deployments has led to a search for diverse, global data sources to support more varied sensor network applications. Social networks are culturally and geographically diverse, and consist of large amounts of rich data from users. This provides a unique opportunity for existing social networks to be leveraged as data sources. Using social media as a data source poses significant challenges. These include the large volume of available data, the associated difficulty in isolating relevant data sources and the lack of a universal data format for social networks. Integrating social and other data sources for use in sensor networking applications requires a cohesive framework, including data sourcing, collection, cleaning, integration, aggregation and querying techniques. While similar frameworks exist, they require long-term collection of all social media data for aggregation, requiring large infrastructure outlays. This paper presents a novel framework which is able to source social data, integrate it into a common format and perform querying operations without the high level of resource requirements of existing solutions. Framework components are fully extensible, allowing for the addition of new data sources as well as the extension of query functionality to support sensor networking applications. This framework provides a consistent, reliable querying interface to existing social media assets for use in sensor networking applications and experiments - without the cost or complexity of establishing new sensor network deployments.

Keywords: data sourcing, data mining, data integration, social media, social data, integration framework, sensor networking, resource preservation

1 Introduction

Sensor networks are a grid of spatially distributed multifunction sensor nodes designed to cooperatively monitor environmental conditions and pass collected data in a collaborative fashion for further analysis. These networks have provided numerous benefits to society since their development, and have previously been deployed in areas to monitor volcano activity, floodplains (Hughes et al. 2008), bushfires and earthquake risk zones (Akyildiz et al. 2002). Human-oriented sensor networks (such as Body Area Networks (Chen et al. 2011)) have also been developed with the aim of providing benefit to individuals through better monitoring of health conditions, as well as epidemic detection for viruses such as influenza (Okada et al. 2009). These systems rely on analysis of data collected by the sensors in order to provide actionable advice, and the majority of these systems have application-specific implementations. There is some repurposing of collected data, but sensor networks for generic use are generally not deployed as acquiring funding without a well-defined scope is difficult.

Sensor networking applications are constrained by available data sources. Supporting new applications often requires deployments of sensor hardware that are able to collect new types of data. This data can then be used in new analytical applications or used to extend and enhance existing applications. Using existing data sources to enhance sensor networking is an attractive prospect, as it does not require the expense of developing, implementing and maintaining a new sensor deployment.

To support the extension of sensor networks in this manner, existing networks capable of providing data must be located. A candidate network should be semantically and structurally similar to sensor networks to allow a smooth application of existing sensor networking techniques to the new data sources, and also support integration with existing sensor networks. It is likely that multiple data sources properly integrated into a cohesive dataset of a standardised format could meet these conditions, as long as the data was procured in a way similar to sensor networks: events are collected from the environment and communicated to a central host.

Social media is an ideal candidate for use as a data source in sensor networking. It is comprised of a number of isolated networks covering much of the global population, providing penetrative reach into diverse communities. It produces very large amounts of variable-quality data with accompanying metadata. When properly filtered and cleaned, it can be a source of good-quality, relevant data for integration into sensor networking applications. Social media also follows similar design patterns to sensor networking, being primarily event-based, and can conceptually be treated much the same way as a traditional sensor network. Social users (nodes) write comments (events) that are transmitted to followers (connections). Additional sensor nodes can

be emulated in a sensor network by injecting social data into the network with appropriate metadata.

The ubiquity of social media within modern society has led to strong userbase growth across all platforms, with 56% of Americans across all age groups now using at least one social network and 96% of those aged between 18-35 (FormulaPR 2011, Edison 2012). Globally, 26% of the population use social media, including 57% of Australians and 46% of Chinese (Singapore 2014). Social media users use tend to use different platforms depending on their nationality - Cyworld is estimated to host 50% of the social networking profiles of South Korean users, while Mixi is more popular in Japan and 20% of Orkut's population is Indian (Vasalou et al. 2010). Integrating these social networks would provide a truly global, culturally-diverse, rapidly-updated data source for sensor networking.

There is great demand for this social data within the academic and industrial communities. Social studies rely heavily on qualitative results from user-generated content such as surveys, while advertising entities harvest user metadata to more appropriately target advertising. Currently, data for these activities is collected directly from users by either asking for their participation in studies or by using web-based tracking technology to monitor their use of the internet. These two forms of data collection have been used in social sensor networks, as participatory (Burke et al. 2006) and opportunistic (Lane et al. 2008) sensing respectively. Problematically, both of these approaches require ongoing software deployments and usage of resources to monitor the environment - potentially dissuading users from continuing participation.

Using social media as a data source for sensor networking presents a number of challenges, from realistic and technical perspectives. Allowing the mass collection of data from users in an easily-queryable format could have unintended consequences, such as the mass-reporting of user locations (Borsboom et al. 2010). Analysis is made more resource-intensive by the high throughput of data posted to social media - Facebook alone collects and warehouses almost half a petabyte of data per day (Ching et al. 2012). Platforms to aggregate social media already exist, such as Datasift (DataSift 2010) and Gnip (Gnip 2008), and provide this service by performing widespread collection of all social media data, requiring substantial storage and processing infrastructure. These platforms are not developed with sensor networking integration in mind, and the event-based behaviour of social nodes is lost during the integration process.

In this paper, we propose a framework which performs data mining, filtering, integration and querying of generic social networks. This framework is designed to integrate with sensor networking systems and analytical tools to allow researchers and industry to leverage user-driven content for the purposes of event-detection, metadata analysis and general user analysis. This is achieved through conflating the concepts of social networks and sensor networks and provides a robust data integration chain with a data querying engine designed to streamline access to desired data. In addition, tools for mining data from social media are detailed and integrated into the framework to provide a more complete picture of each user and their content than merely parsing a single Application Programming Interface (API). The framework provides a simplified query interface to a cohesive data-set made from social media networks, providing researchers and industry with access to a large, global set of user-generated metadata and content.

Section 2 presents a review of literature and previous work completed in the areas of social data mining and opportunistic sensing. Section 3 analyses the use of social media as a data source for sensor networking, as well as the challenges involved in doing so. Section 4 describes a framework designed to alleviate some of these challenges by optimising the data collection process. Section 5 evaluates the data sourcing algorithm ability to discover potential data sources relevant to a given topic. Finally, Section 6 presents some conclusions.

2 Background

The process of collecting data from disparate social networks and integrating it for common use utilises techniques from a broad spectrum of computer science research. Supplying user data as a source for sensor networking is commonly referred to as participatory or opportunistic sensing, where participatory sensing refers to the direct contribution of data from users and opportunistic sensing passively observes users. This section examines the use of both of these methods of data collection from social networks, as well as examining previous applications of social data in analytical studies. Finally, data integration techniques are examined for use in linking collected data across social networks.

2.1 Participatory and Opportunistic Sensing

Sensor networks are typically a spatially-distributed mesh of potentially thousands of sensors, with sensor nodes deployed in a configuration befitting the desired application. Each node cooperates to move collected data back to the sink node - using other nodes as a relay to increase communications range. The sink node then typically has a connection back to more powerful resources for processing the information, though this may also be collected manually. Low-power wireless sensor networks are often used to collect data from the natural environment, the human body, mechanical equipment and many other sources. This data is, in turn, analysed and logged or used to actionable effect - for example, to trigger a warning in the event of an earthquake.

Participatory sensing uses existing mobile devices and users as nodes within a sensor network (Burke et al. 2006) by encouraging users to gather, analyse and share local knowledge in what is commonly referred to as "crowdsourcing" (Kanhere 2011). By treating users as sensor nodes, participatory sensing takes advantage of resources that already exist to extend the reach of sensor networks for a number of purposes, including urban planning and policy development. Smartphones also come with a number of sensors that can be made available for participatory sensor networks, including the important contextual metadata sensors of location and time.

Participatory sensing requires direct and active participation of users, which comes with a number of challenges. One of the key issues with using people as nodes in sensor networks is that they are often less reliable than hardware sensors (Hughes et al. 2014). Users are able to choose whether to provide data on a requested topic, and may choose not to do so. Participatory sensing is named as such for a

reason - without active participation, the system is unable to produce useful outcomes.

Further research in using participatory sensing for urban sensor networks has resulted in the development of several auxiliary approaches. Opportunistic sensing reduces the burden on the user by lessening direct participation - instead relying primarily on the devices the user carries around (Lane et al. 2008). Applications on smartphones can take sensor measurements without bothering the user and pass it along to sink nodes over mobile networks. While quality data is still dependent on the user's presence in an area of interest (Min et al. 2013), the user is no longer required to directly answer queries. Data produced by hardware sensors through opportunistic sensing is objective and is in a standard format (Campbell et al. 2006). Data is potentially of good quality, as it tends not to suffer from errors introduced by human input. Under this approach, collected data is only provided by sensors directly attached to the device, and users cannot be queried for additional information. This use of smartphones in opportunistic sensing is commonly referred to as Mobile CrowdSensing (Ganti et al. 2011).

In order to leverage opportunities provided by both participatory and opportunistic sensing, some architectures combine both techniques (Guo et al. 2014). This allows users to provide data as requested and fill contextual data using opportunistic sensing. Concerns with data quality arising from direct user involvement remain.

A major problem with many implementations of both participatory and opportunistic sensor systems is the requirement of manual application installation by the user. To retrieve data from a smartphone's sensors, an application must be installed that allows the smartphone to operate as a sensor node. These applications are usually neither large nor difficult to install, but even the smallest of hurdles can hamper efforts to leverage users as sensors. These services also consume energy on devices that have limited resources, which may drive some users away. To encourage users to contribute data in an ongoing manner, some kind of compensation is usually required.

To alleviate these issues, sensor middleware suites have been developed (Hachem et al. 2013, Hughes et al. 2009) for smartphones that only require a single installation, even if there are multiple different deployments using the device. These frameworks vastly simplify sensor installation and reconfiguration, further reducing burden on the user.

The framework proposed in this paper aims to use indirect participatory and opportunistic sensing at a higher level by analysing social media. While facing similar data quality challenges to both techniques, we work around user participation by only monitoring existing social media usage. No additional applications need to be installed and no extra resources are consumed by personal devices.

2.2 Social Media

Effectively deriving useful data points from social media is a topic of much discussion, and presents a number of challenges (Maynard et al. 2012). Posts by users on social media are usually free-form - the data comes in no particular standard format, as they often represent part of a stream of consciousness from a user. Additionally, posts are not limited to text and users often share pictures, videos, diagrams or graphs which present unique challenges to data analysis.

The proliferation of social media has driven content and context creation, but has also made it more difficult to locate appropriate data sources (Wandhöfer et al. 2012). Where topics were once discussed in semi-centralised locations such as Usenet, the integration of commenting systems into many different news and blog sites has dispersed information to this point where it is unrealistic to attempt collection of all relevant discussion relating to a topic. Instead, discussion centers can be discovered by tracking the spread of topics across the internet and monitoring the most active communities.

Many existing studies on using social media for event detection tend to focus on a single social network (Li & Cardie 2013, Cameron et al. 2012, Sakaki et al. 2010, Robinson et al. 2013), limiting collection specifically to the demographics represented on the chosen platform. In order to develop a truly global and cross-cultural social media monitoring system, data collection mechanisms should be extensible to any form of online social media, rather than a specific few platforms.

To realise the concept of global generic social sensor networking, a diverse set of data collection and integration techniques need to be utilised. Different social networking platforms operate on different data structures, using different storage engines and producing entirely different output. Even considering this, social media data is conceptually similar across platforms, consisting of a number of common constructs (users, friends, connections, messages, events). Because of this conceptual similarity, it is possible to integrate this data into a single queryable dataset through the use of data collection and integration techniques.

2.3 Generic Collection and Integration

Generically utilising data from different web services requires integration between social media platforms. While programmatic access to web services has vastly improved since the introduction of Web 2.0 paradigms, interoperability and data integration between social networks remains an issue. There is limited interoperability of services provided for the purposes of open authentication, but content sharing is usually limited. There have been some attempts to apply semantic web principles to the problem, including Semantically-Interlinked Online Communities (SIOC) (Breslin et al. 2009) which describes social networks using the Resource Description Framework (RDF) to improve interoperability. While RDF has yet to reach widespread adoption and is unavailable for use with many social media systems, the data structures and principles in use provide a good platform upon which to base further work.

There are a number of challenges surrounding the matching of social media profiles between networks. The FOAF (Friend-of-a-Friend) Project (Brickley & Miller 2000) attempts to extend Semantic Web efforts to social media by providing a base for user profile matching across networks. It does this by combining names and user metadata (such as location, email addresses or education details) to provide a more substantial set of data to improve accuracy of matches. As with the SIOC project, few social networks actively support such efforts and most do not provide FOAF output for users.

The majority of studies conducted using data collected from social media follow a reasonably similar process: conduct (manual or automated) searches of a social network, save or export returned data in a simple format, and perform analysis (online or offline) to determine answers to a particular query. While this process is reasonably generic, there have been no attempts to develop a querying engine for social media analysis. Query engines have been developed for a range of other purposes (Khoury et al. 2010, Madden et al. 2005) with the intention of providing a standard querying interface and abstraction layer on top of complex datasets. The development of a query engine that can genericise collection and analysis over multiple social media platforms without requiring large infrastructure outlays would be a useful addition to social media research.

2.4 Integration of Social Data

Data integration can be a complex process, depending on the complexity, relevancy and size of the converging datasets. Numerous techniques exist to handle the process of querying integrated datasets, designed with different operating requirements. Some techniques require the offline transformation and storage of data for later querying, while others can handle-queries in real-time by inferring the goal of the query and transforming appropriate data as required.

Extract-Transform-Load (ETL) systems are commonly used in data warehousing, where data is initially cleaned and transformed for storage and later use (Vassiliadis 2009). Due to the extensive processing that data undergoes in order to be in a clean state with unified schema, this process is generally not real-time, and may result in data freshness issues. This approach is therefore only useful for use in delayed queries, and its use with social media would exclude any real-time sensing or systems that require a timely response.

There are also processing systems such as Google Cloud DataFlow (Perry 2014) that would be appropriate for the task of integrating large social datasets. BigQuery (Sato 2012) can be used to support a platform providing a queryable interface to real-time integration of social streams that can be use in decision support systems, providing actionable insights.

Some attempts have previously been made to support interoperability between social networks, particularly the Semantically-Interlinked Online Communities (SIOC) project (Breslin et al. 2009). The SIOC project uses the Resource Description Framework specification to integrate some elements of social networks, particularly the association of user profiles over different networks. SIOC exporters have been developed for a limited number of social networks.

Commercial services such as Datasift (DataSift 2010) and Gnip (Gnip 2008) perform integration and long-term storage of social data. The integrated data is then presented to applications through a query API, which is often used by companies to monitor their online social profiles. This allows for rapid response to user complaints directed at social followers (rather than the company itself, through a complaints procedure). The integration process of these systems does not consider the event-based behaviour of social nodes, and are generally unsuitable for use in extending sensor networking. Ongoing access costs can also be a constraining factor for applications intending to use social data, and independently deploying such a system requires a level of processing and storage infrastructure that would be considered infeasible for most projects.

3 Social Media as a Data Source

Social media is comprised of a number of isolated networks covering much of the global population. They are primarily used for facilitating communications between people over the internet, social networks have also been used to gauge user response to advertising (Taylor et al. 2011) and also as transmission mediums for sensor networks. Potential sensor networking applications rely on available data sources, and the penetrative reach of social media into global communities and vast amount of data posted provides an ideal data source for this purpose. Properly prepared, social data is suited for further analysis, particularly for detecting cross-cultural events or insights. Using social media as a data source for sensor networking is not a trivial task. There are significant challenges facing any platform aiming to facilitate this dataflow.

To use social media in this way requires the integration of disparate social networking platforms. There have been attempts to ease the bidirectional migration of data on social networking platforms, as noted in Section 2.3, but these have generally had little industry support. Most networks have an interest in "locking-in" customers, to dissuade them from migrating between networks and losing associated advertising revenue. Easing data integration processes would also accelerate migration between competing networks, so efforts to standardise export formats are unlikely to ever receive industry co-operation. Therefore, new techniques supporting inter-network social data integration need to be developed in order to facilitate the integration of social media into sensor networking.

To enable the use of social media as a data source for sensor networking, a framework supporting this goal must be developed. This framework is made from a number of processes designed to take heterogeneous data sources, integrate them into a common schema, provide generic querying functionality and present appropriate output for further analysis or integration into sensor networks.

This framework must address a number of key challenges:

1. Collecting data from all available sources for use in sensor networks can be infeasible due to the sheer amount of data being produced. To query social media data in an efficient manner while retaining the ability to query as much relevant data as possible, some way of reducing the incoming flow of data to exclude irrelevant sources is required.

2. To use social media as a sensor, sourced data needs to be retrieved. Retrieving social media data can be straightforward if an API is provided, or require manual scraping if key information is not provided by the API.

3. Social media data is noisy because it is almost entirely user-generated and doesn't adhere to a standard structure or format. This data requires extensive cleaning to remove spam and bring low-quality content up to a usable standard (Agichtein et al. 2008). Without

performing this cleaning, using the data in automated applications becomes significantly more difficult.

4. In order to present the collected data in a format that can be integrated with sensor networking applications, there must be a way comparing diverse datasets. Without a method of mediating schematic differences between data sources, every application would require manual mapping of data structures.

5. The execution of queries over datasets as large as those that social media provide can be challenging, due to resource constraints and missing values. Sensor networks also deal with data in many different ways and can require extensive pre-processing and aggregation to be performed prior to use.

6. Some sensor networking applications can integrate data in simple formats such as JSON, but others require more complex techniques (such as event injection) to emulate social nodes as sensor nodes.

In order to properly define steps in the social data integration process, each of these challenges is represented by a key area of functionality, respectively: Sourcing, Collection, Cleaning, Integration, Querying and Presentation. Once a solution for each of these challenges has been identified, they can be joined into a unified process supporting the integration of social media and sensor networking. These challenges are described more fully in the sections below.

3.1 Sourcing

One of the most simple optimisation steps in large-scale data processing applications is to filter incoming data, resulting in reduced processing for each successive step in the integration process. As social media has an extremely broad scope, relevant data sources must be located in order to efficiently leverage social data in sensor networking applications. Without this initial filtering process, much of the social data processed may be completely irrelevant to the application. As this data must undergo cleaning, integration, storage and querying, early filtering can result in significant resource savings. Hence, the process of finding quality data sources is very important.

Sourcing involves locating social data streams that provide data relevant to the intended application. On the internet, many of these sources can provide access to historical or real-time data streams. Both of these types of data can be useful to extend or enhance sensor networking applications. Real-time data can actively replace or enhance sensor nodes with additional data sources, while historical data can provide longitudinal context. Data sources can also be normal rich-text web pages that can require substantial parsing and cleaning before use.

Most of these data sources also provide access to further sources. Social media posts often contain hyperlinks to static web content, and also contain metatags (such as usernames and hashtags). Static content is often interlinked, with most websites providing hyperlinks to other relevant material on associated sites.

Figure 1 presents the process by which sourcing occurs. The query specifies relevant keywords, which can be expanded upon by use of predefined databases and appended to by examining oft-used keywords on strongly-relevant search results. These keywords are used to query known search APIs, returning a list of locations that potentially contain results. The exact method used for data access and searching can vary for each system, as each platform provides differing methods of accessing and filtering data. Some examples of different methods are:

Facebook Using the API, search for public posts with related search terms, popular news feeds and other items of interest. Additionally collect comment authors.

Twitter Using the API, search for public tweets containing related keywords. Additionally collect information about replies to tweets, and also examine hashtags commonly appearing within the initial set.

Blogs Using Google's Search API, search for public blog posts containing relevant keywords, including author information and comments. Additionally collect relevant results from commenters' own blogs.

Forums Using manual page scraping and authentication for forum software such as VBulletin and phpBB, search for forums containing threads relevant to our query.

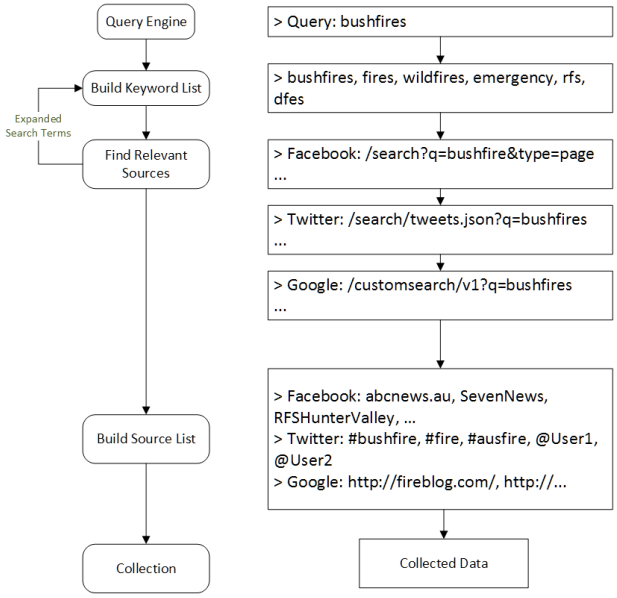


Figure 1: Locating relevant social data sources

Upon completion of this initial phase, a list of relevant places to look for information relating to this single query has been collected. Not all data present within these sources is useful, and some may be completely irrelevant. By selecting a fairly wide sub-set of available data, we have already limited the initial collection to a feasible scope, allowing for more accurate filtering later. This optimisation can potentially exclude obscure results, but this is a necessary trade-off.

3.2 Collection

To use data from identified sources in applications, data must be collected. There are three main approaches to data collection: the use of Application

Programming Interfaces (APIs), access to formatted real-time streams, and ad-hoc data collection.

Using APIs to collect data involves sending a specifically-constructed request to a provided server. This request contains a number of parameters used to focus the scope and range of data returned by the API. The requested data is then returned to the requester in a well-defined format. Some APIs provide a limited view of data available to the source, requiring the requester to follow-up with an ad-hoc request for more data. These initial API requests often lead to further queries for related data (such as collecting user profile data for users that have posted in a comment thread).

Data can be collected from real-time streams by requesting stream access from a source server. The source server then pushes a constant stream of real-time data towards the requester, in what is often called a "firehose" stream. This data is presented in a well-defined format that can be mapped to an appropriate data schema. Many firehose streams consist of all available real-time data being produced by the source. The amount of data provided by this real-time stream can be problematic for systems operating with restricted bandwidth, processing or storage resources, and can easily overwhelm low-resource systems.

Ad-hoc parsing or scraping of data can be used for data sources that have no defined format and do not provide an API or formatted data stream. This usually requires the manual development of parsing algorithms tailored to specific sources. Automated parsing algorithms can also be used, to varying levels of success. Ad-hoc scraping can also be used to enhance data provided by APIs, where data is missing or deliberately restricted from API access.

Figure 2 illustrates an example structure for handling data collection across multiple source types and authentication methods. For many sources, collection can occur through accessing a network-provided API, such as Facebook or Twitter's APIs. Many APIs operate on similar authentication standards (such as OAuth (Hardt 2012), depicted), requiring minimal work to write wrappers for a generic scraping engine even across a wide variety of different sources. Other sources require collection through page-scraping, a more resource-intensive process that involves writing parsers for source pages and handling page authentication in a customised manner. While API access is generally less resource-intensive and requires less development work than scraping, scraping can potentially provide more data.

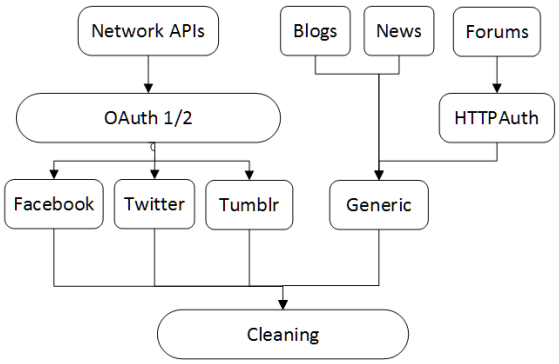


Figure 2: Collection module flow

Data collection is limited by the interfaces provided by the API, e.g. the Facebook API does not provide location information for users, even if the data is set to be publically viewable. A second-pass scraper can access the profile page directly to collect this information, providing more complete meta-data but is also more resource-intensive. Second-pass collection can also involve further queries to the API for related data (such as collecting user profile data for users that have posted in a comment thread).

The collection process can be very easily distributed across resources, as each first-pass operation is isolated. The initial source list can be packaged by a workflow manager and work assigned in an optimised manner to ensure that requests are spread evenly over workers. When workers have run out of allocated work, second-pass collection can occur in a manner that ensures atomicity.

3.3 Cleaning

Data collected from social media and other internet sources is noisy, and can require extensive cleaning and processing. Most social data is user generated and adheres to no particular structure, primarily being unstructured conversational language. Differences in data formats between sources can also be problematic, such as the use of different date standards between cultures and systems. Conceptual differences between individual data elements across different networks can also require manipulation of data into a unified standard.

There can also be structural problems with social data depending on its age and nature. Legacy data has often undergone repeated format and schema shifts, so data from these sources can require extensive cleaning. Modern sources adhere to stricter standards and usually require less cleaning.

The method of collection can determine the extent of data cleaning necessary. API-provided results are usually retrieved from the database and output without presentation, and therefore adhere to internal database quality standards. Data returned from ad-hoc scrapers can require extensive cleaning, including removal of undesirable elements such as HTML tags and extraneous characters.

It is at this point that cleaning data for privacy reasons may be handled. Anonymisation of users can be performed during the cleaning process to ensure user privacy for applications that do not require identifying information to be stored.

3.4 Integration

Data from different social sources are collected using their original schemas. Attempting to query data across these many schemas can be difficult without proper data integration, as it requires the query engine to specifically deal with many different data formats and sources. Data integration provides the ability to query data over a mediated schema, in which all sources can be treated in a generic manner. To use this data set in sensor networking applications, collected data must be integrated.

There are a number of available data integration techniques supporting this goal, including those discussed in Section 2.4. The integration process can be operated by predefined mappings from collected data schema to the global schema, such as the RDF specifications used in the SIOC project (Breslin et al. 2009). It is possible to use automated mapping algorithms (Doan et al. 2001) to develop page and API wrappers that require minimal modification by a developer, reducing development time.

These mappings can be applied to data in a number of different ways, depending on the database engine used. Data mappings to relational designs can store the integrated data in relational database management systems, while there are also options for NoSQL, key:value and tuple stores. Each provides the ability to perform a different form of integration, so any integration design process should also take the choice of query engine into account.

3.5 Querying

To provide only relevant and desirable data to sensor networking applications, there must be a method of restricting, aggregating and analysing integrated data. Analysis performed may require results aggregated by categorical variables, and this can be handled using querying. Querying is an important part of data analysis, and is readily available in all database systems.

As the integrated data is loaded into a relational database management system with a unified schema, query processing and optimisation is greatly simplified. Additional operators and aggregate functions can be added to the query syntax to allow for the sourcing and collection mechanisms detailed in 3.1 and 3.2 respectively, including the ability to filter by Site and restrict potential data sources by keyword. Queries can be provided in both SQL-like syntax or to a RESTful API, using JSON or XML.

The utility of integrating social data into sensor networking applications ultimately hinges on the ability to adequately query the data. Other systems designed to integrate diverse data sources into applications place significant emphasis on their querying engines. Software projects such as Google's BigQuery and Facebook's Presto were designed specifically to query large-scale sets of data in real-time. Querying social data therefore needs to be efficient in order to provide real-time results to sensor networking applications.

3.6 Presentation

Effective data presentation is required in order to use integrated data in sensor networking applications. Applications often handle different input data formats, such as XML, JSON or event packets. Presenting the data in this format for integration should be handled in such a way as to be compatible with any application.

Presentation can be delegated to client applications, with the querying engine only providing result data in a variety of text-based formats such as JSON and XML. This data can then be provided for import into other systems or directly analysed for use in graphical applications. For use with automated systems, queries can be designed to provide simplified output to directly trigger actions or provide detailed output for further use in Decision Support Systems.

Direct integration is more involved but allows for social nodes to be used directly in sensor networking applications. This works by developing output formatters that exist within those networks and rebroadcast data in the format required, often event packets. This allows social data nodes (users) to directly act as nodes within the sensor network.

4 A Framework for Supporting Social Media as a Data Source

In order to support the integration of generic social media data into sensor networking applications, the functional components described in Section 3 are implemented in a streamlined data processing framework. This framework encapsulates all necessary operations from initial data request to sourcing, collection, cleaning, integration, querying and presentation of returned results. For ease of presentation, the framework architecture is presented in multiple views.

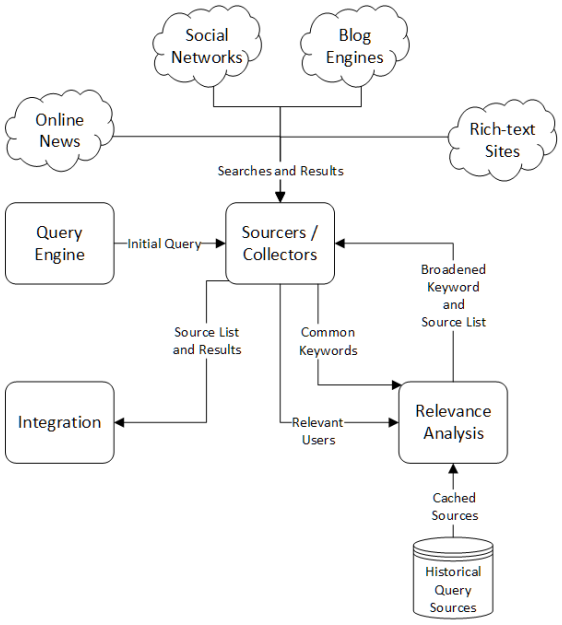


Figure 3: The process of finding new data sources

Figure 3 presents the workflow for sourcing and collection. In this stage, the framework takes the initial query and expands upon the specified keywords in an iterative process. Potential sources are taken from historical data and queried for relevance, and commonly-recurring users and keywords are crawled to discover additional relevant sources. A wide range of sources are discovered, analysed and discarded during this process to ensure adequate data coverage of relevant sources.

Sourcing is a two-step process. Using an initial seed set of keywords, the Sourcers locate sources and use content from these sources to expand and reinforce the keyword lists, as well as isolating relevant metatags (such as URLs or usernames). In the second step, the metatags are put back through the Sourcers to locate additional sources and further reinforce keyword and source lists. Each additional iteration of this process serves to further strengthen the keyword and source list, effectively using machine learning to focus the search space.

Sourcing can be performed in two modes. The first is a narrow search that attempts to find the most relevant sources while being hesitant to expand the search space. The second is a much broader search that uses all available content to train the keyword set and broaden the search space, finding potentially-relevant sources over a much larger area. Narrow searches are designed to find the most relevant sources quickly, while the broad search can find distantly relevant sources but additional resource requirements.

For organisational reasons, the sourcing components are integrated with the collection components. Functionality is shared between these two processes, with some content collection required to assess relevance of sources (and complete collection required during the collection phase).

The sourcing and collection software processes can be distributed across multiple resources, only requiring synchronisation to finalise the completed source list and return the completed collection results. Individual site wrappers may also be distributed, which can be particularly desirable in the instance of micromanaging API throttle limits that tend to differ between platforms. Distribution of these processes provides significant performance boosts in the sourcing, collection and filtering phases.

Relevant sources are cached for two purposes. The first is to expedite the sourcing process for repeated similar queries and reduce overhead and external API usage, which associates sources with particular keyword sets. The second is to identify rich-text websites that contain a large amount of useful data or pages. By keeping track of which sites are providing useful data sets, candidates for further wrapper development can be identified.

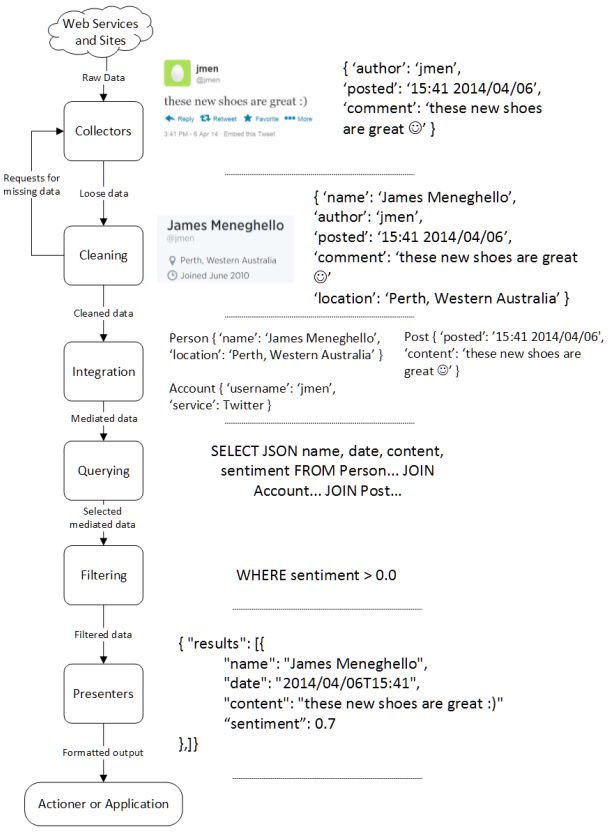


Figure 4: Example data flow within framework

The framework wraps around a Relational DataBase Management System (RDBMS), as shown in Figure 4. This wrapping occurs in multiple directions by allowing modified incoming queries, automating data collection, extending available aggregate and filtering functions and also providing for formatted output. For simplification, Figure 4 does not show processes relating to sourcing and first-pass filtering, which are instead shown in Figure 3.

After determining a source list, the framework scrapes information from sources using collectors. The collectors are either source-specific wrappers or generic web scrapers that can collect varying amounts of information from sources. Source-specific wrappers (e.g. Facebook or Twitter) return data in a standard format that can be easily integrated, while the generic scraper returns data that may be low-quality or require cleaning, as discussed in Section 3.3. This cleaning process can require further related data to be requested from sources.

The integration component takes cleaned data and integrates it over the mediated schema, as shown in Figure 5. Once complete, the data is fit to be inserted into an RDBMS using a pre-defined schema. This schema was developed using both the Semantically-Interlinked Online Communities (Breslin et al. 2009) and Friend-of-a-Friend (Brickley & Miller 2000) projects as inspiration, allowing for simple output to common semantic web formats while still retaining a high standard of performance over large datasets. The schema has been modified to work within an RDBMS, which retaining as much flexibility as possible.

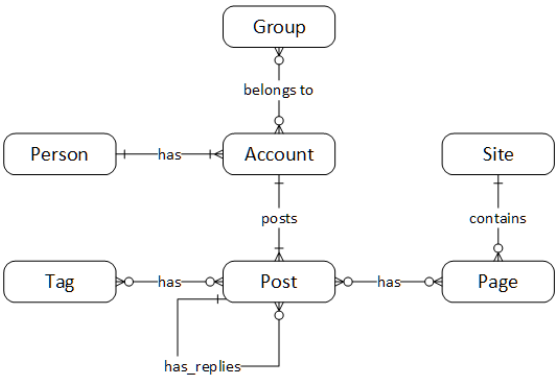


Figure 5: Internal mediated data model

Figure 5 presents the internal data model used to store integrated web data. Designed for use with relational database management systems, the model supports enough expression to adequately cover most social network and social page examples. Not all aspects of the model are utilised by every network, and some networks use certain models differently: while Twitter and Facebook can use the Tag model to represent hashtags and other metadata, blogs can use the same model to represent word tags as part of the word clouds used in most blogs.

Some model entities in Figure 5 see significant reuse due to their generic nature. Blog posts, comments, tweets, statuses and forum posts are all conceptually similar, generally consisting of a limited attribute set: title, date (posted/created), author (and other user details), content and a set of replies, along with other assorted metadata. A simple Post entity (related to Accounts to provide author and commenter data) can satisfy all of these requirements, including any comments or replies by using an adjacency relationship.

Queries submitted to the framework are translated before being passed through to the RDBMS, as there are extensions made in order to support sourcing and presentation and the original queries are not SQL. Custom second-pass filters and aggregators can be called directly, assuming the RDBMS supports user-created functions. Using the filters in this way also allows the system to cache and optimise queries natively where possible,

without requiring extensions to the database engine.

Once the query has been completed, the data can be formatted by presenters. These presenters can be in the form of a RESTful Web API that outputs JSON, XML or graphed images, or can alternatively be integrated into sensor networking systems through use of custom event generators. Normally, presentation is handled at an application-level and is inappropriate for inclusion, but the platform effectively wraps the RDBMS and provides this functionality to enable simple access to the data. Accessing RDBMSs without an interface layer raises the barrier to entry for users, requiring direct console access and usually provides data in awkwardly-formatted text.

5 Evaluation

One of the prominent and novel advantages of the proposed framework over existing approaches is the use of on-demand social data sourcing. Effective data analytics relies on the presence of quality data sources, so this is an important goal. The evaluation provided therefore emphasises data sourcing, and examines the relevance and quality of data sources discovered.

5.1 Experimental Setup

A proof of concept experiment for the sourcing algorithm was implemented in Python 3.4, taking advantage of the Natural Language ToolKit (Bird et al. 2009) library for text mining. The sourcing algorithm was executed on an Amazon EC2 m3.medium instance, and individual experiments were conducted in a single run to ensure that the available social data did not dramatically change between runs.

Sourcers were implemented for Twitter and Google Custom Search, as well as a generic web scraper used to further build keyword lists from sources. Both services are subject to API throttling limits, and these limits are taken into account during the sourcing process.

The process for each experiment consists of a series of iterations. A single iteration consists of the following steps (subject to configuration values):

1. Search Sourcers using initial seed keyword
2. Generate target source list
3. Retrieve content from list of sources
4. Mine content for commonly-occurring keywords and metatags (URLs, Usernames, Emails)
5. Further add to source and keyword lists
6. If [broad_search]: Search Sourcers using metatag lists (ie. Twitter users) and add to sources
7. Iterate using expanded keyword set

There are a number of possible configuration options for each experiment, which are explained below:

broad_search Whether the sourcing algorithm should also take sources from collected content

max_search_results The number of results to return from a Sourcer search (important to avoid API throttling)

The broad search algorithm also collects special metatags from content, such as usernames and hashtags on Twitter, and email addresses or URLs in static content. These metatags are then used to broaden the search scope, discovering additional search vectors and providing a significantly higher number of data sources while being subject to much higher noise levels. These searches are examined in the next section.

5.2 Relevance

The relevance of discovered data sources is an important metric in evaluating the usefulness of this sourcing algorithm. In order to evaluate this, the algorithm was given a broad seed keyword ("Australian politics") and let to run over multiple iterations, with an upper limit on the number of search results returned from each API of 250 for Twitter and 50 for Google. The source list was exported to a comma-separated value file and each source was manually evaluated for relevance to the topic. The algorithm was then executed twice: once as a normal search, and once with the additional broad searching options enabled.

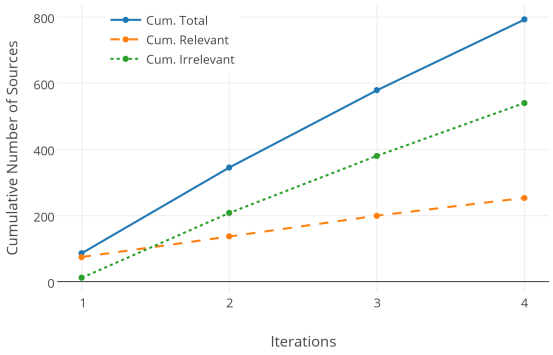
We define sources as relevant based on a number of factors: whether the source provides data related to the queried topic, the ability of the source to provide ongoing data and other sources, and the quality of data provided and its ability to be used in queries. If a source satisfies these criteria, it is considered relevant. All other sources, including those recorded as a result of algorithmic mishaps (such as misparsed hyperlinks and advertising servers), are deemed as irrelevant. Relevance is a therefore a boolean result.

As to be expected, the most relevant sources are quickly and easily found by a normal search with a low percentage of irrelevant results in the first sourcing iteration, as seen in Figure 6a. The broad search finds a higher number of relevant sources, but with a significantly higher number of irrelevant sources, shown in Figure 6b. In both instances, these sources are expanded and new keywords are derived, and additional relevant sources continue to be found at a lessening rate.

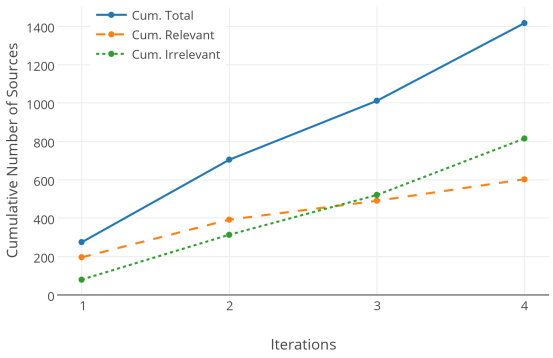
After the first two iterations, new sources continue to be found at a relatively linear rate. The relevance of discovered sources decline steadily relative to the total after the second iteration, but still maintain an acceptable rate of discovery. Figure 7b shows a steep increase in irrelevant sources during the fourth iteration, as the search space expands out beyond any semblance of relevance. The broad search maintains a much more even percentage of relevant results over the search space.

5.3 Keywords

The search keywords (initially seeded as "Australian politics") are expanded based on discovered content and the most relevant keywords float to the top of the list. The first iteration of both searches immediately expand the keyword set to contain mostly relevant keywords, as seen in Table 8a. Successive iterations narrow the search space down to a specific set of topics that quite accurately frame Australian politics. Interestingly, the broad search (which relies more heavily on page content rather than Twitter) contains a higher number of historical keywords, seen in Table 8b. A number of these keywords from a broad search relate to the government as of 2013, whereas those from the

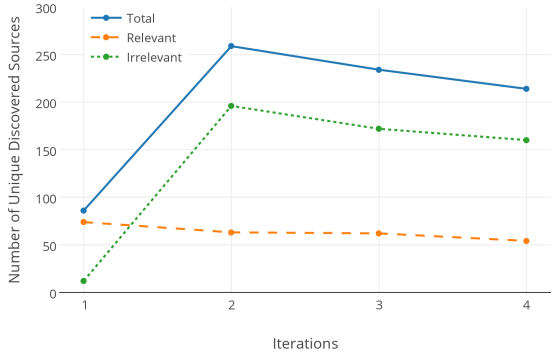


(a) Normal search

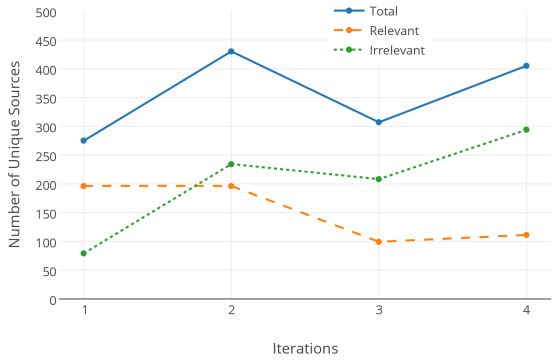


(b) Broad search

Figure 6: Cumulative relevance of data sources



(a) Normal search



(b) Broad search

Figure 7: Relevance of data sources

normal search in Table 8a relate more to the current state of Australian politics, circa 2014.

The difference in keyword sets between the search types also affects the relevancy of discovered sources after successive iterations. The iterative improvement of the search space during broad searchers potentially explains why even the third and fourth iterations of searching still contain a relatively high percentage of relevant results, as seen in Figure 7b. The normal search relies on a much smaller set of newer content from which to derive keywords, resulting in a lower discovery rate of historical data sources.

5.4 Signal-to-Noise

The signal-to-noise ratio of each search type was also examined, relative to the total number of sources discovered. A normal search discovers relevant data sources at a ratio of over 7:1 during the first iteration (Figure 9a), but immediately drops to a very low success rate in successive iterations. By comparison, the broader search starts with a much lower success rate of approximately 2.5:1 (Figure 9b), but maintains a steadier ratio well into later iterations, providing a more steady flow of new relevant sources. As explained in Section 5.3, this is likely due to the broader search touching on a larger source of historical data sources due to differences in keyword selection.

5.5 Analysis

The results of the sourcing algorithm indicate that the search methods (normal and broad) operate along different parameters. Normal searches rely more heavily on new data provided by sources such as Twitter, while the broad searches derive keywords primarily from older data sources such as newsfeeds and articles. As a result, the training of keyword sets tend toward two different trends: modern and historical, but could also indicate a disconnect in discussion between traditional media and social media. Both of these search types are useful, depending on the desired application. Overall, both search types provided a significant number of relevant data sources and ultimately achieved their goal - to optimise the collection of social media data.

There are a number of improvements that could be made to the sourcing algorithm. One would be to increase the use of training to include potential sources, preferring those new sources that had multiple existing links to discovered sources. A second involves a combination of both approaches, using historical keywords to search social media sources and modern keywords to search historical data sources.

6 Conclusion

This paper presents a framework that supports the sourcing, collection, cleaning, integration and presentation of social data for experiments and applications. The output provided by the framework can be used to drive generic sensor networking and other social analytic applications without requiring the significant hardware infrastructure investment used to store social data for later querying. This allows for the use of social media as a data source for generic sensor networking applications, without

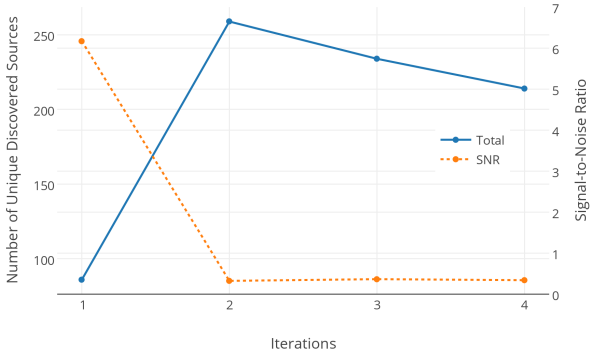
Iterations			
1	2	3	4
politics	abbott	abbott	abbott
australia	australia	government	tony
government	government	australia	australia
australian	news	tony	government
university	australian	news	news
party	minister	minister	minister
minister	party	party	party
abbott	politics	people	people
news	tony	pm	politics
media	people	politics	australian

(a) Normal search

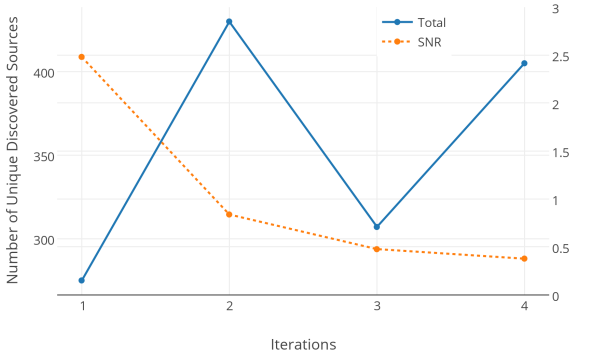
Iterations			
1	2	3	4
politics	australia	australia	abbott
australia	government	abbott	australia
australian	politics	government	rudd
government	party	rudd	labor
party	australian	party	government
news	abbott	minister	party
media	minister	labor	minister
world	labor	australian	news
minister	rudd	politics	australian
abbott	pm	news	election

(b) Broad search

Figure 8: Top 10 keywords used for searches



(a) Normal search



(b) Broad search

Figure 9: Signal-to-Noise Ratio of data sources

requiring new hardware deployments. The integration of disparate social and static media platforms also supports cross-cultural and cross-platform experimentation and analysis without requiring extensive user knowledge or experience.

This work also evaluates the use of a novel new data sourcing algorithm, designed to optimise the task of collecting relevant data for use in the framework. Two different approaches to sourcing are evaluated, with relevant data sources identified. A narrow search approach is found to discover relevant social media sources, while a broader search is more adept at discovering historical data sources. As the framework is designed to operate over both new and historical data, both approaches are useful.

As future work, improvements to the sourcing algorithm are recommended to extend its effectiveness to new and historical data sources simultaneously, providing the most relevant data sources possible for use in data analysis and applications.

References

Agichtein, E., Castillo, C., Donato, D., Gionis, A. & Mishne, G. (2008), Finding high-quality content in social media, *in* 'Proceedings of the 2008 International Conference on Web Search and Data Mining', ACM, pp. 183–194.

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y. & Cayirci, E. (2002), 'Wireless sensor networks: a survey', *Computer networks* **38**(4), 393–422.

Bird, S., Klein, E. & Loper, E. (2009), *Natural language processing with Python*, O'Reilly Media, Inc.

Borsboom, B., Amstel, B. v. & Groeneveld, F. (2010), 'Please rob me'.
URL: *http://pleaserobme.com/*

Breslin, J., Bojars, U., Passant, A., Fernandez, S. & Decker, S. (2009), 'Sioc: Content exchange and semantic interoperability between social networks'.

Brickley, D. & Miller, L. (2000), 'The friend of a friend (FOAF) project'.
URL: *http://www.foaf-project.org/*

Burke, J. A., Estrin, D., Hansen, M., Parker, A., Ramanathan, N., Reddy, S. & Srivastava, M. B. (2006), 'Participatory sensing', *Center for Embedded Network Sensing* .

Cameron, M. A., Power, R., Robinson, B. & Yin, J. (2012), Emergency situation awareness from twitter for crisis management, *in* 'Proceedings of the 21st international conference companion on World Wide Web', ACM, pp. 695–698.

Campbell, A. T., Eisenman, S. B., Lane, N. D., Miluzzo, E. & Peterson, R. A. (2006), People-centric urban sensing, *in* 'Proceedings of the 2Nd Annual International Workshop on Wireless Internet', WICON '06, ACM, New York, NY, USA.

Chen, M., Gonzalez, S., Vasilakos, A., Cao, H. & Leung, V. C. (2011), 'Body area networks: A survey', *Mobile Networks and Applications* **16**(2), 171–193.

Ching, A., Murthy, R., Molkov, D., Vadali, R. & Yang, P. (2012), 'Under the hood: Scheduling MapReduce jobs more efficiently with corona'.

URL: *https://www.facebook.com/notes/facebook-engineering/under-the-hood-scheduling-mapreduce-jobs-more-efficiently-with-corona/10151142560538920*

DataSift (2010), 'DataSift'.
URL: *http://datasift.com/*

Doan, A., Domingos, P. & Halevy, A. Y. (2001), Reconciling schemas of disparate data sources: A machine-learning approach, *in* 'ACM Sigmod Record', Vol. 30, ACM, pp. 509–520.

Edison (2012), 'The social habit'.
URL: *http://socialhabit.com/secure/wp-content/uploads/2012/07/the-soCial-Habit-2012-by-edIson-research.pdf*

FormulaPR (2011), 'Social networking in summary'.
URL: *http://www.formulapr.com/fuse/june2011/bitech.pdf*

Ganti, R. K., Ye, F. & Lei, H. (2011), 'Mobile crowdsensing: Current state and future challenges', *Communications Magazine, IEEE* **49**(11), 32–39.

Gnip (2008), 'Gnip'.
URL: *http://gnip.com/*

Guo, B., Yu, Z., Zhang, D. & Zhou, X. (2014), 'From participatory sensing to mobile crowd sensing', *arXiv preprint arXiv:1401.3090* .

Hachem, S., Pathak, A. & Issarny, V. (2013), 'Service-oriented middleware for large-scale mobile participatory sensing', *Pervasive and Mobile Computing* .

Hardt, D. (2012), 'The OAuth 2.0 authorization framework'.

Hughes, D., Crowley, C., Daniels, W., Bachiller, R. & Joosen, W. (2014), User-rank: generic query optimization for participatory social applications, *in* 'System Sciences (HICSS), 2014 47th Hawaii International Conference on', IEEE, pp. 1874–1883.

Hughes, D., Greenwood, P., Blair, G., Coulson, G., Grace, P., Pappenberger, F., Smith, P. & Beven, K. (2008), 'An experiment with reflective middleware to support grid-based flood monitoring', *Concurrency and Computation: Practice and Experience* **20**(11), 1303–1316.

Hughes, D., Thoelen, K., Horré, W., Matthys, N., Cid, J. D., Michiels, S., Huygens, C. & Joosen, W. (2009), LooCI: a loosely-coupled component infrastructure for networked embedded systems, *in* 'Proceedings of the 7th International Conference on Advances in Mobile Computing and Multimedia', ACM, pp. 195–203.

Kanhere, S. S. (2011), Participatory sensing: Crowdsourcing data from mobile smartphones in urban spaces, *in* 'Mobile Data Management (MDM), 2011 12th IEEE International Conference on', Vol. 2, IEEE, pp. 3–6.

Khoury, R., Dawborn, T., Gafurov, B., Pink, G., Tse, E., Tse, Q., Almi'Ani, K., Gaber, M., Röhm, U. & Scholz, B. (2010), Corona: energy-efficient multi-query processing in wireless sensor networks, *in* 'Database Systems for Advanced Applications', Springer, pp. 416–419.

Lane, N. D., Eisenman, S. B., Musolesi, M., Miluzzo, E. & Campbell, A. T. (2008), Urban sensing systems: opportunistic or participatory?, *in* 'Proceedings of the 9th workshop on Mobile computing systems and applications', ACM, pp. 11–16.

Li, J. & Cardie, C. (2013), Early stage influenza detection from twitter.

Madden, S. R., Franklin, M. J., Hellerstein, J. M. & Hong, W. (2005), 'TinyDB: an acquisitional query processing system for sensor networks', *ACM Transactions on database systems (TODS)* **30**(1), 122–173.

Maynard, D., Bontcheva, K. & Rout, D. (2012), 'Challenges in developing opinion mining tools for social media', p. 15.

Min, H., Scheuermann, P. & Heo, J. (2013), 'A hybrid approach for improving the data quality of mobile phone sensing', *International Journal of Distributed Sensor Networks* **2013**.

Okada, H., Itoh, T., Suzuki, K. & Tsukamoto, K. (2009), Wireless sensor system for detection of avian influenza outbreak farms at an early stage, *in* 'Sensors, 2009 IEEE', IEEE, pp. 1374–1377.

Perry, F. (2014), 'Sneak peek: Google cloud dataflow, a cloud-native data processing service'.
URL: *http://googlecloudplatform.blogspot.com/2014/06/sneak-peek-google-cloud-dataflow-a-cloud-native-data-processing-service.html*

Robinson, B., Power, R. & Cameron, M. (2013), A sensitive twitter earthquake detector, *in* 'Proceedings of the 22nd international conference on World Wide Web companion', pp. 999–1002.

Sakaki, T., Okazaki, M. & Matsuo, Y. (2010), Earthquake shakes twitter users: real-time event detection by social sensors, *in* 'Proceedings of the 19th international conference on World wide web', pp. 851–860.

Sato, K. (2012), 'An inside look at google BigQuery, white paper', *Google Inc* .

Singapore, W. A. S. (2014), 'Social, digital & mobile in APAC'.
URL: *http://www.slideshare.net/wearesocialsg/social-digital-mobile-in-apac*

Taylor, D., Lewin, J. & Strutton, D. (2011), 'Friends, fans, and followers: Do ads work on social networks?', *Business Faculty Publications* .

Vasalou, A., Joinson, A. N. & Courvoisier, D. (2010), 'Cultural differences, experience with social networks and the nature of "true commitment" in facebook', *International Journal of Human-Computer Studies* **68**(10), 719–728.

Vassiliadis, P. (2009), 'A survey of extract–transform–load technology', *International Journal of Data Warehousing and Mining (IJDWM)* **5**(3), 1–27.

Wandhöfer, T., Taylor, S., Walland, P., Geana, R., Weichselbaum, R., Fernandez, M. & Sizov, S. (2012), 'Determining citizens' opinions about stories in the news media: analysing google, facebook and twitter', *eJournal of eDemocracy & Open Government (JeDEM)* **4**(2), 198–221.

Data Cleansing during Data Collection from Wireless Sensor Networks

Md Zahidul Islam[1]*, Quazi Mamun[2] and Md. Geaur Rahman[1]

[1]School of Computing and Mathematics
Charles Sturt University, Panorama Avenue, Bathurst, NSW 2795, Australia.
[2]School of Computing and Mathematics
Charles Sturt University, Locked Bag 588, Boorooma Street, Wagga Wagga, NSW 2678, Australia.
Emails: {zislam, qmamun, grahman}@csu.edu.au

Abstract

Quality of data in Wireless Sensor Networks (WSNs) is one of the major concerns for many applications. The data quality may drop due to various reasons including the existence of missing values and incorrect values (also known as noisy or corrupt values) that can be caused by factors such as interference and machine malfunctioning. A drop in data quality may seriously impact the performance of decision support systems. Thus, it is crucial to clean the data before using them. In this paper we analyze the impact of missing values in a WSN data set (which is collected using a *Voronoi* diagram based network architecture) for the data mining tasks such as classification and knowledge discovery. While the quality of the data mining output (classification accuracy) suffers from the existence of the missing values this study shows an improvement when the missing values are imputed through our data cleansing scheme. The proposed scheme uses a corrupt data detection technique and a missing value imputation method for cleaning the data being collected from the sensor nodes. Our empirical analysis indicates the effectiveness of the proposed approach.

Keywords: WSN; data integrity; data cleansing; mobile data collector; *Voronoi* diagram

1 Introduction

The Wireless Sensor Network (WSN) has a wide range of applications in both military and civilian operations and therefore attracted huge attention. The WSNs are usually implemented in unattended and often hostile environment such as military and homeland security operations (Karlof & Wagner 2003, Douceur 2002, Newsome et al. 2004, Ye et al. 2005, Zhu, Setia, Jajodia & Ning 2004, Xiao et al. 2006, Mamun 2011). Various studies on WSN show that it is possible for an attacker to spread malicious code over the whole network by exploiting different mechanisms of sensor nodes without physical contact (Giannetsos et al. 2009, Sharma & Ghose 2011). This may disturb the data collection process from the sensor nodes and introduce incorrect and missing data.

Another type of error in sensor data takes place when sensors' energy levels are fading away (Ni et al. 2009). Usually, sensors are deployed in remote and unattended areas, and thus it is impractical to change the batteries of the sensor nodes. Sometimes sensors (such as automatic weather stations (Khan et al. 2012)) use solar energy to recharge their battery. A long night followed by a cloudy day can cause a sensor to have a flat battery. The sensing capabilities reduce with the deterioration of the energy level. In this circumstance, sensor data produced by those sensors can be erroneous or even completely missing. Erroneous or missing data can also be produced because of interferences and malfunctions.

The existence of missing values in a data set can seriously impact the performance of decision support systems. In order to explain it better, we empirically test the impact of the existence of missing values on the Intel Lab data set that is publicly available in the Intel Berkeley Research lab (*IBRL-Web [online available: http://db.lcs.mit.edu/labdata/labdata.html]* 2014). We artificially create missing values in the data set and then calculate the classification accuracy by applying the C4.5 classifier (Quinlan 1996) on the original data set and the data set having missing values. In this empirical test, 10% of the total attribute values are considered to be missing in the data set with missing values. The procedures of creating missing values and calculating classification accuracy are presented in details in Section 4. The classification accuracy on the data set without missing values and the data set with missing values are presented in Figure 1. We can see from the figure that the classification accuracy drops significantly on the data set with missing values. This suggests that there is a need to clean the erroneous sensor data set obtained from a WSN.

Figure 1: The classification accuracies of C4.5 classifiers on the data sets without missing values and with missing values.

The traditional approach for data cleansing requires manual review of the data where a domain expert reviews the collected data looking for outliers and unusual events. Data cleansing using traditional methods can be slow, expensive and tedious. Often the manual review of collected data can take several months delaying the release of the

*The fist author would like to thank the Faculty of Business COMPACT Fund R4 P55 in Charles Sturt University, Australia.

collected data. Moreover, it may not be able to scale up when a large amount of data is collected (Dereszynski & Dietterich 2007). Therefore, it is crucial to automate the data cleansing process.

Automatic data cleansing is an active research area. A method was proposed by Deresynski and Dietterich to clean sensor data automatically (Dereszynski & Dietterich 2007). It first uses the long-term historical records of a single sensor and thereby derive a probabilistic model of the sensor's behavior over time. Next, the quality of future readings of the sensor are assessed by computing their likelihood given the model. Based on the assessment of the future readings the state of the sensor is evaluated as either good or bad. If a state of a sensor is evaluated as bad then the method ignores the data produced by the sensor during the state. Instead the method then predicts the data based on the model and considers the predicted data for analysis. However, the method accepts the sensor data if the state of the sensor is evaluated to be good.

The performance of the method (Dereszynski & Dietterich 2007) heavily depends on the probabilistic model which is built from the historical data of a single sensor of interest. In a sensor network typically there are a number sensors surrounding a sensor. The data produced by the surrounding sensors could be used to make the model more robust. The surrounding sensors that monitor neighboring and overlapping areas often provide redundant data. This redundancy can be exploited to learn the interdependencies of the senors which in turn can lead to more accurate predictions without requiring long-term historical records (Ramirez 2011).

Multiple sensors are used by a cleansing method (Ramirez 2011) which predicts/assesses the data of a given sensor by considering the sensed data from a number of surrounding sensors. Due to the computational expense of the method it is applied as a post-processing step instead of an on-line monitoring step. Three learning algorithms namely artificial neural network (ANN), k-nearest neighbors (KNN), and locally weighted regression (LWR) are used for the prediction of sensor data. A sensor reading/datum is then compared with the predicted datum in order to compute the likelihood of the sensor reading being erroneous. If a sensor reading is considered to be erroneous then it is replaced by a new value which is estimated based on the predicted value and the actual sensed value.

The learning algorithms used in the method have some limitations. For example, the ANN algorithm generally takes a long training time. LWR needs to deal with the whole training data set every time it processes a new instance.

A possible problem in using historical data for the identification of corrupt values and imputation of missing values can be the dissimilarity between the current and historical data in terms of their patterns. Therefore, some data cleansing techniques (Rahman & Islam 2014, 2011, 2013*a*,*b*, Cheng et al. 2012, Rahman et al. 2012) first find the groups/clusters of similar data. They then clean a datum by using the properties of the data that are similar to the datum value being cleaned.

There are some offline cleaning methods (Mayfield et al. 2010) that are used in a pre-processing step of sensor data analyses. Often the collected data are analyzed through various statistical and data mining techniques for knowledge discovery and future prediction. Therefore, it is common to prepare a static data set and then analyze the data off-line.

In this paper we first examine the impact of missing values in a WSN data set on the data analysis using a classifier. We find that the classification accuracy drops when there are missing values in a WSN data set (see Figure 1). We then present a data cleansing scheme where we first identify incorrect data. The identified incorrect data are then artificially considered to be missing. Finally, these artificial missing values along with any other natural missing values that exist in the data set are imputed.

An advantage of the proposed scheme is the use of the data collected from neighboring sensors within a close time period. Therefore, due to the similarity of the geographic locations of the sensors and the time period when the data are collected the data are expected to be similar to each other. This similarity of the data supports the corrupt data detection and missing value imputation techniques to achieve better results.

Our empirical analysis indicates that the classification accuracy improves when a classifier is built on the data set where missing values are imputed (see Figure 5). Additionally we present detailed experimental result indicating successful corrupt data detection and missing value imputation in a WSN data set. The best noise detection performance achieved in our experimentation on the Intel Lab data set is a detection of as high as 42.7% of the total incorrect values (i.e. 42.7% Error Recall) with 83% Error Precision (see Table 2). Similarly, we achieve 91.7% of the best possible imputation performance (see Table 4).

The remainder of the paper is organized as follows. We present the network architecture model for the proposed data cleansing approach in Section 2. Section 3 presents our data cleansing scheme. In Section 4, we describe the experimental setup and present the simulation results. Finally in Section 5, we present the conclusion of the study.

2 Network Architecture Model

The proposed data cleansing approach can be deployed over all hierarchical networks such as cluster based, tree based and chain oriented network. We in this study consider the chain oriented topology where multiple chains can be constructed. All the chains are restricted to *Voronoi cells* (Mamun 2013, Mamun et al. 2013). Additionally mobile data collectors (MDC) are used to collect data from the deployed sensor nodes (Mamun 2011).

Figure 2 presents the architectural model of the chain oriented topology. The leader nodes are presented using the dots that collect data from the sensor nodes within a *Voronoi cell* that the leader node represents. The leader nodes then send the data to the mobile data collectors (MDCs) when the MDCs visit the polling points. The MDCs visit the polling points regularly.

Various approaches in the literature (Mamun 2011) extend the data gathering scheme for large scaled wireless sensor networks where they use multiple MDCs and the spatial division multiple access (SDMA) technique. Figure 2 shows an example where two MDCs simultaneously travel within the network in order to collect data from the leaders.

Since the Base Station (BS) is generally situated outside the sensing field the long distant sensor nodes are likely to deplete energy much faster than some short distant nodes if the long distant sensor nodes were transmitting data directly to the BS (Zhao & Yang 2012*a*, Chen et al. 2011, Zhao & Yang 2012*b*).

Therefore, some recent studies(Liang et al. 2013, Zhang & Chen 2011, Fei et al. 2011) proposed the use of mobile devices (sink mobility) for the data gathering purpose. The mobile data collection devices can prolong the network lifetime to a great extent by supporting a balance energy consumption among sensor nodes (Zhi et al. 2010, Ma & Yang 2008). Therefore, we utilize multiple MDCs and use the spatial division multiple access (SDMA) technique (Mamun 2011) by dividing the sensing field into a number of non-overlapping regions. Each region is taken care of by using an MDC.

An MDC gathers data from the leaders in a region by traversing through the region. A *Voronoi* diagram is con-

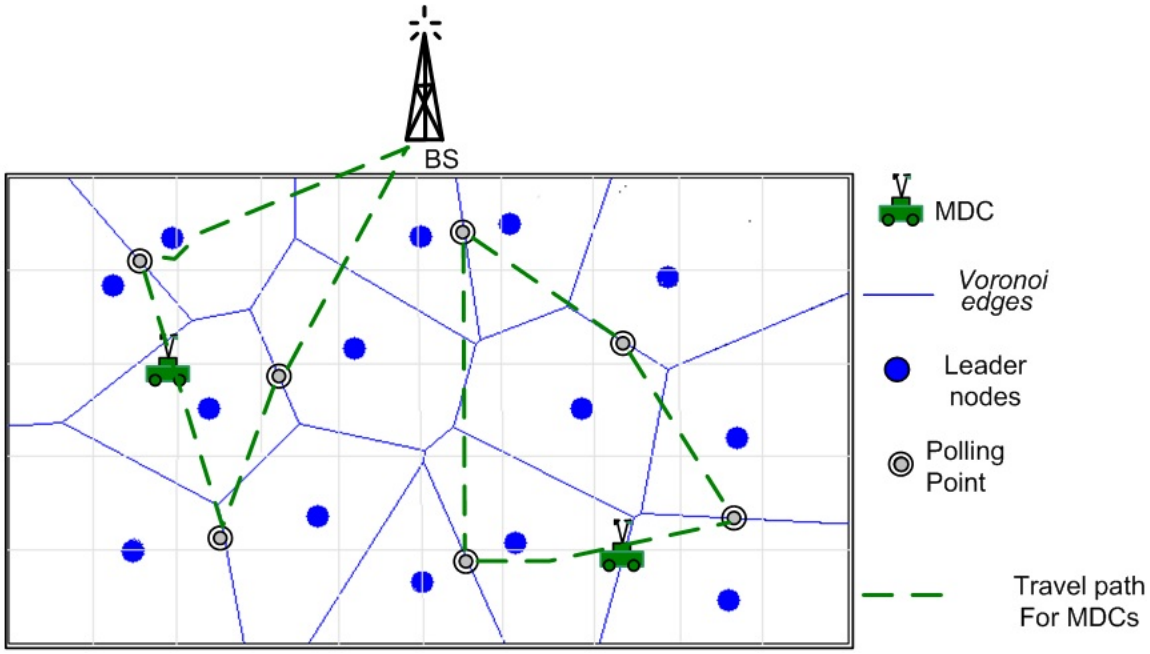


Figure 2: The network architecture model for the proposed anomaly node detection technique (Mamun et al. 2013).

structed with respect to the leader nodes in order to determine the traversal paths of the MDCs. Each MDC is equipped with two antennas in order to allow it to take advantage of the SDMA technique for collecting data from two distinct compatible leader nodes in the same region concurrently. This can reduce the data collection time allowing an MDC to travel to a region more frequently.

We consider a number of polling points (presented in Figure 2 as a dot within a circle) where an MDC stops to collect data from the leader nodes. Polling points are supposed to be in the middle of the leader nodes in a region so that an MDC can take the full advantage of the SDMA technique. An MDC can decode the multiplexing signals concurrently transmitted by the leader nodes of a region. A detailed discussion on the physical layer for concurrent data uploading is provided in the literature (Mamun 2011). We assume that MDCs have access to power supply through the BSs and are not power poor.

Therefore, the MDCs regularly collect data from the leader nodes and transfer the data to the bases stations periodically. Hence, the base station collects all the data from the sensor nodes through the leader nodes and MDCs. It then prepares a data set for further data analysis through statistical approaches and data mining techniques in order to discover knowledge and predict future.

3 The Data Cleansing Approach in the Proposed Scheme

Based on the assumption that the MDCs have access to sufficient power supply we consider them to be computationally powerful enough to run data cleansing techniques when they collect the data from the leader nodes. An MDC uses the following five steps to carry out the data cleansing tasks as shown in Figure 3.

Step 1: Collect the Dataset D_o and Make a Copy D_c.
Step 2: Identify Corrupt Values in D_c.
Step 3: Consider the Corrupt Data as Missing Values in D_c.
Step 4: Impute All Missing Values in D_c.
Step 5: Return D_o, D_c and a report R to the BS.

Step 1: Collect the Dataset D_o and Make a Copy D_c.

An MDC travels to a polling point and collects data from the leader nodes that are allocated to the polling point. The collected data are stored in a two dimensional dataset D_o where rows represent records and columns represent attributes. Each record represents the data collected by a sensor at a particular time. A sensor may collect data on a number of attributes such as temperature, humidity, light and voltage. Each column represents the data on a particular attribute. Therefore, a record r_i contains the data on the attributes as collected by a sensor at a particular time.

The notation r_{ij} represents the j-th attribute value of the i-th record. A data set contains a set of attributes $a = \{a_1, a_2, \cdots a_n\}$ where a_k represents the k-th attribute. The k-th attribute can have a range of possible values called the domain of the attribute. For example, the domain of the attribute temperature can be [-30, 50] meaning that the lowest possible value in this example is -30 degree Celsius and the highest possible value is 50 degree Celsius. The continuous numerical values can be discretized into categories such as [-30, 0], [1, 30] and [31, 60]. The notation a_{kp} represents the p-th category of the k-th attribute.

Every sensor collects data at a regular interval such as once every 30 seconds. Hence, if there are 50 sensors allocated to a polling point and an MDC travels to the polling point once every 30 minutes then each time it travels to the polling point it collects data from all 50 sensors for this period and thereby creates an original dataset D_o that has 50 X 30 X 2 = 3000 records. Each record contains values on the attributes. Some of the values can be incorrect and some of the values can be completely missing. The proposed scheme makes a copy of the dataset D_o into D_c.

Step 2: Identify Corrupt Values in D_c.

We then use an existing corrupt data detection technique such as CAIRAD (Rahman et al. 2012) that identifies any record having a possible incorrect value. It also identifies the value which is suspected to be corrupt. Note that, the proposed scheme can use any suitable corrupt data detection technique and is not limited to CAIRAD only.

CAIRAD first discretizes D_c. It then computes the actual co-appearances of a pair of attribute values (dis-

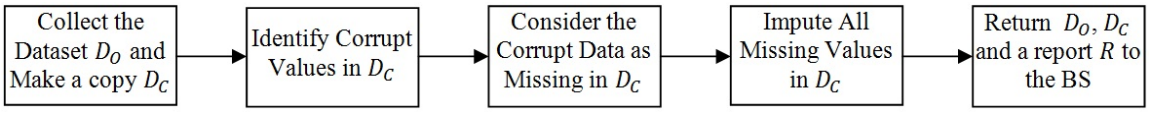


Figure 3: The Data Cleansing Steps of an MDC.

cretized) belonging to two different attributes. For example, the actual co-appearance of a_{kp} and a_{ln} is the actual number of times the p-th category of the k-th attribute co-appears in with the n-th category of the l-th attribute in the same record; for all records over the whole dataset. It also calculates an expected number of co-appearances of the pair of values considering that each value of an attribute is equally likely to appear. Both actual and expected co-appearances are computed for all possible pair of values.

Now for each record r_i, CAIRAD explores any incorrect attribute value. Each value of a pair having a significantly lower actual number of co-appearances than the expected number of co-appearances of the values receives a score of 1. On the other hand if the actual number of co-appearance is not significantly less than the expected number of co-appearances then each value of the pair receives the score of 0. If there are M attributes in a dataset then each attribute value of the record r_i is tested for all other $(M-1)$ attributes. Finally, each value of r_i receives a total score. If the score of an attribute value exceeds a threshold then the value is considered to be corrupt.

Step 3: Consider the Corrupt Data as Missing Values in D_c.

The attribute values $r_{ij}; \forall i, j$ that are identified to be corrupt in Step 2 are now considered to be missing/unavailable. Additionally, it is possible that there are some other attribute values in D_c that are originally missing. That is, the values were missing when the data from the nodes were first collected. Both type of missing values are considered to be missing in D_c.

Step 4: Impute All Missing Values in D_c.

All the missing values are then imputed using an existing imputation technique such as FIMUS (Rahman & Islam 2014). Note that, the proposed scheme can use any existing techniques and is not limited to FIMUS only.

FIMUS imputes the missing values of D_c based on the co-appearances of values belonging to two attributes, similarities of values belonging to an attribute and correlations of attributes. The basic concept of the technique is impute a missing value r_{ij} with a domain value (of the j-th attribute) which has a high co-appearance with other attribute values (i.e. $r_{ik}; \forall k$) of the record r_i, and the correlations of other attributes with the j-th attribute. Moreover, the technique also considers the similarity between the value $r_{ik}; \forall k$ and other domain values of the k-th attribute.

Step 5: Return D_o, D_c and a report R to the BS.

Finally, when the missing values are imputed in D_c the proposed scheme returns D_o and D_c to the BS. It also prepares a report R that contains a flag for the values that have been modified in D_c either because it was originally missing or it was identified to be corrupt. Therefore, analysts can double check whether the modifications are sensible or not, if necessary.

4 Simulation and Analysis

4.1 Data Set

In this section we use the publicly available Intel Lab data (*IBRL-Web [online available: http://db.lcs.mit.edu/labdata/labdata.html]* 2014) to demonstrate the usefulness of the proposed data cleansing scheme to improve the quality of a Wireless Sensor Network dataset. In the Intel Lab data set there are altogether 54 sensor nodes each of which collects data on temperature, humidity, light and voltage once every 30 seconds. We consider that all these 54 nodes are allocated to a polling point and the MDC travels to the polling point once every 30 minutes. Therefore, we copy 60 records from each of the 54 sensors for the same 30 minutes period and thereby produce a dataset with 3240 records. We call the dataset D. However, we realize that in D we have 120 records with missing values. We first remove the records that originally have missing values and thereby obtain a pure data set having 3120 records without any missing values. We use the pure data set in the experimentation.

4.2 Simulation of Corrupt Data

We then artificially create some corrupt values for which the actual values are known from the dataset. We use the following assumptions while artificially creating the corrupt values (Rahman et al. 2012).

We consider four noise patterns namely Simple, Medium, Complex and Blended in which if a record has a noisy value then in the Simple pattern the record can have at most one noisy value. In the Medium pattern a record can have 2 to 50% of attributes with noisy values, whereas in the Complex pattern 50% to 80% of attributes of a record can have noisy values. However, the Blended pattern is the mixture of the three patterns. The Blended pattern contains 25% records having noisy values in simple pattern, 50% in medium pattern and 25% in complex pattern. Since the Blended pattern combines of all three noise patterns we may expect a natural scenario based on the Blended pattern.

For each of noise pattern, we use various noise levels (1%, 3%, 5% and 10%) where x% noise level means x% of the total attribute values (not records) of a data set are noisy. There are altogether 3120 records and each record has four attribute values. Therefore, there are 3120 X 4 = 12480 values out of which say 1% values (i.e. 124 values) are made noisy. Since for the Simple pattern each record can have at most one noisy value there are 124 noisy records.

Moreover, we use three different noise outside ranges 10%, 30%, and 50% based on the domain of an attribute. For example, 10% of the total noisy values (i.e. 12 values) can have the noisy value outside the original domain of the attributes. If the domain of an attribute is [0, 10] then 10% of the noisy values are outside the domain range.

We also consider two noise models namely uniformly distributed (UD) and Overall while creating the noisy values. In the UD model noisy values are equally distributed among the attributes, whereas in the Overall model noisy values are not equally distributed among the attributes. That is, in the worst case scenario all corrupt values may belong to a single attribute only.

Based on the 4 noise levels, 3 noise outside ranges, 2 noise models and 4 noise patterns we have altogether 96 $(4 \times 3 \times 2 \times 4)$ noise combinations. For each of the combinations we create 10 data sets with noisy values. For example, for the combination having "simple" noise pattern, "1%" noisy values, "10%" noise range, and "overall"

model (see Table 5) we generate 10 data sets with noisy values. We therefore create altogether 960 data sets (96 combinations $\times$ 10 data sets/combination) for the Intell Lab data set.

We then apply CAIRAD (Rahman et al. 2012) on the noisy dataset and identify the noisy values. The accuracy of the identification is evaluated through Error Recall (ER) (Zhu, Wu & Yang 2004) and Error Precision (EP) (Zhu, Wu & Yang 2004). ER is the ratio of the correctly identified corrupt values to the total number of the corrupt values. EP is the ratio of the total number of correctly identified corrupt values to the total number of identified corrupt values. The values of both ER and EP vary between 0 and 1, where a higher value indicates a better noise detection.

4.3 Analysis of the Performance of a Corrupt Data Detection Technique

The overall average noise detection performance of the Intel Lab data set is presented in Table 1. The table shows that the noise detection performance in terms of ER and EP are 0.130 and 0.514, respectively, in which each of them is the average value of the performance indicators on 960 data sets having noisy values. The EP value 0.514 means that 51.4% of the total noisy values identified by CAIRAD are originally noisy. Considering the EP values typically achieved by different noise detection techniques on the regular data sets the EP value obtained by CAIRAD on the Intel Lab data set is very encouraging. For example, the EP value achieved by three different techniques namely EDIR, RDCL and CAIRAD on the Adult data set (publicly available in the UCI machine learning repository (Frank & Asuncion 2010)) is 0.171, 0.348 and 0.545, respectively (Rahman et al. 2012).

Table 1: Overall average noise detection performance on Intel Lab data set.

Data Cleansing Technique	ER	EP
CAIRAD	0.130	0.514

In Table 2 we also present the maximum ER and EP values achieved by CAIRAD in our experiments on the Intel Lab data set. CAIRAD achieves an ER value 0.427 for the combination having "1%" noisy values, "50%" noise range, "UD" noise model and "medium" noise pattern. On the other hand, CAIRAD achieves an EP value 0.830 for the combination having "10%" noisy values, "50%" noise range, "UD" noise model and "complex" noise pattern. This indicates that 83% of the identified noisy/incorrect values are actually noisy. Moreover, 42.7% of the actually noisy values are identified in the data set. This is a reasonably high achievement in noise detection which is expected to increase the quality of any data analyses including data mining and statistical analyses.

Table 2: Maximum achievable noise detection performance by CAIRAD on Intel Lab data set.

Data Cleansing Technique	ER	EP
CAIRAD	0.427	0.830

It is also worth noting that CAIRAD uses a user defined threshold called a co-appearance score threshold (λ) that is used to determine whether a value is noisy or correct. CAIRAD uses a default value for λ which is 0.3. The results that we present in Table 1 and Table 2 are obtained based on the default value. However, it may be possible to achieve even higher ER and EP by adjusting the λ values. Thus we calculate the ER and EP values for different λ values for the noise combination having "1%" noisy values, "50%" noise range, "UD" noise model and "medium" noise pattern as shown in Figure 4. From the figure we can see that CAIRAD achieves a high ER value (which is 67.7%) for $\lambda = 0.1$. The ER value indicates that 67.7% of the total noisy values are detected by CAIRAD. The noise detection is expected to increase the data quality and subsequent data analysis in the sensor network data set.

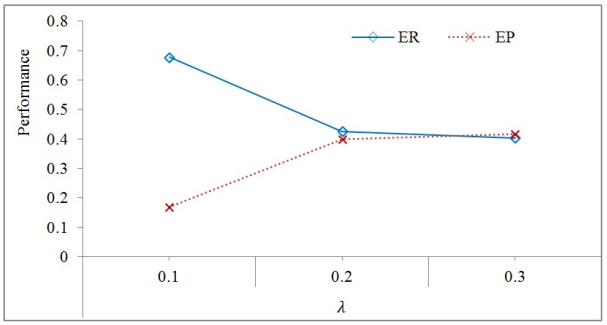


Figure 4: The noise detection performance for different λ values.

4.4 Simulation of Missing Values

After the noise detection we now aim to correct the noisy values. We consider the artificially created noisy values as missing and then impute them by using an imputation technique. Note that a data set may also originally contain missing values. For example, there are 120 records of the Intel Lab data set that originally have missing values.

For the 4 noise levels, 3 noise outside ranges, 2 noise models and 4 noise patterns we have altogether 96 ($4 \times 3 \times 2 \times 4$) missing combinations. For each of the combinations 10 data sets with missing values are created. For the combination having "simple" noise pattern, "1%" noisy values, "10%" noise range, and "overall" model (see Table 6) 10 data sets with missing values are generated. Therefore, altogether 960 data sets (96 combinations $\times$ 10 data sets/combination) with missing values are created for the Intell Lab data set. The original values of the artificially created missing values are known.

We then apply FIMUS on the dataset to impute the missing values. After the imputation the accuracy of the imputation is evaluated through two commonly used metrics; Index of Agreement (d_2) (Willmott 1982) and Root Mean Square Error ($RMSE$) (Junninen et al. 2004). Both metrics estimate the difference between an original value and an imputed value, where the closer the values the better the imputation. The values of d_2 vary between 0 and 1, where a higher value indicates a better imputation. The $RMSE$ values vary between 0 and infinity, where a lower $RMSE$ value indicates a better imputation.

4.5 Analysis of the Performance of a Missing Value Imputation Technique

We now present an overall (i.e. the average value of the performance indicators on 960 data sets having missing values) imputation performance of the Intel Lab data set in Table 3. The d_2 of the imputed values is 0.888 which is very high meaning that the imputed values are very close to the original values. It is worth mentioning that

the maximum d_2 value could be 1.000 which would also indicate that the original and imputed values are the same. Therefore, the d_2 value of 0.888 means that we achieve 88.8% of the maximum possible imputation accuracy, By the maximum possible imputation accuracy we mean the case where all values would be imputed accurately. Similarly, the $RMSE$ of the imputed values is 0.121 which again indicates a good imputation, where the minimum possible $RMSE$ value could be zero and the maximum possible $RMSE$ value could be infinity. A lower $RMSE$ value indicates a better imputation.

Table 3: Overall average imputation performance on Intel Lab data set.

Data Cleansing Technique	d_2	$RMSE$
FIMUS	0.888	0.121

In Table 4 we present the maximum d_2 and $RMSE$ values achieved by FIMUS in our experiments on the Intel Lab data set. FIMUS achieves a d_2 value 0.917 for the combination having "1%" noisy values, "50%" noise range, "UD" noise model and "simple" noise pattern. The high d_2 value indicates a high agreement between the original and imputed values. Besides, for the same combination FIMUS achieves a low $RMSE$ value which is 0.104 that also indicates a high imputation accuracy by FIMUS on the Intel Lab data set.

Table 4: Maximum achievable imputation performance by FIMUS on Intel Lab data set.

Data Cleansing Technique	d_2	$RMSE$
FIMUS	0.917	0.104

4.6 Analysis of the Effectiveness of Imputation based on the Prediction Accuracy

We now analyze the effectiveness of imputation based on the prediction accuracy of a decision tree algorithm such as C4.5 (Quinlan 1996). Note that since all attributes of the Intel Lab data set are numerical we first categorize the values of the attribute "voltage" by applying an existing discretization algorithm called PD (Yang & Webb 2009) so that the attribute can be considered as a class attribute while building a decision tree. We then use a 10 fold cross validation to evaluate the classification accuracy without missing values, with missing values and with imputed values. We now explain the procedure as follows. For each fold, the data set D having n records is divided into two sub data sets, namely the testing data set $D_{testing}$ and training data set D_O. The testing data set contains $\frac{n}{10}$ records of D and the training data set contains the remaining $\frac{9n}{10}$ records of D.

From the training data set we then create three data sets as follows. The first data set is the Original training data set D_O in which there are no records with missing values. The second data set D_C is obtained as follows. Using 10% missing ratios, Overall missing model and Blended missing pattern we artificially create missing values in D_O and get a data set D_F having missing values. We then remove the records having missing values from D_F and get a data set D_C that contains records without any missing values. The third data set D'_F is obtained by imputing the missing values of D_F. We then build decision trees (DTs), namely DT_O, DT_C and DT'_F by applying the $C4.5$ algorithm on D_O, D_C and D'_F, respectively.

We next calculate the classification accuracies of the DTs, on the testing data set $D_{testing}$ as shown in Figure 5. We can see that the classification accuracy of the DT_C, which is built from the data set D_C (*with missing values*), is much lower than the classification accuracy of the DT_O, which is built from the original data set (*without missing values*). We achieve a higher classification accuracy by the DT'_F, which is built from the imputed data set D'_F (*with imputed values*), than the DT_C, which is built from the D_C. The improvement in accuracy indicates the usefulness of the imputation approach in wireless sensor networks data sets.

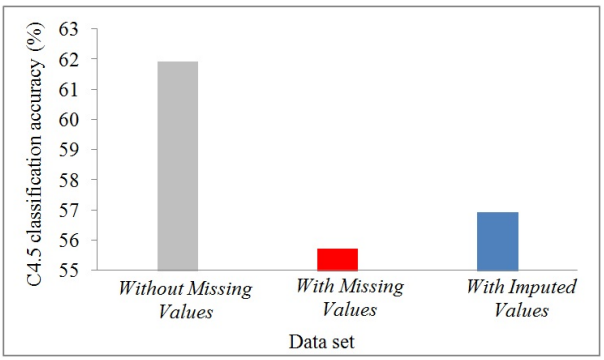


Figure 5: The classification accuracy of C4.5 classifier on the data sets without missing values, with missing values and with imputed values.

4.7 Details Experimental Results on Corrupt Data Detection and Imputation

We also present the detail noise detection and imputation performances for all 96 combinations in Table 5 and Table 6. We present the noise detection performance of CAIRAD based on ER and EP for 96 noise combinations in Table 5. The average values of the performance indicators on 10 data sets having noisy values for each combination of noise level, noise range, noise model and noise pattern is presented in the table. For example, there are 10 data sets having noisy values with the combination Com_2 of "1%" noise level, "10%" noise range, "Overall" noise model and "Medium" noise pattern. In Table 5 we can see that the average of ER and EP for the data sets having Com_2 is 0.297 and 0.276, respectively.

Moreover, in Table 6 we present the imputation performance of FIMUS based on d_2 and $RMSE$ for 96 missing combinations. The average values of the performance indicators on 10 data sets having missing values for each combination of noise level, noise range, noise model and noise pattern is presented in the table. For example, there are 10 data sets having missing values with the combination Com_1 of "1%" noise level, "10%" noise range, "Overall" noise model and "Simple" noise pattern. In Table 6 we can see that the average of d_2 and $RMSE$ for the data sets having Com_1 is 0.913 and 0.107, respectively.

5 Conclusion

In this study we first discuss an Mobile Data Collector (MDC) based approach of the sensor network data collection. Due to a number of reasons including flat battery, equipment malfunctioning and compromise wireless sensor network data may contain incorrect and missing values. We therefore discuss the quality of data mining results obtained from an uncleaned data set and compare it

Table 5: Noise detection performance of a data cleansing technique (such as CAIRAD) on the Intel Lab data set.

Noise combinations				ER	EP
Noise Level	Noise Range	Noise Model	Noise Pattern		
1%	10%	Overall	Simple	0.108	0.232
			Medium	0.297	0.276
			Complex	0.300	0.292
			Blended	0.223	0.273
		UD	Simple	0.087	0.217
			Medium	0.248	0.300
			Complex	0.238	0.301
			Blended	0.154	0.254
	30%	Overall	Simple	0.128	0.278
			Medium	0.306	0.302
			Complex	0.327	0.334
			Blended	0.239	0.317
		UD	Simple	0.122	0.266
			Medium	0.349	0.331
			Complex	0.315	0.322
			Blended	0.240	0.308
	50%	Overall	Simple	0.121	0.268
			Medium	0.419	0.410
			Complex	0.369	0.382
			Blended	0.264	0.379
		UD	Simple	0.152	0.350
			Medium	0.427	0.406
			Complex	0.402	0.400
			Blended	0.265	0.359
3%	10%	Overall	Simple	0.035	0.285
			Medium	0.150	0.445
			Complex	0.143	0.446
			Blended	0.093	0.371
		UD	Simple	0.032	0.260
			Medium	0.128	0.436
			Complex	0.158	0.460
			Blended	0.087	0.387
	30%	Overall	Simple	0.045	0.418
			Medium	0.195	0.587
			Complex	0.200	0.599
			Blended	0.126	0.560
		UD	Simple	0.049	0.442
			Medium	0.190	0.547
			Complex	0.204	0.589
			Blended	0.130	0.550
	50%	Overall	Simple	0.054	0.466
			Medium	0.266	0.671
			Complex	0.231	0.655
			Blended	0.173	0.643
		UD	Simple	0.054	0.467
			Medium	0.270	0.678
			Complex	0.246	0.643
			Blended	0.168	0.640

Noise combinations				ER	EP
Noise Level	Noise Range	Noise Model	Noise Pattern		
5%	10%	Overall	Simple	0.030	0.419
			Medium	0.088	0.503
			Complex	0.083	0.478
			Blended	0.048	0.444
		UD	Simple	0.022	0.359
			Medium	0.083	0.475
			Complex	0.097	0.516
			Blended	0.042	0.378
	30%	Overall	Simple	0.027	0.449
			Medium	0.151	0.682
			Complex	0.148	0.685
			Blended	0.081	0.601
		UD	Simple	0.029	0.472
			Medium	0.136	0.654
			Complex	0.133	0.653
			Blended	0.079	0.590
	50%	Overall	Simple	0.025	0.511
			Medium	0.149	0.751
			Complex	0.166	0.737
			Blended	0.095	0.677
		UD	Simple	0.027	0.499
			Medium	0.160	0.735
			Complex	0.162	0.725
			Blended	0.103	0.706
10%	10%	Overall	Simple	0.015	0.597
			Medium	0.040	0.564
			Complex	0.044	0.577
			Blended	0.020	0.453
		UD	Simple	0.014	0.529
			Medium	0.044	0.592
			Complex	0.041	0.582
			Blended	0.023	0.517
	30%	Overall	Simple	0.013	0.599
			Medium	0.054	0.742
			Complex	0.061	0.763
			Blended	0.024	0.586
		UD	Simple	0.013	0.656
			Medium	0.061	0.748
			Complex	0.062	0.715
			Blended	0.024	0.608
	50%	Overall	Simple	0.009	0.663
			Medium	0.056	0.818
			Complex	0.053	0.828
			Blended	0.025	0.770
		UD	Simple	0.007	0.631
			Medium	0.059	0.819
			Complex	0.055	0.830
			Blended	0.023	0.694

with the result obtained from a clean data set (see Figure 1). It shows that the data mining quality can significantly drop in the presence of incorrect and missing data.

We then present a data cleansing scheme to identify the incorrect data and impute all missing values. We suggest to apply the data cleansing techniques within an MDC even before it transmits the data to the base station. An advantage of the approach can be the utilization of the traveling time when the MDC is moving between two polling points. Once the missing values are imputed we carry out the data mining analysis on the imputed data set and show the improvement in prediction accuracy of the classifier built from the imputed data set (see Figure 5). Moreover, we also present the accuracy of incorrect data identification and missing value imputation that suggest the effectiveness of the data cleansing approaches used in this study for the sensor network data set (see Table 1 to Table 6).

As part of our future work, we aim to propose novel corrupt data detection and missing value imputation techniques that are specifically catered for the wireless sensor network data set. For example, although the proposed data cleansing scheme uses data from neighboring sensors for a close time period the actual cleansing techniques (CAIRAD and FIMUS) do not take further advantage of time series data. It would be interesting to investigate whether further consideration of the time could increase the accuracy of incorrect data detection and missing value imputation. Since the sensors in the wireless network have limited energy, it would also be interesting to investigate the efficiency of our proposed algorithm in such domains. Moreover, we also aim to compare a number of existing techniques with proposed techniques in future.

References

Chen, Y., Tang, Y., Xu, G., Qian, H. & Xu, Y. (2011), A data gathering algorithm based on swarm intelligence and load balancing strategy for mobile sink, *in* 'Intelligent Control and Automation (WCICA), 2011 9th World Congress on', IEEE, pp. 1002–1007.

Cheng, K., Law, N. & Siu, W. (2012), 'Iterative bicluster-based least square framework for estimation of missing values in microarray gene expression data', *Pattern Recognition* **45**(4), 1281–1289.

Dereszynski, E. W. & Dietterich, T. G. (2007), 'Probabilistic models for anomaly detection in remote sensor data streams', *in Proceedings of the Twenty-Third*

Table 6: Imputation performance of a data cleansing technique (such as FIMUS) on the Intel Lab data set.

Missing combinations				d_2	$RMSE$
Noise Level	Noise Range	Noise Model	Noise Pattern		
1%	10%	Overall	Simple	0.913	0.107
			Medium	0.878	0.123
			Complex	0.861	0.130
			Blended	0.896	0.117
		UD	Simple	0.896	0.117
			Medium	0.850	0.141
			Complex	0.899	0.113
			Blended	0.880	0.122
	30%	Overall	Simple	0.888	0.122
			Medium	0.887	0.123
			Complex	0.867	0.134
			Blended	0.882	0.121
		UD	Simple	0.880	0.122
			Medium	0.884	0.117
			Complex	0.879	0.128
			Blended	0.880	0.128
	50%	Overall	Simple	0.906	0.108
			Medium	0.859	0.136
			Complex	0.889	0.120
			Blended	0.883	0.125
		UD	Simple	0.917	0.104
			Medium	0.863	0.132
			Complex	0.885	0.121
			Blended	0.908	0.115
3%	10%	Overall	Simple	0.894	0.117
			Medium	0.873	0.129
			Complex	0.870	0.134
			Blended	0.880	0.128
		UD	Simple	0.896	0.121
			Medium	0.884	0.120
			Complex	0.874	0.129
			Blended	0.887	0.120
	30%	Overall	Simple	0.908	0.112
			Medium	0.876	0.127
			Complex	0.886	0.120
			Blended	0.889	0.123
		UD	Simple	0.911	0.108
			Medium	0.885	0.121
			Complex	0.899	0.115
			Blended	0.897	0.116
	50%	Overall	Simple	0.912	0.107
			Medium	0.888	0.123
			Complex	0.879	0.125
			Blended	0.895	0.120
		UD	Simple	0.909	0.111
			Medium	0.872	0.129
			Complex	0.882	0.124
			Blended	0.891	0.123

Missing combinations				d_2	$RMSE$
Noise Level	Noise Range	Noise Model	Noise Pattern		
5%	10%	Overall	Simple	0.901	0.114
			Medium	0.882	0.124
			Complex	0.871	0.129
			Blended	0.893	0.119
		UD	Simple	0.889	0.122
			Medium	0.882	0.124
			Complex	0.883	0.123
			Blended	0.881	0.125
	30%	Overall	Simple	0.911	0.109
			Medium	0.879	0.125
			Complex	0.879	0.126
			Blended	0.879	0.127
		UD	Simple	0.908	0.112
			Medium	0.874	0.129
			Complex	0.886	0.122
			Blended	0.890	0.122
	50%	Overall	Simple	0.910	0.111
			Medium	0.880	0.123
			Complex	0.870	0.132
			Blended	0.888	0.122
		UD	Simple	0.904	0.114
			Medium	0.881	0.123
			Complex	0.887	0.120
			Blended	0.881	0.125
10%	10%	Overall	Simple	0.904	0.115
			Medium	0.886	0.122
			Complex	0.886	0.123
			Blended	0.892	0.119
		UD	Simple	0.910	0.111
			Medium	0.880	0.124
			Complex	0.883	0.125
			Blended	0.887	0.122
	30%	Overall	Simple	0.901	0.116
			Medium	0.888	0.121
			Complex	0.879	0.123
			Blended	0.890	0.120
		UD	Simple	0.909	0.113
			Medium	0.882	0.123
			Complex	0.886	0.120
			Blended	0.892	0.120
	50%	Overall	Simple	0.903	0.116
			Medium	0.880	0.125
			Complex	0.886	0.122
			Blended	0.894	0.120
		UD	Simple	0.900	0.116
			Medium	0.878	0.123
			Complex	0.885	0.122
			Blended	0.878	0.127

Conference on Uncertainty in Artificial Intelligence (UAI2007) .

Douceur, J. R. (2002), The sybil attack, *in* 'Peer-to-peer Systems', Springer, pp. 251–260.

Fei, X., Boukerche, A. & Yu, R. (2011), An efficient markov decision process based mobile data gathering protocol for wireless sensor networks, *in* 'Wireless Communications and Networking Conference (WCNC), 2011 IEEE', IEEE, pp. 1032–1037.

Frank, A. & Asuncion, A. (2010), 'UCI machine learning repository [online available: http://archive.ics.uci.edu/ml]'. Accessed July 7, 2013.
URL: *http://archive.ics.uci.edu/ml*

Giannetsos, T., Dimitriou, T. & Prasad, N. R. (2009), Self-propagating worms in wireless sensor networks, *in* 'Proceedings of the 5th international student workshop on Emerging networking experiments and technologies', ACM, pp. 31–32.

IBRL-Web [online available: http://db.lcs.mit.edu/labdata/labdata.html] (2014). Accessed August 7, 2014.
URL: *http://db.lcs.mit.edu/labdata/labdata.html*

Junninen, H., Niska, H., Tuppurainen, K., Ruuskanen, J. & Kolehmainen, M. (2004), 'Methods for imputation of missing values in air quality data sets', *Atmospheric Environment* **38**(18), 2895–2907.

Karlof, C. & Wagner, D. (2003), 'Secure routing in wireless sensor networks: Attacks and countermeasures', *Ad hoc networks* **1**(2), 293–315.

Khan, M. A., Islam, M. Z. & Hafeez, M. (2012), Evaluating the performance of several data mining methods for predicting irrigation water requirement, *in* 'Proceedings of the Tenth Australasian Data Mining Conference-Volume 134', Australian Computer Society, Inc., pp. 199–207.

Liang, W., Schweitzer, P. & Xu, Z. (2013), 'Approximation algorithms for capacitated minimum forest problems in wireless sensor networks with a mobile sink', *Computers, IEEE Transactions on* **62**(10), 1932–1944.

Ma, M. & Yang, Y. (2008), Data gathering in wireless sensor networks with mobile collectors, *in* 'Parallel and

Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on', IEEE, pp. 1–9.

Mamun, Q. (2011), Constraint-Minimizing Logical Topology for Wireless Sensor Networks, PhD thesis, Monash University.

Mamun, Q. (2013), 'A tessellation-based localized chain construction scheme for chain-oriented sensor networks', *Sensors Journal, IEEE* **13**(7), 2648–2658.

Mamun, Q., Islam, R. & Kaosar, M. (2013), Ensuring data integrity by anomaly node detection during data gathering in wsns, *in* 'Security and Privacy in Communication Networks', Springer, pp. 367–379.

Mayfield, C., Neville, J. & Prabhakar, S. (2010), Eracer: a database approach for statistical inference and data cleaning, *in* 'Proceedings of the 2010 ACM SIGMOD International Conference on Management of data', ACM, pp. 75–86.

Newsome, J., Shi, E., Song, D. & Perrig, A. (2004), The sybil attack in sensor networks: analysis & defenses, *in* 'Proceedings of the 3rd international symposium on Information processing in sensor networks', ACM, pp. 259–268.

Ni, K., Ramanathan, N., Chehade, M. N. H., Balzano, L., Nair, S., Zahedi, S., Kohler, E., Pottie, G., Hansen, M. & Srivastava, M. (2009), 'Sensor network data fault types', *ACM Transactions on Sensor Networks (TOSN)* **5**(3), 25.

Quinlan, J. R. (1996), 'Improved use of continuous attributes in C4.5', *Journal of Artificial Intelligence Research* **4**, 77–90.

Rahman, M. G. & Islam, M. Z. (2011), A decision tree-based missing value imputation technique for data pre-processing, *in* 'Australasian Data Mining Conference (AusDM 11)', Vol. 121 of *CRPIT*, ACS, Ballarat, Australia, pp. 41–50.
URL: *http://crpit.com/confpapers/CRPITV121Rahman.pdf*

Rahman, M. G. & Islam, M. Z. (2013*a*), kdmi: A novel method for missing values imputation using two levels of horizontal partitioning in a data set, *in* 'The 9th International Conference on Advanced Data Mining and Applications (ADMA 2013), Part II, LNAI 8347', Hangzhou, China, pp. 250 – 263.

Rahman, M. G. & Islam, M. Z. (2013*b*), 'Missing value imputation using decision trees and decision forests by splitting and merging records: Two novel techniques', *Knowledge-Based Systems* **53**, 51–65.

Rahman, M. G. & Islam, M. Z. (2014), 'Fimus: A framework for imputing missing values using co-appearance, correlation and similarity analysis', *Knowledge-Based Systems* **56**, 311–327.

Rahman, M. G., Islam, M. Z., Bossomaier, T. & Gao, J. (2012), Cairad: a co-appearance based analysis for incorrect records and attribute-values detection, *in* 'Neural Networks (IJCNN), The 2012 International Joint Conference on', IEEE, pp. 1–10.

Ramirez, G. (2011), 'Assessing data quality in a sensor network for environmental monitoring'.

Sharma, K. & Ghose, M. (2011), 'Cross layer security framework for wireless sensor networks', *International Journal of Security and Its Applications* **5**(1), 39–52.

Willmott, C. (1982), 'Some comments on the evaluation of model performance.', *Bulletin of the American Meteorological Society* **63**, 1309–1369.

Xiao, M., Wang, X. & Yang, G. (2006), Cross-layer design for the security of wireless sensor networks, *in* 'Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on', Vol. 1, IEEE, pp. 104–108.

Yang, Y. & Webb, G. I. (2009), 'Discretization for naive-bayes learning: managing discretization bias and variance', *Machine learning* **74**(1), 39–74.

Ye, F., Luo, H., Lu, S. & Zhang, L. (2005), 'Statistical en-route filtering of injected false data in sensor networks', *Selected Areas in Communications, IEEE Journal on* **23**(4), 839–850.

Zhang, X. & Chen, G. (2011), Energy-efficient platform designed for sdma applications in mobile wireless sensor networks, *in* 'Wireless Communications and Networking Conference (WCNC), 2011 IEEE', IEEE, pp. 2089–2094.

Zhao, M. & Yang, Y. (2012*a*), 'Bounded relay hop mobile data gathering in wireless sensor networks', *Computers, IEEE Transactions on* **61**(2), 265–277.

Zhao, M. & Yang, Y. (2012*b*), 'Optimization-based distributed algorithms for mobile data gathering in wireless sensor networks', *Mobile Computing, IEEE Transactions on* **11**(10), 1464–1477.

Zhi, Z., Dayong, L., Shaoqiang, L., Xiaoping, F. & Zhihua, Q. (2010), Data gathering strategies in wireless sensor networks using a mobile sink, *in* 'Control Conference (CCC), 2010 29th Chinese', IEEE, pp. 4826–4830.

Zhu, S., Setia, S., Jajodia, S. & Ning, P. (2004), An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks, *in* 'Security and Privacy, 2004. Proceedings. 2004 IEEE Symposium on', IEEE, pp. 259–271.

Zhu, X., Wu, X. & Yang, Y. (2004), Error detection and impact-sensitive instance ranking in noisy datasets, *in* 'PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE', Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, pp. 378–384.

An Efficient Tagging Data Interpretation and Representation Scheme for Item Recommendation

Noor Ifada[1,2], Richi Nayak[1]
[1]School of Electrical Engineering and Computer Science
Queensland University of Technology, Australia
[2]Informatics Department
University of Trunojoyo Madura, Indonesia
noor.ifada@{qut.edu.au, if.trunojoyo.ac.id}, r.nayak@qut.edu.au

Abstract

A tag-based item recommendation method generates an ordered list of items, likely interesting to a particular user, using the users past tagging behaviour. However, the users tagging behaviour varies in different tagging systems. A potential problem in generating quality recommendation is how to build user profiles, that interprets user behaviour to be effectively used, in recommendation models. Generally, the recommendation methods are made to work with specific types of user profiles, and may not work well with different datasets. In this paper, we investigate several tagging data interpretation and representation schemes that can lead to building an effective user profile. We discuss the various benefits a scheme brings to a recommendation method by highlighting the representative features of user tagging behaviours on a specific dataset. Empirical analysis shows that each interpretation scheme forms a distinct data representation which eventually affects the recommendation result. Results on various datasets show that an interpretation scheme should be selected based on the dominant usage in the tagging data (i.e. either higher amount of tags or higher amount of items present). The usage represents the characteristic of user tagging behaviour in the system. The results also demonstrate how the scheme is able to address the cold-start user problem.

Keywords: tagging, data interpretation, data representation, item recommendation, user profile, cold-start user

1 Introduction

Learning users past tagging behaviour is essential to generate quality recommendations (Ifada and Nayak, 2014; Rendle et al., 2009). In social tagging systems, the user tagging behaviour can be inferred from the ternary relation between users, items, and tags. Users annotate items of their interest by using freely-defined tags. The task of tag-based item recommendation system is to predict an ordered list of items to a user based on users tagging behaviour. In real practice, users can use different tags to annotate the same item as well as the same tag can be used for annotating different items. This approach of freely using tags emphasizes on implementing a personalization approach in the recommendation system by using user past tagging behaviour to build the user profile.

A typical tag-based recommendation system customarily interprets the observed data as positive entries. Observed data is the state which users have expressed their interest to items by annotating those items using tags, as illustrated in Figure 1. On the contrary, how should the non-observed data be interpreted, it remains disputed. Non-observed data is a mixture of the following two states: (1) user is not interested with the items – negative entries, and (2) user might be interested to the items in the future – null values. The selection of interpretation scheme is crucial at this stage as the user profiles will be defined based on this representation.

Tensor modelling is a natural approach to represent and analyse the latent relationships inherent in a three-dimensional tagging data model (Ifada and Nayak, 2014; Symeonidis, Nanopoulos and Manolopoulos, 2010). A tensor model is commonly built by implementing the *boolean* scheme to interpret the data (Symeonidis, Nanopoulos and Manolopoulos, 2008). Rendle et al. (2009) have identified drawbacks of using the *boolean* scheme and proposed a *set-based* scheme which has shown more accurate interpretation of the tagging data. However, this *set-based* scheme is customized to the recommendation method proposed. It does not detail how various kinds of *set-based* schemes can be selected according to data characteristics for a recommendation method.

The *set-based* scheme creates an object pairwise ranking representation between the positive entries interpreted from the observed tagging data and the negative entries interpreted from the non-observed data, by two means: (1) user-item set, or (2) user-tag set. These set data represents users tagging behaviours in the system. The user-item set expresses that a user can use multiple tags to annotate an item. From this point of view, the pairwise ranking representation is created by using tags as the pairwise ranking objects. Accordingly, for

each user-item set, the pairwise ranking is generated to represent that the user is more favourable to use tags of positive entries than those of negative entries to annotate an item. Alternatively, the user-tag set indicates that a user can use the same tag to annotate different items and, consequently, the items become the pairwise ranking objects for the data representation. For each user-tag set, the pairwise ranking is generated to represent that the user is more favourable to annotate items of positive entries than those of negative entries using a tag.

The set selection is influenced by the underlying recommendation task, i.e. using the user-item set representation for tag recommendation and the user-tag set for item recommendation. However, in practice, not all recommendation methods can perform their best when they are implemented on different datasets (Gemmell et al., 2011; Ifada and Nayak, 2014; Rendle et al., 2009). Given that users tagging behaviours are captured and represented differently in each social tagging system, the data set to be used in a recommendation system should be selected based on the feature set that defines the representation of user tagging behaviour in that system, i.e. the dominant set observed from the tagging data. The user-item set is more dominant than the user-tag set when the users prefer to use less number of tags in annotating items. On the contrary, the user-tag set is considered more dominant than the user-item set when the users prefer to use more number of tags in annotating items.

The consequence of restricting the selection of interpretation scheme to the task of recommendation is that the method could possibly outperform other methods when implemented to a certain tagging data, however, it yields poor results when it is implemented on a different tagging data (Ifada and Nayak, 2014; Rendle and Schmidt-Thieme, 2010). Another limitation is that a recommendation method could not possibly be implemented with an appropriate scheme, i.e., generating item recommendation by implementing the user-tag set scheme while the dominant feature of the tagging data is user-item set (Gemmell et al., 2011), or generating tag recommendation by implementing the user-item set scheme when the dominant feature is user-tag set (Rendle et al., 2009).

This paper investigates and compares several tagging data interpretation and representation schemes that can affect the performance of tag-based item recommendation systems. These schemes vary in manners how they highlight the representative features of the user tagging behaviours on different systems. Six tagging data interpretations and representations, which are constructed using the *set-based* ranking scheme, are proposed. We analyse how each interpretation scheme forms a distinct data representation which eventually affects the recommendation results.

To test and evaluate the concept, we needed to implement the proposed representations on a recommendation method which employs a *set-based* scheme for building and learning the tensor model. We found no item recommendation method that can be used in experiments. Pairwise Interaction Tensor Factorization (PITF) (Rendle and Schmidt-Thieme, 2010), a tensor based tag recommendation method, uses the *set-based* scheme and has reported high accuracy and scalability performance. We adopted this tag recommendation method to recommend item and named it as Pairwise Interaction Tensor Factorization for Item Recommendation (PITF-I). The tag and item recommendations are two distinct tasks. Tag recommendations are generated with two specified dimensions, i.e. user and item, while the item recommendations are made with only users identity specified and, therefore, the item ranking scores must be calculated for the whole available tags before being sorted as a list of top-*N* recommendation. In this paper, we customize the PITF method so that it is applicable for both user-item and user-tag set schemes. We then show how to select the best set of interpretation scheme to be used for different datasets, given that users tagging behaviours in the different tagging systems are different from one another.

The contribution of this paper is as follows: (1) introducing comprehensive data interpretation schemes to generate user profile representation on tensor models for tag-based item recommendation, (2) proposing a process of selecting the best interpretation scheme to be used on a certain dataset, (3) adapting a pairwise ranking tensor factorization method for implementing various interpretation schemes for tag-based item recommendation, (4) establishing that an efficient user profile presentation is more important than just simply trying to get more dense data representation to generate quality recommendations, and (5) finally, showing how an efficient interpretation scheme is able to address the cold-start user problem.

The remainder of this paper is organized as follows. Section 2 details the related works. Section 3 describes briefly about the basics in the tag-based item recommendation. Section 4 explains the proposed interpretation and representation schemes, a brief description about the method used, and the process of selecting the best interpretation scheme to be used. Section 5 presents the experimental results based on real world datasets. Section 6 concludes the paper.

2 Related Work

Recommendation is a well-established research area (Adomavicius and Tuzhilin, 2005; Zhang, Zhou and Zhang, 2011). In the last few years, tensor modelling (Kolda and Bader, 2009), a well-known approach to represent and analyse latent relationships inherent in multi-dimensions data, is adapted in recommendation systems. The tensor modelling based recommendation methods have shown improved results over the matrix based methods (Rafailidis and Daras, 2013; Symeonidis, Nanopoulos and Manolopoulos, 2010). Existing tensor methods solely interpret the tagging data using *boolean* scheme to build the model and then directly use the tensor reconstruction results from the factorized tensor to generate the recommendations. The *boolean* scheme simply interprets the positive observed tagging data as 1 and the non-observed ones as 0. This scheme fits both the negative and null values as 0 which makes it difficult to predict the ranking list in the future (Rendle et al., 2009).

In other words, by using the *boolean* scheme, these approaches ignore the user's past tagging activities that

have been found most influencive in forming user likelihood for matrix-based recommended methods (Kim et al., 2010). A recent work (Ifada and Nayak, 2014) solves this problem by ranking the reconstructed tensor results utilising the past collaborative data and make the final recommendations. A pairwise tensor factorization model (Rendle and Schmidt-Thieme, 2010), for recommending tags (unlike our paper that recommends items), has also been proposed to solve the recommendation ranking problem. This method uses the *set-based* ranking interpretation scheme when building the tensor.

The *set-based* ranking scheme distinguishes the interpretation between observed and non-observed data. The representation creates pairwise ranking objects between the positive entries interpreted from the observed data and the negative entries interpreted from the non-observed data. The rest of other entries are left as null values. Within each set, the positive entries are assigned higher values than the negative ones (instead of assigning a fixed numeric values to both entries). This scheme interprets the null values as rankings that can be predicted in the future, unlike the *boolean* scheme that over fits the null values using the negative examples as 0. The model based on the *boolean* scheme tries to learn and predict a 0 for each of the negative and null case (Rendle et al., 2009). By using the *set-based* ranking scheme, entries derived by the factorised tensor model can directly be used for generating recommendations. Selecting the appropriate interpretation scheme for tensor modelling is crucial as the tensor model has to learn the total interaction between users, items, and tags represented by the data using this scheme, as well as, the model has to expose the latent relationship among those dimensions to be used for generating the recommendations.

In this paper, we investigate the process of selecting an interpretation scheme for tagging data to improve the performance of tag-based item recommendation systems on various social tagging systems that employ a system specific method to collect user tagging behaviour. As users tagging behaviours in different tagging systems are different from one another, recommendation methods are usually not able to generalize their outperformance if they are implemented on different datasets. Therefore, though the methods are able to show that they outperform their benchmark methods on a dataset, yet they do not show the similar performance when applied on another dataset (Ifada and Nayak, 2014; Rendle and Schmidt-Thieme, 2010).

We demonstrate how different interpretation scheme forms different data representation which eventually affects the recommendation results. Moreover we modify the pairwise tensor factorization method (Rendle and Schmidt-Thieme, 2010), designed for tag recommendation, to generate an ordered list of item recommendations using the six proposed tagging data interpretation and representation schemes.

To the best of our knowledge, this is the first paper that studies the data interpretation schemes for tensor models for generating tag-based item recommendations, in detail.

3 Tag-based Item Recommendation

The task of tag-based item recommendation is to generate an ordered list of items that might be of interest to a user using the collaborative tags. The list of recommended items can be learned from user past tagging behaviour inferred from observed and non-observed tagging data.

3.1 Tagging Data

Let $U = \{u_1, u_2, u_3, \dots, u_{|U|}\}$ be the list of all users, $I = \{i_1, i_2, i_3, \dots, i_{|I|}\}$ be the list of all items, and $T = \{t_1, t_2, t_3, \dots, t_{|T|}\}$ be the list of all tags. Tagging data forms a ternary relationship between users, items, and tags. The observed tagging data can be denoted as $A \subseteq U \times I \times T$, where a vector of $(u, i, t) \in A$ represents the tagging activity of user u who has tagged item i using tag t. The user-vocabulary V_u denotes the list of distinct tags that have been used by user u to annotate any items:

$$V_u = \{t | (u, *, t) \in A\}$$

Whereas the user-collection C_u denotes the list of distinct items that have been tagged by user u using any tags:

$$C_u = \{i | (u, i, *) \in A\}$$

The tagging data can be naturally modelled as a three-dimensional tensor of $\mathcal{Y}^{U \times I \times T}$. Figure 1 illustrates a tensor model representing a toy example of the observed tagging data, $\mathcal{Y}^{3 \times 4 \times 5}$ where $U = \{u_1, u_2, u_3\}$, $I = \{i_1, i_2, i_3, i_4\}$, and $T = \{t_1, t_2, t_3, t_4, t_5\}$. Each slice of the tensor represents a user matrix which contains the user tag usage for an item.

For generating tag-based item recommendations, the ranking representation of tagging data can be inferred from either using the user-item or user-tag sets. The user-item (u, i) sets are all distinct user-item combinations in A as a user can annotate an item with multiple tags. The user-tag (u, t) sets are all distinct user-tag combinations in A since a user can use the same tag to annotate multiple items.

3.2 Interpretation Scheme

The *boolean* scheme interprets the positive observed tagging data as 1 and denotes any other data as 0. Consequently, the user profile is only made from those positive entries exist in A. This scheme unfortunately overfits the negative entries and the null values as the same 0 value (Rendle et al., 2009).

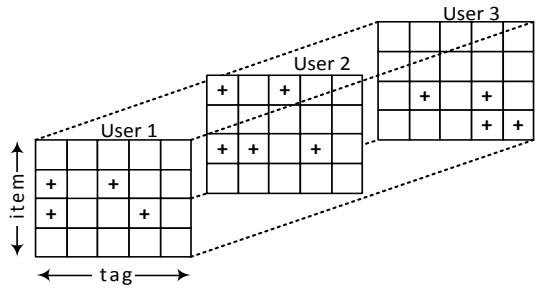
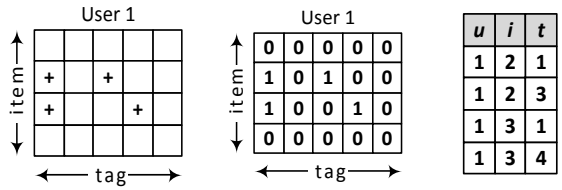


Figure 1: Toy example of observed tagging data with $U = \{u_1, u_2, u_3\}$, $I = \{i_1, i_2, i_3, i_4\}$, and $T = \{t_1, t_2, t_3, t_4, t_5\}$



(a) (b) (c)

Figure 2: The *boolean* interpretation scheme and its representation for User 1 (u_1). (a) Observed entries, (b) Data interpretation, (c) Data representation generated only from the positive entries.

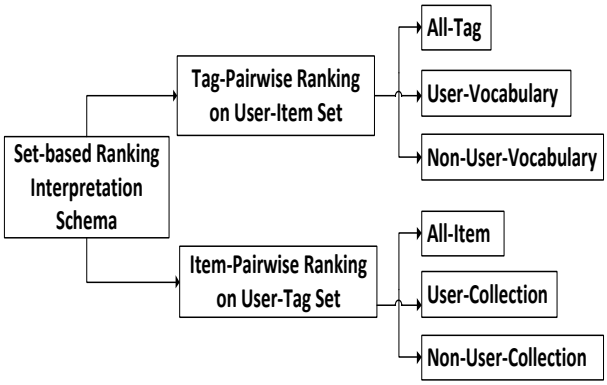


Figure 3: *Set-based* ranking interpretation schemes

Figure 2 shows an example of the scheme and its representation to build the user profile generated from the tagging data in Figure 1. For ease of illustration, it is only showing entries for User 1. Figure 1 shows that User 1 (u_1) has revealed his interest to i_2 and i_3 by annotating them using tags $\{t_1, t_3\}$ and $\{t_1, t_4\}$, respectively.

In contrast to the *boolean* scheme, the *set-based* ranking interpretation scheme distinguishes the positive, negative, and null values. It creates a pairwise classification ranking from the positive entries interpreted from the observed data and the negative entries interpreted from the non-observed data, on each user-item or user-tag set. In this case, the missing or null values are interpreted as the entries to be predicted for generating recommendations.

For representing the ranking within each set, the positive entries are simply given higher values than the negative ones. This indicates that the user favours the positive entries more than the negative entries (Rendle et al., 2009). The ranking order can be learned from A by creating: (1) tag-pairwise ranking on each distinct user-item set (u, i, t_P, t_N); or (2) item-pairwise ranking on each distinct user-tag set (u, t, i_P, i_N).

4 Proposed Tagging Data Interpretation and Its Representation

Given that users tagging behaviours in the different tagging systems are different from one another, the big question is how to best interpret the users tagging data as the user profile will be defined based on this representation. This sections details the six proposed tagging data interpretations and representations schemes, the recommendation method, as well as the process of selecting the best set of interpretation scheme to be used. The map of the interpretation schemes is represented in Figure 3 and the detailed examples are described in Figure 4.

4.1 Tag-Pairwise Ranking on User-Item Set

The tag-pairwise ranking on user-item (u, i) set is the ranking between the tags of positive entries (t_P) and the tags of negative entries (t_N), inferred from each $(u, i) \in A$. The tags of positive entries can easily be derived from the observed data.

Let us consider the toy example as shown in Figure 1. For User 1 (u_1), the tags of positive entries generated from the (u_1, i_1), (u_1, i_2), (u_1, i_3), and (u_1, i_4) sets are: $t_{P(u_1,i_1)} = \emptyset$, $t_{P(u_1,i_2)} = \{t_1, t_3\}$, $t_{P(u_1,i_3)} = \{t_1, t_4\}$, and $t_{P(u_1,i_4)} = \emptyset$, respectively. However, finding the tag values for the non-observed or negative entries is difficult.

We propose to infer and represent these tag values using the following schemes: (a) *all-tag*, (b) *user-vocabulary*, or (c) *non-user-vocabulary*. The ranking function can be formulated as $f(u, i, t_P, t_N) \rightarrow \mathbb{R}$.

4.1.1 All-Tag Pairwise Ranking

The *all-tag* ranking scheme interprets a negative entry (t_N) as follows. For each $(u, i) \in A$, user u is less favourable to annotate item i using any tags other than those appearing in positive entries $(T \backslash t_P)$ (Rendle and Schmidt-Thieme, 2010). The representation of this can be formulated as:

$$D = \{(u, i, t_P, t_N) : (u, i, t_P) \in A \wedge (u, i, t_N) \notin A \wedge t_N \in T \backslash t_P\}$$

In Figure 4, the positive entries show that u_1 has revealed his interest for i_2 by tagging the item using tags $\{t_1, t_3\}$. Given $T = \{t_1, t_2, t_3, t_4, t_5\}$, $t_{P(u_1,i_2)} = \{t_1, t_3\}$, and $T \backslash t_{P(u_1,i_2)} = \{t_2, t_4, t_5\}$, this tagging data is interpreted as u_1 favours $\{t_1, t_3\}$ more than $\{t_2, t_4, t_5\}$ to annotate i_2. The representation can then be generated from the pairwise ranking of $t_P = \{t_1, t_3\}$ and $t_N = \{t_2, t_4, t_5\}$ on (u_1, i_2) set.

4.1.2 User-Vocabulary Pairwise Ranking

The *user-vocabulary* ranking scheme interprets a negative entry (t_N) as follows. For each $(u, i) \in A$, user u is less favourable to annotate item i using any tags of user-vocabulary (V_u) than those appearing in positive entries $(V_u \backslash t_P)$. The representation of this can be formulated as:

$$D = \{(u, i, t_P, t_N) : (u, i, t_P) \in A \wedge ((u, i, t_N) \notin A \wedge t_N \in V_u \backslash t_P)\}$$

In Figure 4, the positive entries show that u_1 has revealed his interest for i_2 using tags $\{t_1, t_3\}$ and for i_3 using tags $\{t_1, t_4\}$. Knowing $V_{u_1} = \{t_1, t_3, t_4\}$, $t_{P(u_1,i_2)} = \{t_1, t_3\}$, and $V_{u_1} \backslash t_{P(u_1,i_2)} = \{t_4\}$, this tagging data is interpreted as u_1 favours $\{t_1, t_3\}$ more than $\{t_4\}$ to annotate i_2. The representation can then be generated from the pairwise ranking of $t_P = \{t_1, t_3\}$ and $t_N = \{t_4\}$ on (u_1, i_2) set.

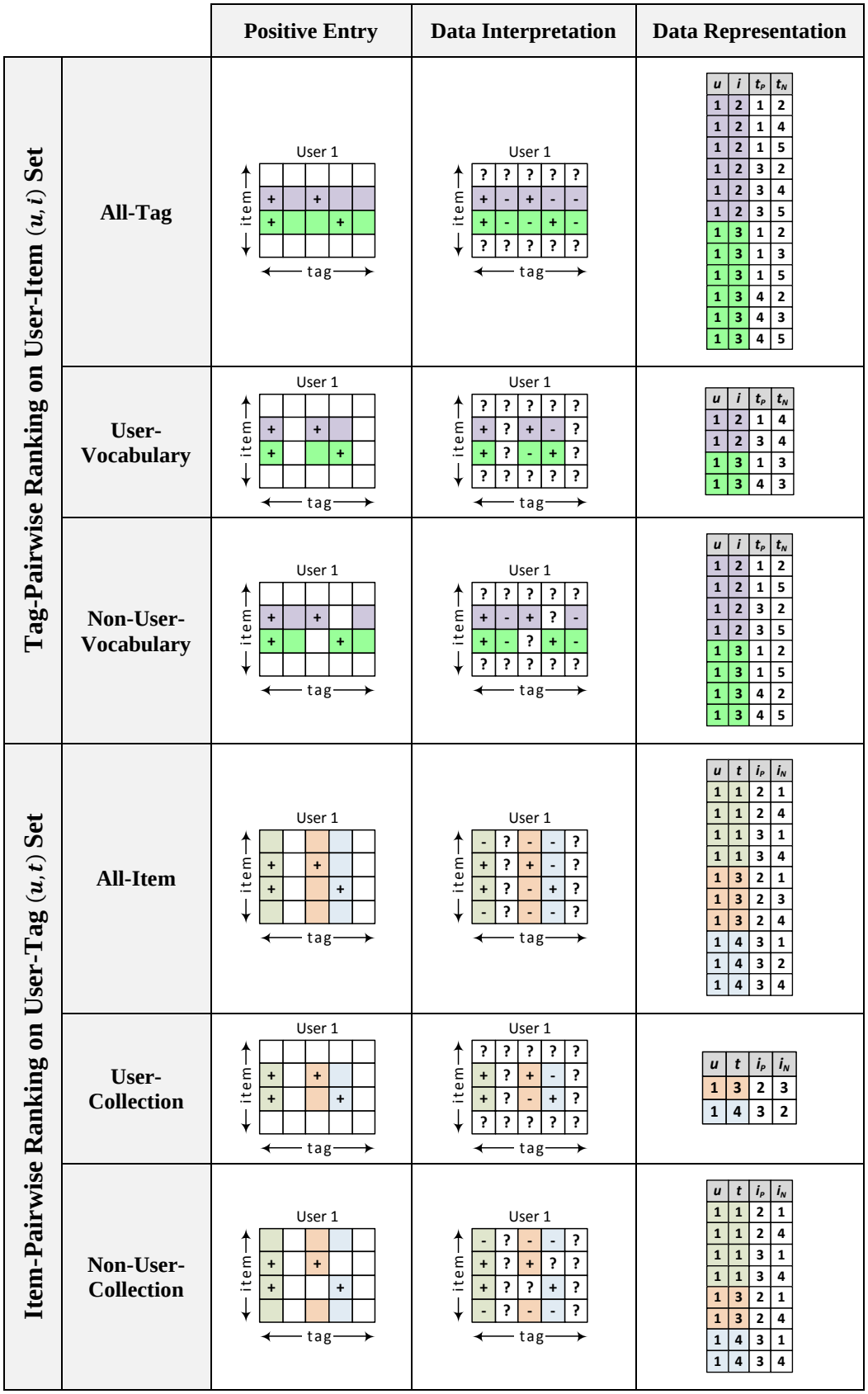


Figure 4: Examples showing the *set-based* ranking interpretation scheme and its various representation for u_1. The *set-based* ranking scheme uses the pairwise ranking of positive entries (interpreted from the observed data) and negative entries (interpreted from the non-observed data) for generating data representation. Other entries, that are to be predicted as recommendations, are noted as missing values (denoted as "?"). For the user-item set, the representation is generated by using tags as the pairwise ranking object (u, i, t_P, t_N) while the user-tag set generates the representation by using items as the pairwise ranking object (u, t, i_P, i_N).

4.1.3 Non-User-Vocabulary Pairwise Ranking

The *non-user-vocabulary* ranking scheme interprets a negative entry (t_N) as follows. For each $(u,i) \in A$, user u is less favourable to annotate item i using other tags that have not been used in any other items $(T \backslash V_u)$. The representation of this can be formulated as:

$$D = \{(u,i,t_P,t_N) : (u,i,t_P) \in A \wedge ((u,i,t_N) \notin A \wedge t_N \in T \backslash V_u)\}$$

In Figure 4, the positive entries show that u_1 has revealed his interest for i_2 using tags $\{t_1,t_3\}$ and for i_3 using tags $\{t_1,t_4\}$. Given $T = \{t_1,t_2,t_3,t_4,t_5\}$, $V_{u_1} = \{t_1,t_3,t_4\}$ and $T \backslash V_{u_1} = \{t_2,t_5\}$, this tagging data is interpreted as u_1 favours $\{t_1,t_3\}$ more than $\{t_2,t_5\}$ to annotate i_2. The representation can then be generated from the pairwise ranking of $t_P = \{t_1,t_3\}$ and $t_N = \{t_2,t_5\}$ on (u_1,i_2) set.

4.2 Item-Pairwise Ranking on User-Tag Set

The item-pairwise ranking on user-tag (u,t) set is the ranking between the items of positive entries (i_P) with the items of negative entries (i_N) inferred from each $(u,t) \in A$. The items of positive entries can easily be derived from the observed data.

Let us consider the toy example as shown in Figure 1. For User 1 (u_1), the items of positive entries generated from the (u_1,t_1), (u_1,t_2), (u_1,t_3), (u_1,t_4), and (u_1,t_5) sets are: $i_{P(u_1,t_1)} = \{i_2,i_3\}$, $i_{P(u_1,t_2)} = \emptyset$, $i_{P(u_1,t_3)} = \{i_2\}$, $i_{P(u_1,i_4)} = \{i_3\}$, and $t_{P(u_1,t_5)} = \emptyset$, respectively. However, finding the item values for the non-observed or negative entries is difficult.

The item values are inferred and represented using the following schemes: (a) *all-item*, (b) *user-collection*, or (c) *non-user-collection*. The ranking function can be formulated as $f(u,t,i_P,i_N) \rightarrow \mathbb{R}$.

4.2.1 All-Item Pairwise Ranking

The *all-item* ranking scheme interprets a negative entry (i_N) as follows. For each $(u,t) \in A$, user u is less favourable to use tag t to annotate any items other than those appearing in positive entries $(I \backslash i_P)$ (Gemmell et al., 2011). The representation of this interpretation can be formulated as:

$$D = \{(u,t,i_P,i_N) : (u,t,i_P) \in A \wedge (u,t,i_N) \notin A \wedge i_N \in I \backslash i_P\}$$

In Figure 4, the positive entries show that u_1 has used t_1 to reveal his interest for items $\{i_2,i_3\}$. Given $I = \{i_1,i_2,i_3,i_4\}$ and $i_{P(u_1,t_1)} = \{i_2,i_3\}$ so that $I \backslash i_{P(u_1,t_1)} = \{i_1,i_4\}$, this tagging data is interpreted as u_1 favours $\{i_2,i_3\}$ more than $\{i_1,i_4\}$ to be annotated using t_1. The representation can then be generated from the pairwise ranking of $i_P = \{i_2,i_3\}$ and $i_N = \{i_1,i_4\}$ on (u_1,t_1) set.

4.2.2 User-Collection Pairwise Ranking

The *user-collection* ranking scheme interprets a negative entry (i_N) as follows. For each $(u,t) \in A$, user u is less favourable to use tag t to annotate any items of user-collection (C_u) than those appearing in positive entries $(C_u \backslash i_P)$. The representation of this interpretation can be formulated as:

$$D = \{(u,i_P,i_N,t) : (u,i_P,t) \in A \wedge ((u,i_N,t) \notin A \wedge i_N \in C_u \backslash i_P)\}$$

In Figure 4, the positive entries show that u_1 has used t_1 to reveal his interest for items $\{i_2,i_3\}$, t_3 for $\{i_2\}$, and t_4 for $\{i_3\}$. Knowing $C_{u_1} = \{i_2,i_3\}$ and $i_{P(u_1,t_1)} = \{i_2,i_3\}$ so that $C_{u_1} \backslash i_{P(u_1,t_1)} = \emptyset$, this tagging data is interpreted as u_1 has no other favours than $\{i_2,i_3\}$ to be annotated using t_1. The representation then cannot be generated on (u_1,t_1) set. On the other hand, a representation can be generated on (u_1,t_3) set as u_1 has only used t_3 to annotate $\{i_2\}$. The tagging data is interpreted as u_1 favours $\{i_2\}$ more than $\{i_3\}$ to be annotated using t_3. The representation of this is the pairwise ranking of $i_P = \{i_2\}$ and $i_N = \{i_3\}$.

4.2.3 Non-User-Collection Pairwise Ranking

The *non-user-collection* ranking scheme interprets a negative entry (i_N) as follows. For each $(u,t) \in A$, user u is less favourable to use tag t to annotate other items that have not been tagged by u with other tags $(I \backslash C_u)$. The representation of this can be formulated as:

$$D = \{(u,i_P,i_N,t) : (u,i_P,t) \in A \wedge ((u,i_N,t) \notin A \wedge i_N \in I \backslash C_u)\}$$

In Figure 4, the positive entries show that u_1 has used t_1 to reveal his interest for items $\{i_2,i_3\}$, t_3 for $\{i_2\}$, and t_4 for $\{i_3\}$. Given $I = \{i_1,i_2,i_3,i_4\}$ and $C_{u_1} = \{i_2,i_3\}$ so that $I \backslash C_{u_1} = \{i_1,i_4\}$, this tagging data is interpreted as u_1 favours $\{i_2,i_3\}$ more than $\{i_1,i_4\}$ to be annotated using t_1. The representation can then be generated from the pairwise ranking of $i_P = \{i_2,i_3\}$ and $i_N = \{i_1,i_4\}$ on (u_1,t_1) set.

4.3 The Pairwise Ranking Method for Item Recommendation

The Pairwise Interaction Tensor Factorization (PITF) (Rendle and Schmidt-Thieme, 2010) is a well-known tag recommendation method. In this paper, we utilise this method for the task of item recommendation. The adaptation is necessarily as the task of recommending tags differs from the task of recommending items.

For tag recommendation, predictions are generated for each predefined user and item combination, i.e. the recommendation system predicts tags for an item to a user. However, for item recommendation, the recommendation system predicts items based on the user information only. Consequently, a method must calculate the item ranking score from the whole available tags before deciding which items are in the top-N recommendation list for the user. The original method works only for a (u,i) set scheme, however, our proposed method, called as PITF-I, is able to generate the recommendations by implementing the two sets: (u,i) and (u,t) schemes.

Using the (u,i) set interpretation scheme, PITF-I method represents the ranking of tagging data as (u,i,t_P,t_N), where (u,i,t_P) is a triple of positive entry and (u,i,t_N) is a triple of negative entry. It then creates a

tensor factorization model which employs an iterative gradient descent algorithm for optimizing the ranking function so that the positive entries are assigned with higher values than the negative entries. This ensures the notion that the user favours the positive entries more than the negative ones. The model is formulated as:

$$\hat{\mathcal{Y}} \approx \hat{U}_k \cdot \hat{T}_k^U + \hat{I}_k \cdot \hat{T}_k^I \quad (1)$$

where $\hat{U}_k$ is the user factor matrix, $\hat{I}_k$ is the item factor matrix, $\hat{T}_k^U$ is the tag factor matrix with respect to users, $\hat{T}_k^I$ is the tag factor matrix with respect to items, k is the size of factors, and $\hat{\mathcal{Y}}$ is the reconstructed personalized tag-ranking tensor. The element-wise relevance recommendation ranking score is calculated as follows:

$$\hat{y}_{u,i,t} \approx \sum_{j=1}^{k} \hat{u}_{u,j} \cdot \hat{t}_{t,j}^U + \sum_{j=1}^{k} \hat{\imath}_{i,j} \cdot \hat{t}_{t,j}^I \quad (2)$$

Using the (u,t) set interpretation, PITF-I exchanges the roles of items and tags with respect to each other. The data representation is the ranking of tagging data as (u,t,i_P,i_N), where (u,t,i_P) is a triple of positive entry and (u,t,i_N) is a triple of negative entry. Consequently, the model formulation becomes:

$$\hat{\mathcal{Y}} \approx \hat{U}_k \cdot \hat{I}_k^U + \hat{T}_k \cdot \hat{I}_k^T \quad (3)$$

where $\hat{U}_k$ is the user factor matrix, $\hat{T}_k$ is the tag factor matrix, $\hat{I}_k^U$ is the item factor matrix with respect to users, $\hat{I}_k^T$ is the item factor matrix with respect to tags, k is the size of factors, and $\hat{\mathcal{Y}}$ is the new tensor. The relevance recommendation ranking score is calculated as:

$$\hat{y}_{u,t,i} \approx \sum_{j=1}^{k} \hat{u}_{u,j} \cdot \hat{\imath}_{i,j}^U + \sum_{j=1}^{k} \hat{t}_{i,j} \cdot \hat{\imath}_{i,j}^T \quad (4)$$

4.4 Selecting the Set

Selecting a set for user profile representation cannot trivially be restricted based upon the recommendation task, i.e. based on the predefined notion of using the (u,i) set scheme for tag recommendation and using the (u,t) set scheme for item recommendation. This limitation has shown to cause the recommendation methods to perform at a varied level when they are implemented on various datasets (Gemmell et al., 2011; Ifada and Nayak, 2014; Rendle et al., 2009). A recommendation method performs best to its capacity when it is applied on the social tagging system for which it was built for. The performance degrades when it is applied to another social tagging system in which the user tagging behaviour varies. For example, the user tagging behaviour on the Delicious website (http://delicious.com) differs from the user behaviour on the LastFM website (http://www.last.fm/).

The difference can easily be perceived from the statistic of tagging data listed in Section 5.1. As the number of unique tags is much larger than the number of unique items in the Delicious data, the average number of unique tags used by each user to annotate an item overrides the number of unique items to be annotated using a tag. This analysis suggests that the (u,t) set in Delicious is more dominant than the (u,i) set. On the contrary, the number of unique tags is much less than the number of unique items in the LastFM data. This shows that the (u,t) set is less dominant than the (u,i) set as the average number of unique tags used by each user to annotate an item is less than the number of unique items to be annotated using a tag. The comparison between these two datasets (including other two datasets) is illustrated in Figure 5.

The characteristic of user tagging behaviour can be assessed by comparing the number of (u,i) and (u,t) sets in the data. It can be said that the dominant feature set (reflected by the larger distribution) is representative of the user tagging behaviour in that system and, therefore it can determine the interpretation scheme for building user profiles. The (u,i) set is more dominant than the (u,t) set when the users tend to use less number of tags in annotating items. The (u,t) set is more dominant than the (u,i) set when the users tend to use more number of tags in annotating items. We conjecture that, for Delicious data, the best performance of recommendation methods can be obtained when the (u,t) set interpretation scheme is implemented to build the user profile model. However, the (u,i) set interpretation scheme will become the best choice for modelling the user profile of LastFM data.

5 Empirical Analysis

Experiments are conducted to investigate the tagging data interpretation and representation schemes that can improve the performance of tag-based item recommendation systems by highlighting the representative features of user tagging behaviours on different datasets. We implemented six tagging data interpretations which resulted in different representations. The representation of each set is generated as pairwise ranking between the positive entries generated from the observed tagging data and the negative entries generated from the non-observed data using the following schemes: (a) *all-tag*, (b) *user-vocabulary*, (c) *non-user-vocabulary*, (d) *all-item*, (e) *user-collection*, and (f) *non-user-collection*. The first three schemes are based on the (u,i) set interpretation scheme, while the last three are based on the (u,t) set interpretation scheme.

5.1 Dataset

The offline experiments use several real-world tagging datasets to implement the proposed user profile representations for generating item recommendation. Adapting the standard practice of eliminating noise and decreasing the data sparsity (Nanopoulos, 2011; Rafailidis and Daras, 2013; Symeonidis, Nanopoulos and Manolopoulos, 2010), the datasets are refined by using the p-core technique (Batagelj and Zaveršnik, 2002), i.e. selecting users, items, and tags that have occurred in at least p number posts. Post is the set of distinct user-item combinations in the observed tagging data. In general we choose $p = 10$ to refine the dataset as this core value has shown a stable recommendation performance in our previous work after a systematic and extensive experiments (Ifada and Nayak, 2014). However, we use $p = 5$ instead for CiteULike and MovieLens datasets as they do not contain a 10-core.

The details of four tagging datasets used in this paper are:

Delicious (http://delicious.com/)**.** It is a website that facilitates its users to save, organize and discover interesting links on the web. The Delicious dataset is generated with 50,991 observed tagging data resulted from 10–core refinement, and consists of 2,009 users, 1,485 items and 2,589 tags.

LastFM (http://www.last.fm/)**.** It is a website that gives user personalized recommendations based on the music the user listens to. The LastFM dataset is generated with 99,211 observed tagging data resulted from 10-core refinement, and consists of 867 users, 1,715 items and 1,423 tags.

CiteULike (http://www.citeulike.org/)**.** It is a website that provides a service for managing and discovering scholarly references. The CiteULike dataset is generated with 59,832 observed tagging data resulted from 5-core refinement, and consists of 2,536 users, 3,091 items and 6,949 tags.

MovieLens (http://movielens.org)**.** It is a website which provides a personalized movie recommendation. The MovieLens dataset is generated with 25,103 observed tagging data resulted from 5-core refinement, and consists of 571 users, 1,684 items and 1,559 tags.

Figure 5 shows the set-based statistics to observe the characteristic of user tagging behaviour on each dataset. This information is used for selecting a profile presentation, i.e. based on the dominant number of sets.

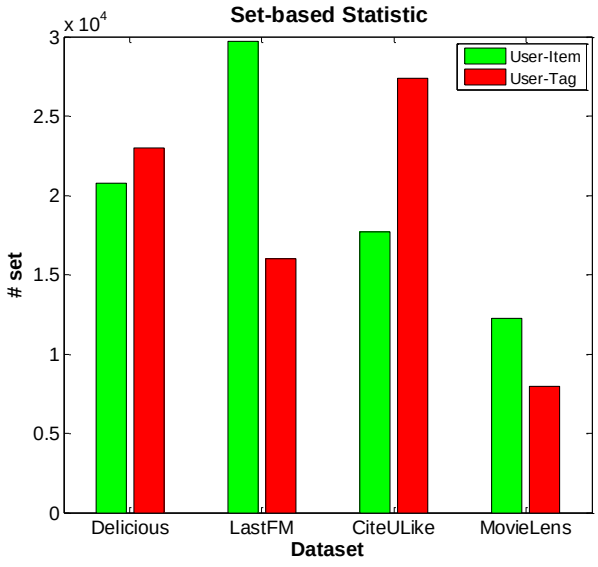


Figure 5: Set-based Statistic on Each Dataset

5.2 Evaluation Criteria

To evaluate the quality of recommendation, we implemented the 3-fold cross-validation and we divided the dataset randomly into a training set D_{train} (80%) and a test set D_{test} (20%) based on the number of posts data. D_{train} and D_{test} do not overlap in posts, i.e., there exist no triplets for a user-item combination in the training set if a triplet $(u, i, *)$ is present in the test set. The recommendation task is to predict and rank the Top-N items for the users present in D_{test}.

The performance is measured using F1-Score. F1-Score is a harmonic mean of overall precision and recall. Precision is the ratio of number of relevant items (all items in the post by the user in D_{test}) in the Top-N list to the total number of Top-N recommended items. Recall is the ratio of the number of relevant items in the Top- N list to the total number of relevant items.

$$Precision(D_{test}, N) = avg_{(u,i)\in D_{test}} \frac{|Test_u \cap TopN_u|}{|TopN_u|} \quad (1)$$

$$Recall(D_{test}, N) = avg_{(u,i)\in D_{test}} \frac{|Test_u \cap TopN_u|}{|Test_u|} \quad (2)$$

$$F1(D_{test}, N) = \frac{2 \cdot Precision(D_{test},N) \cdot Recall(D_{test},N)}{Precision(D_{test},N) + Recall(D_{test},N)} \quad (3)$$

Where $Test_u$ is the set of items tagged by target user in the D_{test} and $TopN_u$ is the Top-N list of items recommended to user from the reconstructed tensor $\hat{\mathcal{Y}}$.

5.3 Result and Discussion

5.3.1 Effect of Interpretation Set Selection on Recommendation Accuracy

The characteristic of users tagging behaviours of each dataset can be identified from the set-based statistic, as illustrated in Figure 5. The statistic shows that users of Delicious dataset have the same tagging behaviour as those of CiteULike, i.e. the (u, i) set is less dominant than the (u, t) sets. Conversely, the users of LastFM and MovieLens have similar tagging behaviour, i.e. the (u, i) set is more dominant than the (u, t) sets. We conjecture that the dominant set on each dataset defines which set should be used for interpreting the tagging data. This means that Delicious and CiteULike datasets should be best interpreted using the (u, t) set interpretation scheme, while LastFM and MovieLens datasets should be best interpreted using the (u, i) set scheme.

Figure 6 displays the F1-Score comparison on Top-N lists for the recommendation accuracy on each dataset which ascertains our claim that both Delicious and CiteULike achieve their best recommendation performance when their tagging data is interpreted using the *non-user-collection* of (u, t) set pairwise ranking interpretation scheme. Similarly, LastFM and MovieLens perform best when their tagging data is interpreted using the *non-user-vocabulary* of (u, i) set pairwise ranking interpretation scheme.

These results also verify that, for generating the best pairwise ranking representation, the negative entries of non-observed data should be interpreted from: (1) the tags that have not been used in any other items by user u for annotating item i on each (u, i) set, and (2) the items that that have not been tagged with any other tags by user u using tag t on each (u, t) set. Figure 6 also illustrates that the *all-tag* scheme which uses any tags other than those appearing in positive entries to annotate item i (Rendle and Schmidt-Thieme, 2010) or the *all-item* scheme which uses any items other than those appearing in positive entries using tag t (Gemmell et al., 2011) cannot interpret

non-observed data as negative entries properly and results in inferior recommendation performance.

On the other hand, though it sounds reasonable that, for the (u, i) set scheme, i.e. using the *user-vocabulary* interpretation, the negative entries should be interpreted from the tags of user-vocabulary other than those appearing in positive entries to annotate item i, however our experiments show that this interpretation severely impacts the recommendation performance for all datasets. Likewise, the items of user-vocabulary other than those appearing in positive entries using tag t should not be interpreted as the negative entries for the (u, t) set scheme, i.e. using the *user-collection* interpretation.

5.3.2 Data Representation Density

We examine the data representation density resulting from the six proposed interpretation schemes and compare with that of the *boolean* scheme as listed in Table 1. This observation is not merely to show that the *set-based* interpretation scheme is able to generate more dense data representation than the *boolean* scheme as this conclusion has been preliminary discovered (Rendle et al., 2009). Our focus is mainly to explore the representation density resulted from various *set-based* interpretation schemes, particularly comparison of the *non-user-vocabulary* and *non-user-collection* schemes to their counterparts, i.e. *all-tag* and *all-item*, respectively. Accordingly, when we highlight the superiority of the *non-user-vocabulary* scheme over *all-tag* scheme, it means that we can also state the same thing about that of the *non-user-collection* scheme over the *all-item* scheme.

Results in Table 1 show that the data representation densities of *user-vocabulary* and *user-collection* schemes are significantly lower than the other *set-based* schemes. It indicates that these two schemes, in comparison to others, are not able to interpret the tagging data efficiently and therefore the generated data representations are not able to include all relationships properly.

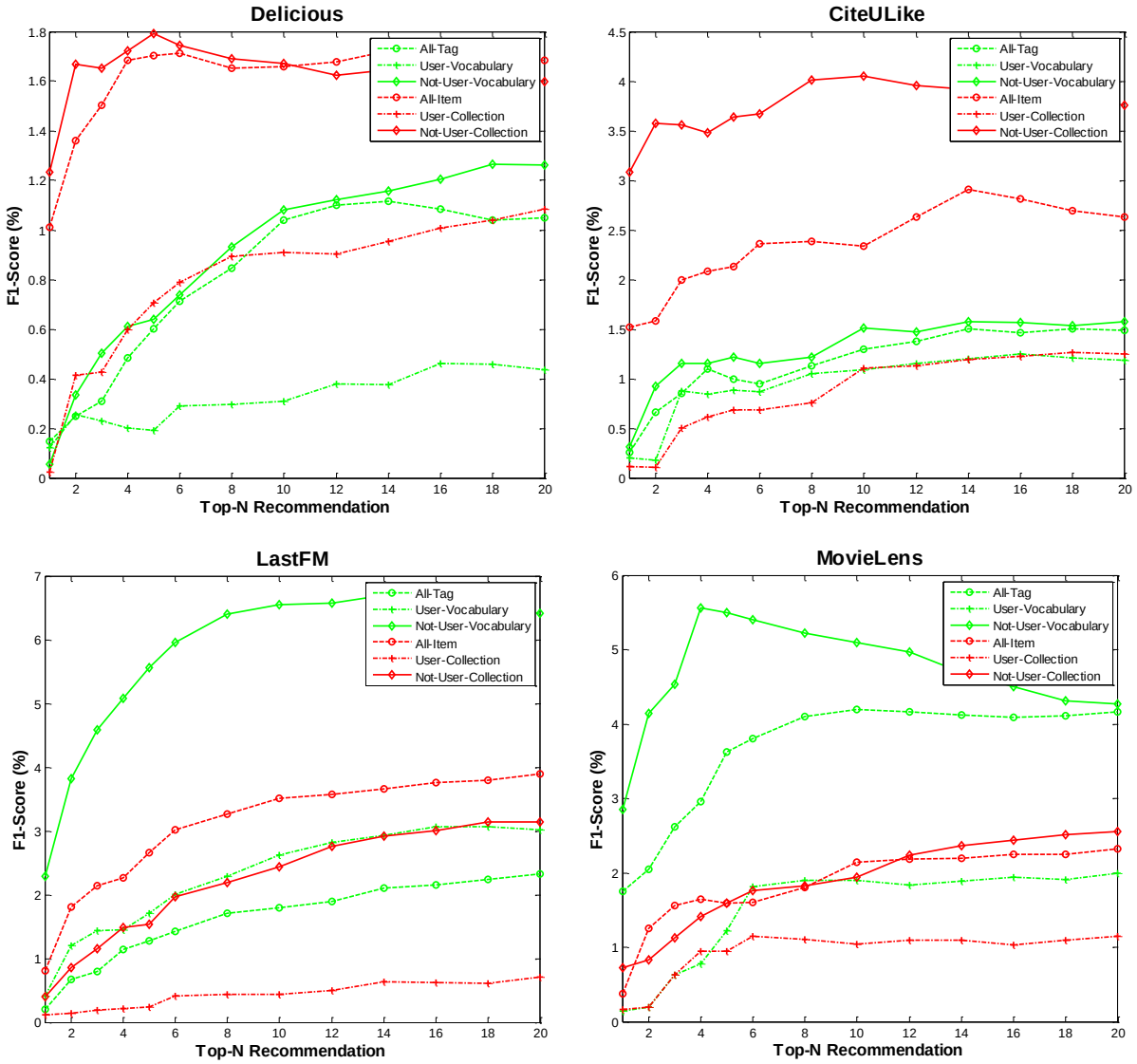


Figure 6: F1-Score comparison on Top-*N* lists for the recommendation accuracy resulting from the six proposed *set-based* interpretation schemes on various datasets

INTERPRE-TATION SCHEME \ DATASET		Delicious	LastFM	CiteULike	MovieLens
boolean		0.0006%	0.0042%	0.0001%	0.0015%
Tag-Pairwise Ranking on User-Item (u, i) Set	**All-Tag**	1.4980%	6.0184%	0.6902%	2.2693%
	User-Vocabulary	0.0273%	0.1262%	0.0099%	0.1626%
	Non-User-Vocabulary	***1.4707%***	***5.8922%***	***0.6803%***	***2.1067%***
Item-Pairwise Ranking on User-Tag (u, t) Set	**All-Item**	0.8540%	7.2087%	0.3064%	2.4224%
	User-Collection	0.0241%	0.3641%	0.0069%	0.2489%
	Non-User-Collection	***0.8299%***	***6.8447%***	***0.2996%***	***2.1735%***

Table 1: The Comparison of data representation density resulting from the *boolean* and *set-based* ranking interpretation schemes on various datasets

Table 1 shows that the data representation density of *non-user-vocabulary* is less than that of *all-tag* scheme. Yet, as established in Section 5.3.1, the former scheme outperforms the later. This fact clarifies that the *all-tag* scheme includes some relationships that are not meant to be. On each (u, i) set, the scheme uses all tags other than those appearing in positive entries to annotate item i as negative entries for the pairwise ranking representation. This interpretation is incorrect as some of those tags have actually been used by user u to annotate other items which means that those used tags should not be interpreted as negative entries. It confirms that the interpretation scheme strongly determines the recommendation performance. To generate quality recommendations, an efficient user profile presentation is more important, instead of just simply trying to get more dense data representation as other researchers had done previously (Cui et al., 2011; Leginus, Dolog and Žemaitis, 2012; Rafailidis and Daras, 2013). These methods practically implement the *boolean* scheme to interpret the tagging data and then applied clustering techniques for reducing the tag dimension to represent the semantically similar tags. These approaches are able to generate more dense data, yet they still interpret the tagging data improperly.

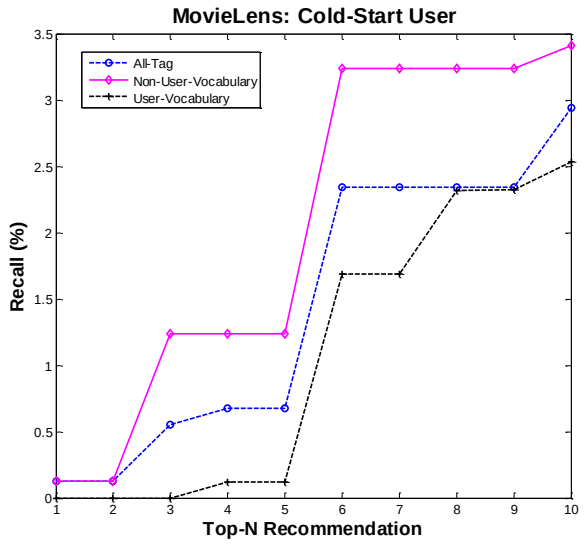


Figure 7: Cold-Start User Recall Recommendation on *all-tag*, *non-user-vocabulary*, and *user-vocabulary*

5.3.3 Cold-start User Problem

We carry out further experiments to examine the impact of an interpretation scheme to recommendation performance in addressing the cold-start user problem. We identify the cold-start user problem as a situation in which a user has annotated a single item only with limited number of tags. Due to limited usage data, we cannot infer the user preferences on the system.

We compare the recommendation performance on MovieLens dataset by implementing the (u, i) set interpretation scheme as it is the best choice of scheme to model the user profile. Since our main focus is on how the interpretation scheme is able to address the cold-start user problem, we use the recall metric to demonstrate the coverage of recommendations. Figure 7 shows the recall of recommendations generated using the *non-user-vocabulary* scheme outperforms the results of *all-tag*. As expected, the results of the *user-vocabulary* scheme have shown the worst performance. This fact confirms that the approach of interpreting negative entries from tags that have not been used by user u in any other items is resulting quality recommendations for both the active and the cold-start users. We can also state the same thing for the (u, t) set interpretation scheme, i.e. the negative entries are best interpreted from items that have not been tagged by user u using any other tags.

6 Conclusion

In this paper, we proposed six *set-based* tagging data interpretation schemes and representations to investigate an efficient scheme leading to build effective user profiles for generating item recommendations. For each interpretation scheme, a data representation is produced, where for each set, pairwise ranking is generated between the positive entries of observed tagging data and the negative entries of non-observed data. We have shown that the set to be used as the interpretation scheme must be selected based on the dominant number of set observed from the tagging data as it represents the unique characteristic of user tagging behaviour in the system. We implemented the PITF-I method, using a pairwise ranking representation between the positive entries and the negative entries. The PITF-I method represents the ranking of tagging data as (u, i, t_P, t_N) when the dataset has a dominant number of (u, i) sets. Alternatively, the

method represents the ranking of tagging data as (u, t, i_P, i_N) when the dataset has a dominant number of (u, t) sets.

The proposed representations are extensively evaluated on four datasets which exhibit different tagging behaviour characteristics. Empirical analysis shows that the improper interpretation, of negative entries to be ranked pairwise with the positive entries, results in inferior recommendation performance. The best scheme for pairwise ranking representation should generate the negative entries interpreted from either the tags or the items that have not been used by user u in any other items or tags, i.e. *non-user-vocabulary* or *non-user-collection* schemes, respectively. We also show how this scheme is able to address the cold-start user problem. In the future, we are planning to investigate the tagging data interpretation and representation schemes for a list-wise ranking tensor factorization method.

7 Acknowledgement

This work is supported by the Directorate General of Higher Education (DGHE) Indonesia. Computational resources and services are provided by the HPC and Research Support Group, Queensland University of Technology, Brisbane, Australia.

8 Reference

Adomavicius, G., and Tuzhilin, A. (2005). Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering, 17*(6), 734-749.

Batagelj, V., and Zaveršnik, M. (2002). Generalized Cores. *arXiv preprint cs/0202039*,

Cui, J., Liu, H., He, J., Li, P., Du, X., and Wang, P. (2011). TagClus: a random walk-based method for tag clustering. *Knowledge and information systems, 27*(2), 193-225.

Gemmell, J., Schimoler, T., Mobasher, B., and Burke, R. (2011). Tag-based Resource Recommendation in Social Annotation Applications. In *User Modeling, Adaption and Personalization*. 111-122, Springer.

Ifada, N., and Nayak, R. (2014). A Two-Stage Item Recommendation Method Using Probabilistic Ranking with Reconstructed Tensor Model. In *User Modeling, Adaptation, and Personalization*. 98-110, Springer.

Kim, H.-N., Ji, A.-T., Ha, I., and Jo, G.-S. (2010). Collaborative Filtering based on Collaborative Tagging for Enhancing the Quality of Recommendation. *Electronic Commerce Research and Applications, 9*(1), 73-83.

Kolda, T., and Bader, B. (2009). Tensor Decompositions and Applications. *SIAM Review, 51*(3), 455-500.

Leginus, M., Dolog, P., and Žemaitis, V. (2012). Improving Tensor Based Recommenders with Clustering. In *User Modeling, Adaptation, and Personalization*. 151-163, Springer Berlin Heidelberg.

Nanopoulos, A. (2011). Item Recommendation in Collaborative Tagging Systems. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 41*(4), 760-771.

Rafailidis, D., and Daras, P. (2013). The TFC Model: Tensor Factorization and Tag Clustering for Item Recommendation in Social Tagging Systems. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 43*(3), 673-688.

Rendle, S., Balby Marinho, L., Nanopoulos, A., and Schmidt-Thieme, L. (2009). Learning optimal ranking with tensor factorization for tag recommendation. *Proc. The 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Paris, France, 727-736. ACM.

Rendle, S., and Schmidt-Thieme, L. (2010). Pairwise Interaction Tensor Factorization for Personalized Tag Recommendation. *Proc. The 3rd ACM International Conference on Web Search and Data Mining*, New York, USA, 81-90. ACM.

Symeonidis, P., Nanopoulos, A., and Manolopoulos, Y. (2008). Tag Recommendations based on Tensor Dimensionality Reduction. *Proc. The 2008 ACM Conference on Recommender Systems*, Lausanne, Switzerland, 43-50. ACM.

Symeonidis, P., Nanopoulos, A., and Manolopoulos, Y. (2010). A Unified Framework for Providing Recommendations in Social Tagging Systems Based on Ternary Semantic Analysis. *IEEE Transactions on Knowledge and Data Engineering, 22*(2), 179-192.

Zhang, Z.-K., Zhou, T., and Zhang, Y.-C. (2011). Tag-Aware Recommender Systems: A State-of-the-Art Survey. *Journal of Computer Science and Technology, 26*(5), 767-777.

A Multidimensional Collaborative Filtering Fusion Approach with Dimensionality Reduction

Xiaoyu Tang, Yue Xu, Ahmad Abdel-Hafez, Shlomo Geva
Science and Engineering Faculty,
Queensland University of Technology,
Brisbane, Australia

xiaoyu.tang@connect.qut.edu.au, {yue.xu, a.abdelhafez, s.geva}@qut.edu.au

Abstract

Multidimensional data are getting increasing attention from researchers for creating better recommender systems in recent years. Additional metadata provides algorithms with more details for better understanding the interaction between users and items. While neighbourhood-based Collaborative Filtering (CF) approaches and latent factor models tackle this task in various ways effectively, they only utilize different partial structures of data. In this paper, we seek to delve into different types of relations in data and to understand the interaction between users and items more holistically. We propose a generic multidimensional CF fusion approach for top-N item recommendations. The proposed approach is capable of incorporating not only localized relations of user-user and item-item but also latent interaction between all dimensions of the data. Experimental results show significant improvements by the proposed approach in terms of recommendation accuracy.

Keywords: multidimensional data, neighbourhood, dimensionality reduction, collaborative filtering, recommender systems.

1 Introduction

In recent years, the development of Web 2.0 techniques and various smart devices have created new opportunities for recommender systems, by revealing more information additional to user-item transactions. For example, Social Tagging Systems (STS) encourage users to employ user-defined keywords to help manage content in a personalized way. Recommender systems built upon STS (Tso-Sutter et al. 2008) utilize social tagging to improve recommendation mechanisms. Context-Aware Recommender Systems (CARS) (Adomavicius and Tuzhilin 2011, Karatzoglou et al. 2010) incorporate context information (e.g. time, location, weather, etc.) into recommendation models to predict new relations more accurately. Tags and contextual information can be treated as additional dimensions to user-item matrix. Thus, the data used by these recommender systems share the property that each user-item transaction involves multiple entities other than merely a user and an item.

The top-N item recommendation task for multidimensional data has been tackled in many different ways. For the neighbourhood-based Collaborative Filtering (CF) approaches, researchers have presented various ways to utilize multidimensional data in user/item profiling and in neighbourhood formation through explicit conversion of dimensions (Marinho et al. 2012, Tso-Sutter et al. 2008, Liang et al. 2010). For example, Liang proposed to construct user profiles by using tags so as to utilize the multiple relationships among users, items and tags for extracting the semantic meaning of each tag for users (Liang et al. 2010). However, these approaches mostly work in ad hoc ways which leads to that they cannot be directly applied to data with more dimensions. Moreover, they cannot take into account the latent relations in data through merely explicit relations from neighbourhood. Differently, some recent Tensor Factorization (TF) based models (Symeonidis et al. 2010, Karatzoglou et al. 2010, Rendle et al. 2009) model multidimensional data as tensors (i.e. multidimensional arrays) and are able to discover holistic latent relationships in data. However, pure TF-based recommendation models lack the ability to utilize localized relationships which are often the privilege of neighbourhood-based CF approaches. Furthermore, the increase of the dimensionality of data can cause serious efficiency problem for the factorization process, which largely restrict the application in practice.

Despite various recommendation models have been proposed in the categories of neighbourhood-based approaches and factorization models, they essentially only deal with parts of relations existing in data. Neighbourhood-based approaches work with user-user or item-item neighbourhood relations, while TF utilizes the global latent interaction between different dimensions. There has not been any research which simultaneously incorporates all these different types of relations in multidimensional data for making recommendations. This is the objective of this paper.

In this paper, we propose to profile users and items through conducting dimensionality reduction on multidimensional data, and we present a novel generic Multidimensional Collaborative Filtering Fusion (MCFF) approach for top-N item recommendation using multidimensional data. Three different levels of structures of data can be captured and utilized simultaneously by the proposed recommendation model. Our approach first transforms data to model user and item profiles by means of observing data from the user and item dimensions respectively. Then, dimensionality reduction is conducted

on transformed data for removing noises and revealing implicit relations between all dimensions. Finally, the proposed approach captures the refined localized user-user and item-item relations and also global latent relations between all dimensions, to generate item recommendations.

The contributions of our work are as follows:

- Our profiling method models users and items based on holistic relations in the entire data, and it is directly generalizable to profiling for other entities or dimensions, and extendable to *N*-dimensional data. Compared to existing neighbourhood-based approaches for multidimensional data, our profiling approach can incorporate the multidimensional interaction between different dimensions into the profiles of users/items, and is able to remove noise and keep sound efficiency.
- The proposed multidimensional CF fusion recommendation approach takes advantages of not only the localized neighbourhood relations of users and items, but also holistic latent relations between all dimensions. This enables the recommendation algorithm to understand data more completely than pure TF-based CF models.

We have conducted extensive experiments to validate the effectiveness of the proposed multidimensional profiling method and to evaluate the performance of the proposed recommendation approach MCFF against some state-of-the-art multidimensional CF recommendation algorithms. The experimental results show that our approaches substantially improve the performance of top-N item recommendation in terms of precisions/recalls/F1 scores.

The rest of this paper is organized as follows: Section 2 summarizes the related work. In Section 3 we propose a multidimensional profiling approach for representing users and items. Based on that, we integrate the profiling method into neighbourhood-based CF approaches and propose a novel multidimensional CF recommendation model which fuses user and item neighbourhoods with implicit holistic interaction assimilated. Experimental results are given in Section 4, which shows superior performance of the proposed recommendation model. Finally, Section 5 concludes this paper.

2 Related Work

Traditionally, most of the CF recommender systems are categorized into two families: neighbourhood-based approaches and latent factor models (Adomavicius and Tuzhilin 2005). The neighborhood-based CF recommender systems are usually based on nearest neighborhood relations. Examples include user-based and item-based CF (Desrosiers and Karypis 2011). The latent factor models (Koren et al. 2009, Symeonidis et al. 2010) have received much attention due to its competitive performance in Netflix competition. The entities of data in these traditional CF recommender systems often include only users and items. This kind of data and the recommender systems are 2-dimensional, since each user-item transaction is only associated with two entities: user and item.

The development of information systems working with multidimensional data, such as social tagging systems and context-aware systems, have promoted the recommendation systems to incorporate data with more dimensions. Different categories of recommendation approaches have been proposed for multidimensional data scenario in recent years. Marinho et al. discussed how conventional CF can be applied for computing recommendations in multidimensional data environments through dimension projection (Marinho et al. 2012). They referred to this type of recommendation approaches as projection-based CF. The approaches which fall into this type usually project data between different dimensions in order to reduce the data spaces and predict new user-item relations. Tso-Sutter et al. proposed to extend the typical user-item matrix with tags which are taken as pseudo users and pseudo items (Tso-Sutter et al. 2008). Liang et al. proposed to construct tag-based user profiles using the multiple relationships among users, items and tags to find the semantic meaning of each tag for each user individually (Liang et al. 2010). Tagommenders (Sen et al. 2009) predicts users' preferences for items based on their inferred preferences for tags. They proposed to combine tag preference inference algorithms with tag-aware recommenders and showed empirically that their approach outperforms classic CF algorithms. Although at least three dimensions of data are considered in these approaches, they are not directly generalizable to more dimensions of information. Besides, most of these approaches do not have the ability to incorporate latent multidimensional relations in data for recommendation making. These disadvantages limit their recommendation capacity.

Differently, latent factor models enjoy the ability to discover latent relationships from a holistic perspective. For this category of CF models, a newly emerging stream of methods focusing on multidimensional data is tensor factorization. TF-based recommendation models formulate users, items and additional dimensions such as tags, as multidimensional matrices which are called tensors. Multiverse Recommendation (Karatzoglou et al. 2010) is a TF-based model for context-aware item recommendation which utilizes Tucker Decomposition (TD) for rating prediction task with the user-item-context N-dimensional tensor data. Time is used as the context in this method. Rendle et al. proposed a different approach for creating the initial tensor which expresses user-item-tag relations (Rendle et al. 2009). Instead of using the 0/1 interpretation scheme, they used a so-called Post-Based Ranking Interpretation (PBRI). Symeonidis et al. introduced a unified framework which provides three types of recommendations in STS, using a 3-order tensor to model the relations of users, items, and tags (Symeonidis et al. 2010). Multi-way latent semantic analysis is conducted using Higher-Order Singular Value Decomposition (HOSVD). They reported superior recommendation performance of their model for item recommendation compared to other approaches. To sum up, the TF-based CF models enjoy similar advantages of 2-dimensional latent factor models and are able to use more information from additional dimensions. However, although these approaches hold the holistic perspective of data with latent relationships discovered, they neglect the localized relations which usually are extracted by nearest

neighbourhood approaches. Additionally, in real-world implementations, some other drawbacks like low computing efficiency, curse of dimensionality or lengthy training time may become severe problems as the size and dimensions of the data increase, while neighbourhood-based CF usually performs much better when these concerns matter a lot.

As aforementioned, the extraction and utilization of global latent relations and explicit user's/item's localized relations are the core of most CF approaches to provide quality recommendations. However, no research has been done to incorporate all these three layers of relations for making item recommendations. Furthermore, no previous work has proposed a generalizable multidimensional method for user/item profiling and neighbourhood formation. We believe that a novel CF approach effectively utilizing multidimensional latent relations and localized explicit relations possesses an all-sided view of data relations and thus has the ability to provide recommendation of high accuracy, while still enjoy the desirable efficiency in practice. This is the focus of this paper.

3 Multidimensional Collaborative Filtering Fusion

In this paper, for the sake of simplicity, we will describe the proposed approaches with three dimensions: users, items and tags, as in the context of STS. In fact, tags can be replaced with other entities such as item features or categories. The profiling and recommendation approaches proposed in this section can be generalized to data with more dimensions. We define U, I and T as disjoint non-empty finite sets, whose elements are users, items and tags, respectively. In this way, the data is 3-dimensional.

3.1 Multidimensional User/Item Profiling

In this section, we propose a multidimensional profiling approach for users and items. In our approach, the 3-dimensional user-item-tag data is represented as a 3-order tensor $\mathcal{A} \in \mathbb{R}^{|U|\times|I|\times|T|}$, in which a tensor element is represented by a 3-tuple (u, i, t). In the simplest case, the value of (u, i, t) is defined as:

$$e_{u,i,t} = \begin{cases} 1, & \text{if the transaction } (u,i,t) \text{ exists} \\ 0, & \text{otherwise} \end{cases}$$

For social tagging, a transaction or tag assignment (u, i, t) exists if user u collected item i with tag t.

Generally, users' item preferences are represented by users' explicit ratings or implicit ratings. In the context of this paper, the item preference of a user u to an item i, denoted as $r_{u,i}$, is defined as $r_{u,i} = 1$ if u collected i with at least one tag, otherwise $r_{u,i} = 0$ indicating that the user's preference to this item is unknown.

Matricization, also known as unfolding or flattening, is the process of reordering the elements of an *N*-order tensor into a matrix (Kolda and Bader 2009, Acar and Yener 2009). Some decomposition techniques apply matricization to tensors for extracting and explaining data properties in order to understand the data structure. Illustration of a matricization operation for a 3-order tensor $\mathcal{A} \in \mathbb{R}^{|U|\times|I|\times|T|}$ is given in Figure 1. The three modes/dimensions of the tensor $\mathcal{A}$ are users (U), items (I) and tags (T). Figure 1 shows the U-mode unfolding of the tensor $\mathcal{A}$, denoted as $\mathcal{A}_{(U)} \in \mathbb{R}^{|U|\times|I||T|}$.

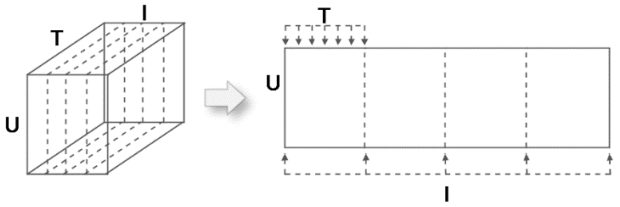


Figure 1: Matricization of a 3-order tensor

Formally, in the mode-n matricization of a 3-order tensor $\mathcal{A} \in \mathbb{R}^{I_1\times I_2\times I_3}$, a tensor element (i_1, i_2, i_3) maps to a matrix element (i_n, j) (Kolda and Bader 2009), where

$$j = 1 + \sum_{\substack{k=1 \\ k\neq n}}^{3} (i_k - 1) J_k \qquad (1)$$

and $J_k = \prod_{\substack{m=1 \\ m\neq n}}^{k-1} I_m$.

Inspired by the tensor matricization, we propose to represent users and items by matricizing the tensor $\mathcal{A} \in \mathbb{R}^{|U|\times|I|\times|T|}$ by U-mode and by I-mode. In this way, users are represented by vectors instead of matrices in which each user u is represented by a binary vector $\overrightarrow{u^e}$. Each element u_k^e in $\overrightarrow{u^e}$ corresponds to an item-tag pair (i, t), and $u_k^e = 1$ if $e_{u,i,t} = 1$, otherwise $u_k^e = 0$. Items' representations are similarly formed. The outcomes of the two matricization operations are two matrices: a matrix $\mathcal{A}_{(U)} \in \mathbb{R}^{|U|\times|I||T|}$ with U mapped to row vectors and a matrix $\mathcal{A}_{(I)} \in \mathbb{R}^{|I|\times|U||T|}$ with I mapped to row vectors. Hence, $\mathcal{A}_{(U)}$ can be represented as a vector $< \overrightarrow{u_1^e}, \overrightarrow{u_2^e}, \dots, \overrightarrow{u_{|U|}^e} >^{\intercal}$ and $\mathcal{A}_{(I)}$ can be represented as a vector $< \overrightarrow{\imath_1^e}, \overrightarrow{\imath_2^e}, \dots, \overrightarrow{\imath_{|I|}^e} >^{\intercal}$, where $\overrightarrow{u^e}$ and $\overrightarrow{\imath^e}$, which represent a user and an item respectively, are the following vectors:

$$\overrightarrow{u^e} = < e_{u,i_1,t_1}, e_{u,i_2,t_1}, \dots, e_{u,i_{|I|},t_{|T|}} >$$

$$\overrightarrow{\imath^e} = < e_{u_1,i,t_1}, e_{u_2,i,t_1}, \dots, e_{u_{|U|},i,t_{|T|}} >$$

Compared to the tag-aware CF fusion model (Tso-Sutter et al. 2008), the user and item profiles created by the matricization of tensors can essentially preserve the multidimensional semantic relations in the data. However, this also brings up new problems. First, matricization of tensors may lead to misinterpretation if the data are noisy (Acar and Yener 2009). Also, since usually the number of items and tags are quite large, tensor matricization could deteriorate the efficiency of neighborhood formation using the U-mode and I-mode unfolding matrices $\mathcal{A}_{(U)}$ and $\mathcal{A}_{(I)}$ as the profiles of users and items, respectively. In order to solve these problems, we propose to conduct SVD on $\mathcal{A}_{(U)}$ and $\mathcal{A}_{(I)}$ to discover the latent factors and to reduce the representation spaces.

We apply SVD on the matrix $\mathcal{A}_{(U)}$ and matrix $\mathcal{A}_{(I)}$ separately in the same way. Taking $\mathcal{A}_{(U)}$ as an example, through factorizing the matrix $\mathcal{A}_{(U)}$ via the SVD process, latent factors can be extracted and $\mathcal{A}_{(U)}$ can be represented as:

$$\mathcal{A}_{(U)} = \mathcal{U}_{|U|\times|U|} \cdot \mathcal{S}_{|U|\times|I||T|} \cdot \mathcal{V}^T_{|I||T|\times|I||T|} \qquad (2)$$

By preserving a certain amount of information in the data, i.e., specifying the number of factors to be retained as $k_u \leq |U|$, we can project the representations of users from the vector space $\mathbb{R}^{|I||T|}$ onto the latent factor space $\mathbb{R}^{k_u}$, so as to reduce the dimensions of user profile representations. The space projection operation is fulfilled by the following equation:

$$\mathcal{UF}_{|U|\times k_u} = \mathcal{U}_{|U|\times k_u} \cdot \mathcal{S}_{k_u \times k_u} \quad (3)$$

where $\mathcal{U}_{|U|\times k_u} \in \mathbb{R}^{|U|\times k_u}$ and $\mathcal{S}_{k_u\times k_u} \in \mathbb{R}^{k_u \times k_u}$ represent the truncated matrices of $\mathcal{U}_{|U|\times|U|}$and $\mathcal{S}_{|U|\times|I||T|}$ respectively, given the number of factors k_u. $\mathcal{UF}_{|U|\times k_u}$ is a matrix where each row vector represents a user's preference measurement in the new latent factor space.

With the reduced user representations, neighbourhood formation can proceed efficiently and accurately. We will discuss this in the next section.

Similar procedure can be defined to reduce item representations by applying SVD on the *I*-mode unfolding matrix $\mathcal{A}_{(I)}$ to generate a truncated matrix $\mathcal{IF}_{|I|\times k_i}$ with a given factor number k_i for the item space. The profiles of a user u and an item i in latent factor spaces are represented as follows:

$$\overrightarrow{u^f} =< f_1^u, f_2^u, \dots, f_{k_u}^u >$$

$$\overrightarrow{i^f} =< f_1^i, f_2^i, \dots, f_{k_i}^i >$$

where $\overrightarrow{u^f}$ and $\overrightarrow{i^f}$ are row vectors in $\mathcal{UF}_{|U|\times k_u}$ and $\mathcal{IF}_{|I|\times k_i}$, respectively, $1 \leq k_u \leq |U|$, $1 \leq k_i \leq |I|$. k_u and k_i are the given numbers of factors for decomposing $\mathcal{A}_{(U)}$ and $\mathcal{A}_{(I)}$ respectively.

The extension of the multidimensional profiling approaches proposed in this section to *N*-dimensional data is straightforward. For the mode-n matricization of an *N*-order tensor $\mathcal{A} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$, a tensor element $(i_1, i_2, \cdots, i_N)$ maps to a matrix element (i_n, j), where $j = 1 + \sum_{k=1, k\neq n}^{N}(i_k - 1)J_k$ with $J_k = \prod_{m=1, m\neq n}^{k-1} I_m$ (Kolda and Bader 2009).

In Section 3.2 and Section 3.3, we propose to integrate the multidimensional profiling methods into two neighbourhood-based CF approaches and further propose the MCFF approach based on this profiling method.

3.2 Multidimensional Neighbourhood-based Collaborative Filtering

In this section, we present a user-based CF algorithm integrated with the multidimensional user profiling approach proposed in Section 3.1. The item-based CF algorithm can be similarly integrated with the proposed multidimensional item profiling approach.

The standard user-based CF algorithm (Su and Khoshgoftaar 2009) works with the following procedure:

First, formulate user interests into user profiles for each user. For example, Tso-Sutter et al. proposed to extend the typical user-item matrix with tags which are taken as pseudo users and pseudo items (Tso-Sutter et al. 2008). Differently, user profiles in our approach are created by the multidimensional profiling method presented in Section 3.1.

Secondly, generate user neighbourhoods based on a predefined similarity measurement between any two users, such as Jaccard similarity or Cosine similarity. In our approach, since the user profiles are vectors consisting of real numbers, Cosine similarity is used and it is given in Equation (4):

$$sim(u_i, u_j) = Cosine(u_i, u_j) = \frac{\overrightarrow{u_i^f} \cdot \overrightarrow{u_j^f}}{\left\|\overrightarrow{u_i^f}\right\| \cdot \left\|\overrightarrow{u_j^f}\right\|} \quad (4)$$

Finally, for each target user, based on the item preferences of this user's neighbour users, compute a preference prediction for each new item and then produce a ranked list of top-N item recommendation. The preference prediction $P_{u,i}^{UCF}$ to a new item i for a target user u is given as:

$$P_{u,i}^{MUCF} = \sum_{v\in N_u, i\in I_v} \left(r_{v,i} \cdot sim(u, v)\right) \quad (5)$$

where N_u are the neighbour users of target user u. I_v is the set of items collected by user v. $r_{v,i}$ which is user v's item preference for item i is defined in Section 3.1.

Likewise, item-based CF with multidimensional item profiling can be formulated similarly:

$$P_{u,i}^{MICF} = \sum_{j\in I_u, i\in N_j} \left(r_{u,j} \cdot sim(i, j)\right) \quad (6)$$

where N_j are the neighbour items of a collected item j which are new to user u. I_u is the set of items collected by user u, and $sim(i, j)$ is the similarity between item i and item j.

Thereby, the two multidimensional neighbourhood-based CF approaches are proposed. They are able to use 3-dimensional user-item-tag data to profile users and items more accurately as stated in Section 3.1. In Section 4, we will empirically demonstrate the multidimensional neighbourhood-based CF approaches can show better recommendation performance than their standard counterparts.

3.3 Fusing User-based and Item-based CF for Multidimensional Item Recommendation

As an additional dimension of transaction data beyond users and items, tags can be seen as features specific to individual transactions, i.e., they are usually related to users and items at the same time. That is, tags (or additional features of other types) are local information for transactions. In this way, the relations in the multidimensional data seen from the aspects of users or items can be different. For example, a user collects the movie *Titanic* with the tag "*love*"; a different user collects the same movie with the tag "*disaster*". This indicates a recommendation model which can appropriately utilize localized neighbourhood relations from both user and item perspectives may lead to improvement of recommendation quality, which forms the basis of some previous works (Wang et al. 2006, Tso-Sutter et al. 2008, Bar et al. 2013, Lee and Olafsson 2009).

In Section 3.2, two neighborhood-based CF approaches with multidimensional user and item profiling have been proposed. A CF fusion approach can be used to unify the power of user neighborhoods and item neighborhoods together for recommendation. In this section, we propose a Multidimensional CF Fusion

(MCFF) approach which fuses the two neighbourhood relations in a way similar to the tag-aware CF fusion model (Tso-Sutter et al. 2008).

CF fusion for the top-N item recommendation task is done by combining the predictions of user-based and item-based CF approaches. In order to compare our MCFF approach with the tag-aware CF fusion model, following the tag-aware CF fusion model, the predictions of user-based CF part and item-based CF part in our fusion approach are computed differently. For the predicting item problem in user-based CF part, recommendations are a list of items that is ranked by decreasing frequency of occurrence in the ratings of his/her neighbours. The following equation gives the preference prediction of user u for an unused item i by the user-based CF part in the fusion model:

$$P_{u,i}^{MUCF2} = \frac{|\{v|v \in N_u, i \in I_v\}|}{|N_u|} \quad (7)$$

where N_u are the neighbour users of target user u, and I_v is the set of items used by a neighbour user v.

For the item-based CF part, the top-N item recommendation is to compute a list of items that is ranked by the decreasing sum of the similarities of neighbouring items, which have been used by user u. This preference prediction of the item-based CF part is given by Equation (6).

Since the preference predictions computed by user-based CF and item-based CF come from different computation methods, they have different scales of values. A normalization process of the preference predictions is needed to unify the recommendations from the two neighborhood-based CF parts, which produces the final preference prediction used for top-N item recommendation ranking:

$$P_{u,i}^{MCFF} = \lambda \cdot \frac{P_{u,i}^{MUCF2}}{\sum_{j \in \widetilde{I_u}} P_{u,j}^{MUCF2}} + (1-\lambda) \cdot \frac{P_{u,i}^{MICF}}{\sum_{j \in \widetilde{I_u}} P_{u,j}^{MICF}} \quad (8)$$

where $0 \leq \lambda \leq 1$, $\widetilde{I_u}$ is the set of new items to be recommended to target user u. Note the neighbourhood sizes of users and items are defined by the same parameter k.

The proposed MCFF approach for multidimensional data can reasonably enhance the recommendation performance, since this approach is able to not only efficiently utilize the multidimensional semantic relations, but also bring out the recommendation power of the localized neighbourhood relations of both users and items. In addition, the application of dimensionality reduction to the unfolded matrices can dramatically reduce the dimension problem while preserving the multidimensional interaction. In fact, our empirical analysis has shown that the proposed MCFF approach provides very promising performance.

4 Evaluation

In this section, we present empirical analysis based on real data collected from Bibsonomy and Delicious. Experimental results show the high effectiveness of the proposed multidimensional user/item profiling approach for making recommendations. The evaluation results of MCFF approach show significantly superior performances compared to other state-of-the-art CF approaches for multidimensional data.

4.1 Datasets

We conducted experiments using datasets from Bibsonomy (Knowledge and Data Engineering Group 2007) and Delicious (Wetzker et al. 2008). The Bibsonomy dataset was collected on 30 April 2007. The Delicious dataset was collected on January 2004. Following the evaluation of TF approach (Symeonidis et al. 2010) to make the datasets less sparse, the notion of *p*-core (Jäschke et al. 2007) was applied to the datasets. The *p*-core of level *k* means that each user, tag and item has/occurs in at least *k* posts. Following the evaluation of the TF approach, we use $k = 5$ for both of the two datasets. The original Delicious dataset contains 2419 users, 30838 items and 10926 tags. With $k = 5$, the Delicious dataset contains 216 users, 337 items, and 247 tags. The Bibsonomy dataset we obtained is already applied with $k = 5$ by the dataset provider, Knowledge and Data Engineering Group (Knowledge and Data Engineering Group 2007), and it contains 116 users, 361 items and 412 tags.

4.2 Evaluation Settings

4.2.1 Recommendation Models

Following are the proposed approaches to be examined:

- **Multidimensional Item-based CF (MiCF)**. This is the item-based CF approach integrated with the multidimensional item profiling proposed in Section 3.2.
- **Multidimensional User-based CF (MuCF)**. This is the user-based CF approach integrated with the multidimensional user profiling proposed in Section 3.2.
- **Multidimensional CF Fusion (MCFF)**. This is the multidimensional CF fusion approach proposed in Section 3.3.

In order to compare our proposed approaches against state-of-the-art recommendation algorithms as well as conventional neighborhood-based CF approaches, we have adopted the following models as the baseline models:

- **Item-based CF (iCF)**. This is the item-based CF approach (Deshpande and Karypis 2004). It is actually a 2-dimensional recommendation method with the implicit rating data as input.
- **User-based CF (uCF)**. This is the user-based CF approach (Adomavicius and Tuzhilin 2005). Similar to iCF, it is also a 2-dimensional recommendation method with the implicit rating data as input.
- **Tag-aware CF Fusion (tCFF)**. This CF fusion model uses tags as pseudo users in item-based CF and as pseudo items in user-based CF to extend the profiling ability of the two approaches (Tso-Sutter et al. 2008).
- **Tensor Factorization based CF (TF)**. Symeonidis et al. proposed a tensor factorization based recommender framework which uses HOSVD for factorizing 3-order user-item-tag tensors (Symeonidis et al. 2010). They use kernel-SVD in the process to further improve the recommendation accuracy of the

reconstructed tensors. Item recommendations are generated directly based on reconstructed tensors.

4.2.2 Evaluation Metrics

To evaluate the performance of top-N item recommendation, we adopt precision, recall and F1 score as the evaluation metrics (Herlocker et al. 2004). We conducted a 5-fold cross validation. For each run, we randomly choose 75% observed data of each user to form the training set, and the remaining 25% are used as testing data for evaluation.

4.2.3 Algorithms' Settings

Following are the specific settings used in the algorithms to be evaluated for the datasets.

- **iCF**. We have varied the parameter for the item neighbourhood size from 10 to 300 with a step size of 5 for the two datasets. For the Bibsonomy dataset, the best result was achieved when the item neighbourhood size equals to 100. For the Delicious dataset, the best result was achieved when the neighbourhood size equals to 40.
- **uCF**. For the Bibsonomy dataset, we have varied the parameter for the user neighbourhood size from 10 to 100 with a step size of 5, and the best result was achieved when the neighbourhood size equals to 30. For the Delicious dataset, we have varied the parameter for the neighbourhood size from 10 to 200 with a step size of 5, and the best result was achieved when the neighbourhood size equals to 30.
- **TF**. We follow the TF-based recommendation approach (Symeonidis et al. 2010) to determine the three dimensional parameters of core tensors. For the Bibsonomy dataset, we found when the three parameters were set as 96, 60 and 274 this model achieved its best results. For the Delicious dataset, we found when the three parameters were set as 61, 96 and 211 this model achieved its optimal results.
- **tCFF**. For both of the two datasets, we have varied the λ parameter from 0 to 1 by an interval of 0.1 and the neighborhood k parameter from 10 to 300 by an interval of 5. For the Bibsonomy dataset, we have found the best λ being 0.8 and k being 20. For the Delicious dataset, we have found the best λ being 0.7 and k being 70.

Following are the settings for the proposed models.

- **MiCF**. For the Bibsonomy dataset, we found the best results from this method came with factor number parameter k_i as 343 and item neighbourhood size as 100. For the Delicious dataset, we found the best results from this method came with factor number parameter k_i as 78 and user neighbourhood size as 70.
- **MuCF**. For the Bibsonomy dataset, we found the best results from this method came with factor number parameter k_u as 114 and user neighbourhood size as 30. For the Delicious dataset, we found the best results from this method came with factor number parameter k_u as 128 and user neighbourhood size as 60.
- **MCFF**. We have varied the λ parameter from 0 to 1 by an interval of 0.1 and the neighborhood k parameter from 10 to 300 by an interval of 5 for both of the two datasets. For the Bibsonomy dataset, we set the factor number parameter k_i as 343 and k_u as 114. We have found the best λ was 0.4 and k was 10. For the Delicious dataset, we set the factor number parameter k_i as 78 and k_u as 128. We have found the best λ to be 0.7 and k to be 30.

4.3 Experiment Results and Discussion

In this section, we present the detailed experiment results and discuss the performance of the recommendation models in terms of precision, recall and F1 score as the evaluation metrics.

4.3.1 Multidimensional Models

In this section, we compare the proposed multidimensional CF fusion model MCFF with two state-of-the-art multidimensional CF models: TF and tCFF. Specially, in Table 1 and Table 2, we give the precisions and recalls of the three recommenders for the Bibsonomy dataset respectively. Table 3 and Table 4 show the precisions and recalls of them for the Delicious dataset.

For each top-N value in Table 1 to Table 4, the largest values in each row are made bold to be more visible. The improvement of MCFF against the larger one between TF and tCFF is given in the last column for each line.

Top-N	TF	tCFF	MCFF	Improvement (%)
1	0.137931	0.112068	**0.189655**	37%
2	0.133620	0.116379	**0.176724**	32%
3	0.123563	0.120689	**0.186781**	51%
4	0.107758	0.120689	**0.176724**	46%
5	0.106897	0.115517	**0.158621**	37%
6	0.097701	0.107758	**0.143678**	33%
7	0.093596	0.102216	**0.139162**	36%
8	0.089439	0.092672	**0.127155**	37%
9	0.083333	0.089080	**0.118773**	33%
10	0.079310	0.082758	**0.109482**	32%

Table 1: Precisions of the three recommendation models for Bibsonomy dataset

Top-N	TF	tCFF	MCFF	Improvement (%)
1	0.029386	0.026598	**0.043586**	48%
2	0.057867	0.055901	**0.078361**	35%
3	0.078496	0.086119	**0.127735**	48%
4	0.088341	0.112434	**0.150250**	34%
5	0.110382	0.133640	**0.171977**	29%
6	0.122768	0.145023	**0.181867**	25%
7	0.143122	0.165190	**0.202885**	23%
8	0.156869	0.173811	**0.211089**	21%
9	0.161842	0.180211	**0.222519**	23%
10	0.171433	0.182537	**0.225577**	24%

Table 2: Recalls of the three recommendation models for Bibsonomy dataset

Top-N	TF	tCFF	MCFF	Improvement (%)
1	0.060185	**0.087962**	0.064815	-26%
2	0.055555	0.060185	**0.071759**	19%
3	0.049383	0.055556	**0.063271**	14%
4	0.046296	0.053240	**0.059028**	11%

Top-N	TF	tCFF	MCFF	Improvement (%)
5	0.045370	0.052778	**0.055556**	5%
6	0.043210	0.047839	**0.051698**	8%
7	0.041667	0.043650	**0.048942**	12%
8	0.039931	0.041667	**0.048611**	17%
9	0.039095	0.040637	**0.045267**	11%
10	0.036111	0.039351	**0.043056**	9%

Table 3: Precisions of the three recommendation models for Delicious dataset

Top-N	TF	tCFF	MCFF	Improvement (%)
1	0.019424	**0.027160**	0.020634	-24%
2	0.034509	0.038760	**0.047653**	23%
3	0.044501	0.052114	**0.061341**	18%
4	0.054370	0.066890	**0.073687**	10%
5	0.064748	0.082688	**0.087268**	6%
6	0.075743	0.088154	**0.094109**	7%
7	0.084790	0.094230	**0.103740**	10%
8	0.093124	0.102312	**0.116838**	14%
9	0.101303	0.110299	**0.122129**	11%
10	0.105161	0.119326	**0.126539**	6%

Table 4: Recalls of the three recommendation models for Delicious dataset

As shown in Table 1 to Table 4, basically for all top-N values, the proposed approach MCFF shows significantly superior recommendation performance compared to TF and tCFF. Although all of these three models utilize the relations between the three dimensions (users, items and tags) in the data in their own ways, compared to the other two approaches, the MCFF model makes use of not only the relationships between the three entities, but also the power of neighbourhoods. In comparison to TF, MCFF can unify the local relationships that are discovered by user-based and item-based neighbourhoods, which the TF model is incapable of. Compared to tCFF, on the one hand MCFF better preserves the multidimensional semantic relations in the data, which means the user and item neighbourhood formation are more accurate; on the other hand, MCFF also integrates the neighbourhood relations with holistic implicit relations among users, tags and items in the data through the dimensionality reduction applied on the entire data. In MCFF, different types of relations in the data can compensate each other. It is the unification of not only both user-based and item-based neighbourhoods but also holistic latent relations that leads the proposed model MCFF to the best recommendation performance. In addition, in Table 1, we can also observe that for some high top-N values (e.g., 1, 2, 3), TF shows better precisions than tCFF. This is because as enough tags are provided in the data, TF can show better top recommendations than tCFF, due to TF's ability to utilize the ternary relations among the users, items and tags, while tCFF discards this information.

Compared to Delicious dataset, MCFF shows higher improvement for Bibsonomy dataset. This may come from the fact that there are relatively more tags than users and items in Bibsonomy dataset, while in Delicious dataset the number of tags is smaller than the number of items. Since for both of the two datasets, we applied p-core of level k with $k = 5$, this indicates more relations regarding tags can lead to further recommendation improvement of MCFF.

4.3.2 Single Neighbourhood-based CF Models

In this section, in order to evaluate the effectiveness of the proposed multidimensional profiling approach, we compare the two multidimensional neighbourhood-based CF models MuCF and MiCF proposed in Section 3.1 and in Section 3.2, with their corresponding neighbourhood-based CF approaches, uCF and iCF. Specifically, in Table 5 and Table 6, we give the precisions and recalls of the recommenders for the Bibsonomy dataset. Table 7 and Table 8 present the precisions and recalls for the Delicious dataset.

Top-N	uCF	MuCF		iCF	MiCF
1	0.112068	**0.120689**		0.086206	**0.137931**
2	0.103448	**0.112068**		0.081896	**0.185344**
3	**0.117816**	**0.117816**		0.080459	**0.175287**
4	**0.116379**	0.107758		0.071120	**0.163793**
5	**0.103448**	0.100000		0.074137	**0.141379**
6	**0.094828**	**0.094828**		0.070402	**0.127873**
7	0.086206	**0.093596**		0.065270	**0.120689**
8	0.081896	**0.089440**		0.062500	**0.115301**
9	0.083333	**0.088123**		0.059386	**0.107279**
10	0.083620	**0.084483**		0.058620	**0.101724**

Table 5: Precisions of the single neighbourhood-based CF models for Bibsonomy dataset

Top-N	uCF	MuCF		iCF	MiCF
1	0.026087	**0.030420**		0.018813	**0.029581**
2	**0.056409**	0.049024		0.035600	**0.080644**
3	**0.082994**	0.080327		0.050214	**0.115904**
4	**0.101750**	0.089840		0.058876	**0.148074**
5	**0.112656**	0.107882		0.076505	**0.160539**
6	**0.123729**	0.117051		0.085358	**0.174338**
7	0.128354	**0.132278**		0.095928	**0.190031**
8	0.136575	**0.141515**		0.103297	**0.201811**
9	**0.154755**	0.153659		0.111193	**0.213270**
10	**0.169037**	0.161369		0.120339	**0.218762**

Table 6: Recalls of the single neighbourhood-based CF models for Bibsonomy dataset

Top-N	uCF	MuCF		iCF	MiCF
1	**0.069444**	**0.069444**		**0.050925**	0.046296
2	**0.064814**	**0.064814**		0.034722	**0.050925**
3	0.055556	**0.060185**		0.038580	**0.040123**
4	0.048611	**0.061342**		0.037037	**0.040509**
5	0.047222	**0.053703**		0.037962	**0.041666**
6	0.043209	**0.047068**		0.033951	**0.040123**
7	0.041005	**0.042328**		0.031084	**0.038359**
8	0.039352	**0.040509**		0.030092	**0.035301**
9	0.037037	**0.038065**		0.029320	**0.033951**
10	0.036574	**0.037037**		0.028703	**0.035185**

Table 7: Precisions of the single neighbourhood-based CF models for Delicious dataset

Top-N	uCF	MuCF		iCF	MiCF
1	**0.025733**	0.024035		0.013966	**0.016497**
2	**0.043518**	0.043056		0.016975	**0.031177**
3	0.054784	**0.056565**		0.028877	**0.036236**
4	0.061021	**0.073894**		0.039293	**0.048769**

Top-N	uCF	MuCF		iCF	MiCF
5	0.069969	**0.079732**		0.054263	**0.063468**
6	0.076721	**0.081230**		0.059819	**0.073421**
7	0.084341	**0.086014**		0.063952	**0.082025**
8	**0.093883**	0.093729		0.069816	**0.085754**
9	0.096796	**0.098256**		0.077885	**0.090088**
10	**0.106885**	0.105765		0.085562	**0.102087**

Table 8: Recalls of the single neighbourhood-based CF models for Delicious dataset

For both precision and recall, as we can see in Table 5 to Table 8, MiCF shows superior performance than iCF consistently. This is because the multidimensional item profiling approach is able to take into consideration the additional tag information and to utilize the 3-dimensional relationships, which leads to more refined item profiles. With this, neighbourhood formation is more accurate and thus the recommendation shows improved performance.

Interestingly, the comparison of MuCF against uCF is not consistent on the two datasets for precision and recall, as shown in Table 5 to Table 8. This phenomenon may be explained by the quantitative differences between users and items in the datasets. Since the unfolding matrices used in MuCF and in MiCF come from the same tensor, the information provided by these two matrices are essentially the same. Under this condition, the smaller number of users compared to items means there are larger number of non-zero elements in $\overrightarrow{u^e}$ than in $\overrightarrow{\imath^e}$. In Figure 2 and Figure 3, we present the distribution of counts of tag assignments (i.e., 3-tuple (u, i, t)) of each user and of each item in the two datasets, which are the non-zero elements in $\overrightarrow{u^e}$ and in $\overrightarrow{\imath^e}$. We sorted the user indices and item indices by the counts of tag assignments in ascending order to make the curves smooth. The red circles represent the counts of tag assignments of each user, and the blue squares correspond to that of items. As we can see, averagely each user has more tag assignments than each item does. Also, because the numbers of users are smaller than the items in the two datasets, the available factors in $\overrightarrow{u^f}$ will be less than those in $\overrightarrow{\imath^f}$. This implies the reduction from $\overrightarrow{u^e}$ to $\overrightarrow{u^f}$ leads to potentially more information loss than that from $\overrightarrow{\imath^e}$ to $\overrightarrow{\imath^f}$. In this way, $\overrightarrow{u^f}$ may not always provide sufficient information for each user in MuCF to generate more accurate profiles than what uCF does. On the other hand, for uCF and iCF, it is also due to the lower number of users and higher number of items that the profiles generated in uCF can be better than that generated by iCF. Because each user in uCF can potentially have more information for profiling. Moreover, it also increases the possibility for uCF to obtain neighbourhoods with higher quality than iCF. This results in the better performance of uCF over iCF. Similar observation was given previously by Desrosiers and Karypis (2011). Thus, the improvement of MuCF over uCF is less effective as shown in Table 5 to Table 8.

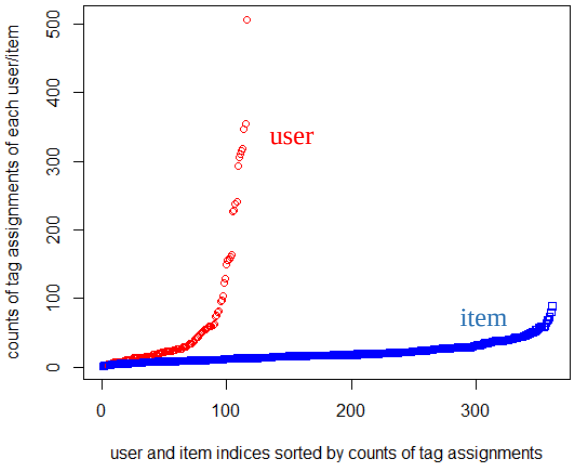


Figure 2: Distribution of counts of tag assignments of each user/item in the Bibsonomy dataset

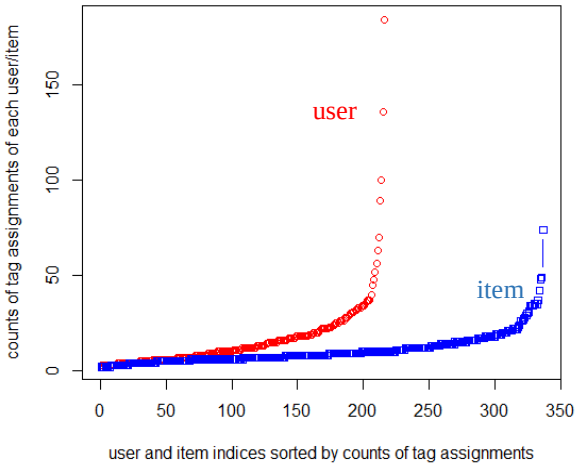


Figure 3: Distribution of counts of tag assignments of each user/item in the Delicious dataset

4.3.3 Overall Comparison of All Models

Figure 4 and Figure 5 show the experimental results of all recommendation models for the Bibsonomy dataset and the Delicious dataset respectively. As shown in the two figures, the proposed CF fusion model MCFF outperforms all of the rest of the recommendation models. To sum up, the proposed multidimensional CF fusion approach can incorporate not only the strengths of both user neighbourhood and item neighbourhood but also the multidimensional latent relations. Thereby it shows significant improvement compared to the rest of all other models in the experiments.

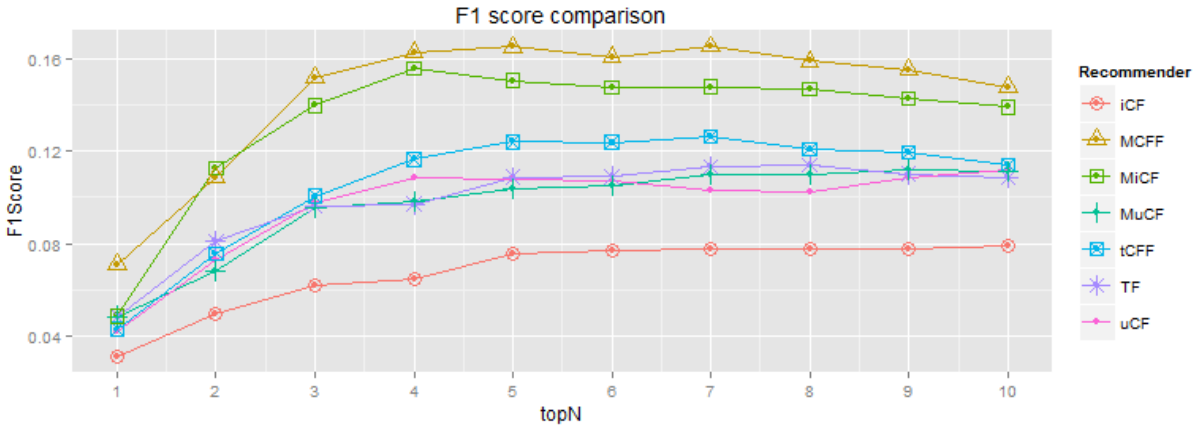


Figure 4: F1 scores of the recommenders for the Bibsonomy dataset

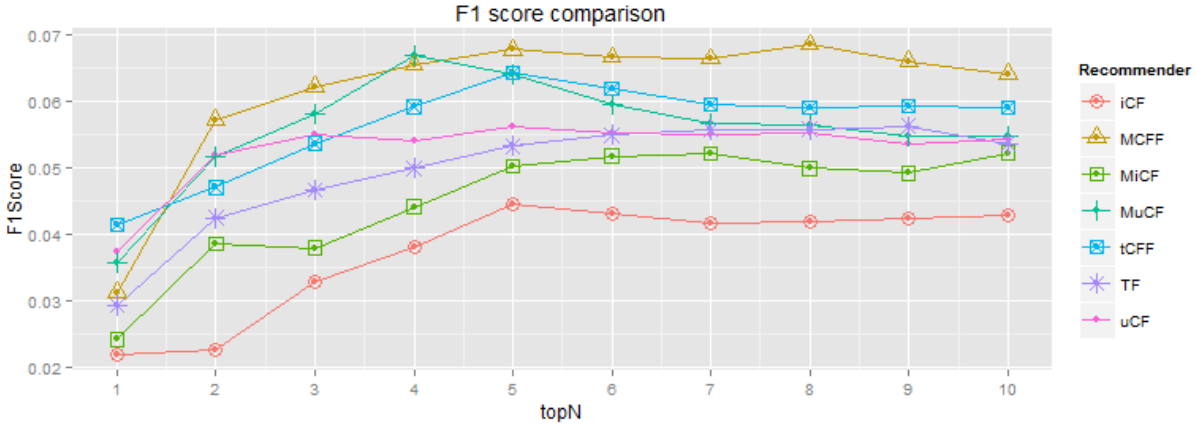


Figure 5: F1 scores of the recommenders for the Delicious dataset

5 Conclusion and Future Work

The increasing availability of multidimensional data and applications has provided recommender systems with new opportunities and challenges. In this paper, we proposed a multidimensional profiling approach for users and items for neighbourhood-based CF approaches, and proposed a multidimensional CF fusion approach based on that. We first model multidimensional data as a tensor. Through unfolding the tensor by different modes of the tensor, the multidimensional relations in the data can be exposed and used for user/item representation. We further utilize SVD to reduce the dimensionality and remove irrelevant noise in data. Then, based on the latent factors obtained, we can profile users and items efficiently and effectively. Note other dimensions in the data, e.g., tags, can also be profiled using a similar method if needed. After that, the conventional user-based CF and item-based CF can be enhanced using the multidimensional profiling technique. Finally, a novel CF fusion approach is proposed to unify the two multidimensional neighbourhood-based CF approach and thus gain superior recommendation performance. Additional feature information can be easily utilized via the proposed multidimensional profiling approach. Besides, recommendation of entities other than users and items, e.g. tags, can also be done by similar strategy. Experimental studies of the proposed multidimensional CF fusion approach on the Bibsonomy and Delicious datasets have shown significant improvements with regards to precision, recall and F1 score, compared to other state-of-the-art recommendation models. This confirms the proposed multidimensional profiling and CF fusion methods are effective.

For the future work, we intend to examine the integration of different types of additional features in the proposed approach, for example, time or location. Also, we want to explore the application of the proposed multidimensional profiling technique in tag recommender systems. For users who collected small numbers of items, the multidimensional relation for them are difficult to obtain because of the lack of sufficient information, this is also a problem worth looking into. We may give special attention to these special users in order to get a higher overall recommendation performance.

6 Acknowledgments

Computational resources and services used in this work were provided by the HPC and Research Support Group, Queensland University of Technology, Brisbane, Australia.

7 References

Acar, E. and Yener, B. (2009): Unsupervised multiway data analysis: A literature survey. *Knowledge and Data Engineering, IEEE Transactions on*, 21(1): 6-20.

Adomavicius, G. and Tuzhilin, A. (2005): Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6): 734-749.

Adomavicius, G. and Tuzhilin, A. (2011): Context-aware recommender systems. In *Recommender systems handbook*. 217-253. *Recommender systems handbook.* Springer.

Bar, A., Rokach, L., Shani, G., Shapira, B. and Schclar, A. (2013): Improving simple collaborative filtering models using ensemble methods. In *Multiple Classifier Systems*. 1-12. *Multiple Classifier Systems.* Springer.

Deshpande, M. and Karypis, G. (2004): Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22(1): 143-177.

Desrosiers, C. and Karypis, G. (2011): A comprehensive survey of neighborhood-based recommendation methods. In *Recommender systems handbook*. 107-144. *Recommender systems handbook.* Springer.

Herlocker, J., Konstan, J., Terveen, L. and Riedl, J. (2004): Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems*, 22(1): 5-53.

Jäschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L. and Stumme, G. (2007): Tag recommendations in folksonomies. In *Knowledge Discovery in Databases: PKDD 2007*. 506-514. *Knowledge Discovery in Databases: PKDD 2007*. Springer.

Karatzoglou, A., Amatriain, X., Baltrunas, L. and Oliver, N. (2010): Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. *Proceedings of the fourth ACM conference on Recommender systems*, 79-86, ACM.

Knowledge and Data Engineering Group, University of Kassel: Benchmark Folksonomy Data from BibSonomy. http://www.kde.cs.uni-kassel.de/bibsonomy/dumps/. Accessed 30 June 2007.

Kolda, T.G. and Bader, B.W. (2009): Tensor decompositions and applications. *SIAM review*, 51(3): 455-500.

Koren, Y., Bell, R. and Volinsky, C. (2009): Matrix factorization techniques for recommender systems. *Computer*, 42(8): 30-37.

Lee, J.-S. and Olafsson, S. (2009): Two-way cooperative prediction for collaborative filtering recommendations. *Expert Systems with Applications*, 36(3): 5353-5361.

Liang, H., Xu, Y., Li, Y., Nayak, R. and Tao, X. 2010. Connecting users and items with weighted tags for personalized item recommendations. *Proceedings of the 21st ACM conference on Hypertext and hypermedia*. Toronto, Ontario, Canada: ACM.

Marinho, L.B., Hotho, A., Jäschke, R., Nanopoulos, A., Rendle, S., Schmidt-Thieme, L., Stumme, G. and Symeonidis, P. (2012): *Recommender systems for social tagging systems.* Springer.

Rendle, S., Balby Marinho, L., Nanopoulos, A. and Schmidt-Thieme, L. (2009): Learning optimal ranking with tensor factorization for tag recommendation. *Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining*, 727-736, ACM.

Sen, S., Vig, J. and Riedl, J. (2009): Tagommenders: connecting users to items through tags. *Proceedings of the 18th international conference on World wide web*, Madrid, Spain, 671-680, ACM.

Su, X. and Khoshgoftaar, T.M. (2009): A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009.

Symeonidis, P., Nanopoulos, A. and Manolopoulos, Y. (2010): A unified framework for providing recommendations in social tagging systems based on ternary semantic analysis. *IEEE Transactions on Knowledge and Data Engineering*, 22(2): 179-192.

Tso-Sutter, K.H.L., Marinho, L.B. and Schmidt-Thieme, L. (2008): Tag-aware recommender systems by fusion of collaborative filtering algorithms. *Proceedings of the 2008 ACM symposium on Applied computing*, 1995-1999, ACM.

Wang, J., De Vries, A.P. and Reinders, M.J.T. (2006): Unifying user-based and item-based collaborative filtering approaches by similarity fusion. *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 501-508, ACM.

Wetzker, R., Zimmermann, C. and Bauckhage, C. (2008): Analyzing social bookmarking systems: A del. icio. us cookbook. *Proceedings of the ECAI 2008 Mining Social Data Workshop*, 26-30, IOS Press.

Real-time Collaborative Filtering Recommender Systems

Huizhi Liang[1,2] **Haoran Du**[2] **Qing Wang**[2]

[1] Department of Computing and Information Systems,
The University of Melbourne,
Victoria 3010, Australia
Email: huizhi.liang@unimelb.edu.au

[2] Research School of Computer Science,
The Australian National University,
Canberra ACT 0200, Australia
Email: duhaoranshux@hotmail.com, Qing.wang@anu.edu.au

Abstract

Recommender systems can help users deal with the information overload issue. Many real-world communities such as social media websites require real-time recommendation making to capture the recent updates of the communities. This brings challenges to existing approaches which mainly build recommendation models at offline. In this paper, we discuss real-time collaborative filtering recommendation approaches. The proposed approaches use locality sensitive hashing (LSH) to construct user or item blocks, which facilitate real-time neighborhood formation and recommendation making. The experiments conducted on a Twitter dataset demonstrate the effectiveness of the proposed approaches.

Keywords: Real-time, Locality Sensitive Hashing, Collaborative Filtering, Recommender System

1 Introduction

Recommender systems is one of the popular personalization applications, which help to solve the information overload issue of users in online communities, i.e., making suggestions regarding which information is most relevant to an individual user. Collaborative filtering approaches such as user-based and item-based K-nearest neighbor methods are widely used to make recommendations in various areas (Adomavicius & Tuzhilin 2005). Collaborative filtering recommender systems usually consist of two phases: (1)An offline model-building phase to build a model storing correlations between users and items. (2) An on-demand recommendation phase that uses the model to make recommendations (Chandramouli et al. 2011).

However, the traditional offline collaborative filtering recommender systems fail to capture the rapid changes of online communities to make real-time recommendations. For example, with the rapid growth of users in social media communities, there are a large number of micro-blog topics emerging every day. They include not only a small number of hot or stream topics but also a large number of less popular topics. Thus, it is important to recommend personally interesting topics to users (Liang et al. 2012). However, since the topics of micro-blogs are constantly changing, it brings difficulty for an offline-built model to capture the latest updates in social media communiteis (Liang et al. 2012).

Neighborhood formation is the key component of collaborative filtering recommender systems. Typically, pair-wise comparisons such as Cosine similarity calculation are commonly used to build the correlations (i.e. find the the nearest neighbors of each user or item). To meet the requirement of real-time response, we need to decrease the number of pair-wise comparisons and find the nearest neighbor users and candidate items quickly. Blocking or indexing techniques can help to significantly decrease the number of comparisons (Christen 2012). The objects in a database can be inserted into one or more blocks according to some blocking criteria, such that only objects within a block are compared with each other. The current blocking techniques are mainly focusing on content features, such as inverted indexing of keywords, and phonetic encoding functions (e.g., Soundex, Double Metaphone) (Christen 2012). Locality sensitive hashing (LSH) (Gionis et al. 1999) is an approximate blocking approach that uses a set of hash functions to map data objects such as users or items within a certain distance range into the same block with a given probability. It can filter out those data objects with low similarities for a given data object, thus decreasing the number of comparisons (Gan et al. 2012). LSH can generate blocks quickly and has advantages such as dimension reduction, noise-tolerant, and similarity-preserving. It has been widely used in industries, such as personalized news recommendation in Google (Li et al. 2011).

We discuss real-time collaborative filtering recommendation approaches. The proposed approaches employ the LSH techniques to construct user and item blocks. Then, we propose approaches to form neighborhood and make recommendations in real-time, based on the generated user and item blocks.

2 Related Work

Recommender systems have been an active research area for more than a decade. The recommendation approaches based on explicit ratings are the major focus. The tasks of recommender systems include rating prediction and top N recommendation. The former

This research was partially funded by the Australian Research Council (ARC), Veda Advantage, and Funnelback Pty. Ltd., under Linkage Project LP100200079. Note the first two authors contributed equally.

task that is to predict the rating value a user will give to a rated item while the latter one is to recommend a set of unrated or new items to the target user (Adomavicius & Tuzhilin 2005). The Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are widely used to measure the accuracy of the rating prediction while precision and recall are commonly used for the top N recommendation. For explicit ratings, both tasks are applicable while for implicit ratings, the top N recommendation is more applicable (Adomavicius & Tuzhilin 2005). Recommender systems can be broadly classified into three categories: content-based, collaborative filtering (CF), and hybrid approaches (Adomavicius & Tuzhilin 2005). The user-based and item-based K-nearest neighborhood collaborative filtering are widely used in various application areas.

Approximate blocking techniques such as LSH and tree-based indexing (Bawa et al. 2005) are widely used in nearest neighbor and similarity search in applications such as image search (Dong et al. n.d.), recommender systems (Li et al. 2011), and entity resolution (Kim & Lee n.d., Liang et al. 2014, Li et al. 2013). Recently, some work has been proposed to make real-time recommendations (Chandramouli et al. 2011). For example, Abbar et al. (Abbar et al. 2013) proposed a real-time recommender system for diverse related articles. Li et al. (Li et al. 2012) proposed interest-based real-time content recommendation in online social communities. Diaz-Aviles et al. (Diaz-Aviles et al. 2012) proposed real-time top N matrix factorization recommendation in social streams. Moreover, approximate blocking techniques such as LSH and tree based indexing (Bawa et al. 2005) are used to make efficient news recommendations (Li et al. 2011). However, how to make real-time collaborative filtering recommendations still needs to be explored.

3 Problem Definition

We define some key concepts used in this paper.

- **Users:** $U = \{u_1, u_2, ..., u_{|U|}\}$ contains all users in an online community who have rated or published items.
- **Items (i.e., Products, Topics):** $C = \{c_1, c_2, ..., c_{|C|}\}$ contains all items rated or published by users in U. Items could be any type of online information resources or products in an online community such as web pages, video clips, music tracks, photos, movies, books, topics of micro-blogs (Liang et al. 2012) etc.
- **User profile:** A user profile is a collection of information about a user, such as demographic information, interests or preferences, opinions, friends or other network information. Users' interests or preferences are typical information to profile users. We use binary or numeric weight values for items to represent a user's interests or preferences for items.

Let $u_i \in U$ be a target user, C_{u_i} be the item set that user u_i already has, $\mathbf{C}_{u_i}$ be the candidate item set that are unknown to user u_i, i.e., $\mathbf{C}_{u_i} = C - C_{u_i}$. Let $c_x \in \mathbf{C}_{u_i}$ be a candidate item, $\mathcal{A}(u_i, c_x)$ be the predicted score of how much the user u_i would be interested in item c_x. The problem of top N item recommendation is defined as generating a set of ordered items $c_l, ..., c_m \in \mathbf{C}_{u_i}$ to the use u_i, where $\mathcal{A}(u_i, c_l) \geq ... \geq \mathcal{A}(u_i, c_m)$.

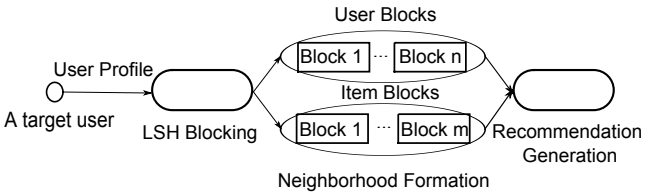


Figure 1: The Framework of Real-time Collaborative Filtering Recommender Systems

4 The Proposed Approach

In this section, we discuss how to conduct real-time collaborative filtering recommendations. The system framework is shown in Figure 1. It describes the key components of real-time collaborative filtering recommender systems, including LSH blocking, neighborhood formation, and recommendation generation.

For real-time user-based collaborative filtering approach, we construct user blocks based on LSH blocking scheme. For a given target user $u_i \in U$, we firstly get the hash signatures of this user based on a LSH family. Then, $u_i \in U$ is allocated to a set of blocks that use the hash signatures as block identifiers. The users that are in the same blocks with $u_i \in U$ are selected as being the neighbor users of u_i. Then, we select the candidate items from the neighbor users and generate a list of recommended items to u_i.

For real-time item-based collaborative filtering approach, we construct item blocks based on a LSH blocking scheme. For a given target user $u_i \in U$, we get the hash signature of each item $c_j \in C_{u_i}$ based on a LSH family. Then, each item $c_j \in C_{u_i}$ is allocated to a set of blocks. The items that are in the same blocks with $c_j \in C_{u_i}$ are selected as being the neighbor items of item c_j. Then, the neighbor items are selected as the candidate items for user $u_i \in U$. The top N ranked candidate items are selected as recommended items for $u_i \in U$.

In the following, we first discuss the LSH blocking scheme that is used to construct user or item blocks based on their Cosine similarities. Then we discuss how to select nearest neighbor users or items based on the generated user or item blocks. After that, we discuss how to make real-time user-based and item-based recommendations.

4.1 LSH Blocking Scheme

Let h denote a hash function for a given distance measure D, $Pr(i)$ denote the probability of an event i, p_1 and p_2 are two probability values, $p_1 > p_2$, $0 \leq p_1, p_2 \leq 1$. h is called (d_1, d_2, p_1, p_2)-sensitive for D, for any data objects x and y, the following conditions hold:

1. if $D(x, y) \leq d_1$ then $Pr(h(x) = h(y)) \geq p_1$
2. if $D(x, y) > d_2$ then $Pr(h(x) = h(y)) \leq p_2$

Popularly used LSH families include the minHash family for Jaccard distance (Anand & Ullman 2011), the random hyperplane projection family for Cosine distance (Anand & Ullman 2011), and the p-stable distribution family for Euclidean Distance (Anand & Ullman 2011). As Cosine distance/similarity is popularly used to measure the similarity of two users or items that are represented as vectors. We discuss how to generate user and item blocks based on a random hyperplane projection family that approximates the Cosine distance/similarity of two vectors.

4.1.1 Random Hyperplane Projection

The random projection method of LSH (Anand & Ullman 2011) is designed to approximate Cosine distance/similarity of any two vectors. The basic idea of this technique is to choose a d-dimensional random hyperplane and use the hyperplane to hash input vectors.

Given an input vector with $n-$dimensions $\vec{x}$, a family $\mathcal{H}_r$ of hash functions such that, for a randomly chosen vector $\overrightarrow{v} \in V$ in a n-dimensional space, a hashing function $h \in \mathcal{H}_r$ is defined as:

$$h(\vec{x}) = \begin{cases} 1 & \text{if } \vec{v} \cdot \vec{x} > 0; \\ 0 & \text{if } \vec{v} \cdot \vec{x} < 0 \end{cases}$$

Each possible choice of $\overrightarrow{v}$ defines a single hash function. This hash function produces a single bit signature for the input vector $\vec{x}$. $\mathcal{H}_r$ contains a set of such functions (i.e., d-dimension) and produces a set of bit signatures. Accordingly, the probability that such hash function family separates two vectors $\vec{x}$ and $\vec{y}$ is directly proportional to the angle between the two vectors (Anand & Ullman 2011):

$$Pr[h(\vec{x}) = h(\vec{y})] = 1 - \frac{\theta(\vec{x}, \vec{y})}{\pi} \quad (1)$$

Following Equation 1, we have,

$$cos(\theta(\vec{x}, \vec{y})) = cos((1 - Pr[h(\vec{x}) = h(\vec{y})])\pi) \quad (2)$$

Thus, Equation 2 provides us a way of approximately calculating Cosine distance/similarity between two vectors. A vector with dimension n is mapped to a binary signature vector with dimension d based on the hash family $\mathcal{H}_r$, usually $d \ll n$. From a probabilistic viewpoint, the more random vectors we use, the more accurate the Cosine distance/similarity between two vectors is. After we get the binary signature of two input vectors $\vec{x}$ and $\vec{y}$, we can use the Hamming distance of their signatures to measure their similarity or distance.

4.1.2 Random Bit Sampling for Hamming Distance

Usually computing the Hamming distance of data objects requires pair-wise similarity/distance calculation. For example, a target user needs to compare with all the other users. To reduce the number of pair-wise comparisons, we can generate blocks for data objects based on their Hamming distance of signatures. We use random bit sampling (Anand & Ullman 2011) to approximate the Hamming distance over d-dimensional signature vectors $\{0,1\}^d$.

A LSH family $\mathcal{F}$ for Hamming distance is simply the family of all the projections of data objects on one of the d coordinates, i.e., $\mathcal{F} = \{h : \{0,1\}^d \rightarrow \{0,1\} \mid h(x) = x_i, i = 1, ..., d\}$, where x_i is the ith coordinate of x. A random function h from $\mathcal{F}$ simply selects a random bit from the input vector. The basic signature is called a length-1 signature and the hash function is called a length-1 hash function.

To amplify the collision probability, given a (d_1, d_2, p_1, p_2)-sensitive family $\mathcal{F}$, we can construct new families $\mathcal{H}_s$ by the combination of AND-construction or OR-construction of $\mathcal{F}$ (Anand & Ullman 2011). Let $\mathcal{H}_s = \{H_1, H_2, ..., H_l\}$ denote a LSH family that has l number of length-k hash functions. Each length-k hash function is formed by AND-construction of k length-1 hash functions. The l number of hash functions of $\mathcal{H}_s$ has an "OR" relationship between each other. The collision probability of $\mathcal{H}_s$ can be estimated with $p_{k,l} = 1 - (1 - p^k)^l$ where p denotes the collision probability of a length-1 hash function.

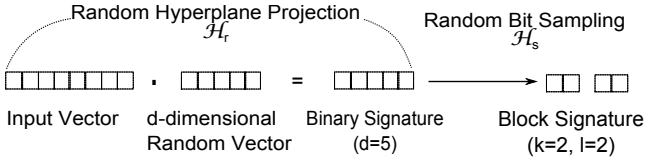


Figure 2: The LSH Blocking Scheme

The LSH blocking scheme $\mathbf{H}$ that are used in this paper consists of two hash families, $\mathbf{H} = \{\mathcal{H}_r, \mathcal{H}_s\}$. Firstly, we use a random hyperplane projection hash family $\mathcal{H}_r$ with d hash functions to get a d-dimensional binary hash signature for each input vector. Then, for each generated hash signature vector, we use a random sampling hash family $\mathcal{H}_s$ to get l length-k hash signatures. $\mathbf{H}$ has three parameters: d, k, and l. Figure 2 illustrate the process of getting the block signature for one input vector.

4.2 Neighborhood Formation

Neighborhood formation is to generate a set of like-minded peers (i.e., similar users) for a target user $u_i \in U$ or a set of similar items for an item $c_i \in C$. This paper adopts the "K-Nearest-Neighbors" technique to find the neighborhood for a user or an item. Typically, the user based K-Nearest-Neighborhood formation approach selects the top K neighbor users with shortest distances to a user u_i through computing the distances between user u_i and all other users of U. While the item-based K-Nearest-Neighborhood formation approach selects the top K neighbor items with the shortest distances to an item c_i through calculating the distances between item c_i and all other items. The distance or similarity measure can be calculated through various kinds of proximity computing approaches such as Cosine similarity and Pearson correlation (Adomavicius & Tuzhilin 2005).

To find the neighborhood of each target user $u_i \in U$ quickly, we construct user blocks and item blocks for users and items respectively. A $|C|$-sized item vector with weight values (denoted as $\vec{u_i}$) which represents user u_i's item preferences is used to profile user u_i. With the LSH blocking scheme $\mathbf{H}$, we can get l hash signatures (denoted as S_i) for user $u_i \in U$. The users that have the same signature will be allocated into the same block. Parameter k decides the similarity threshold (i.e., Hamming distance threshold) of a block. The users in the same blocks with user u_i are the neighbor users of u_i. Thus, we can form neighbourhood quickly via hashing. This approach can filter out users that have low similarities with the target user u_i, thus decreasing the number of pair-wise comparisons. With hashing, we also can update the neighborhood of users quickly after users update their item preferences.

Similarly, we can construct item blocks for each item. For each item c_j, a $|U|$-sized user vector with weight values (denoted as $\vec{c_j}$) is used to represent item c_j. With the LSH blocking scheme $\mathbf{H}$, we can get l hash signatures (denoted as S_j) for item $c_j \in C$. The setting of parameters of $\mathbf{H}$ can be different from the setting to construct user blocks for the purpose of preserving items with different similarity ranges in blocks. The items in the same blocks with item c_j are the neighbor items of c_j.

4.3 Real-time Recommendation Generation

For a given target user $u_i \in U$, with the LSH blocking scheme $\mathbf{H}$, we can generate signatures and construct blocks for each user and each item to form neighborhood. We discuss user-based and item-based real-time recommendation approach.

4.3.1 User-based Recommendation

With users blocks, we can find neighbor users quickly. For a given target user $u_i \in U$, the users in the same user blocks with $u_i \in U$ are the neighbor users of user $u_i \in U$, denoted as $\mathcal{N}_{u_i}$. Typically, we can calculate the pair-wise Cosine similarity of the neighbor user and $u_i \in U$ to select K-nearest neighbor users. However, for large-scale datasets, the number of neighbor users can be large and it is time-consuming to conduct pair-wise comparisons for all neighbor users in the same block with $u_i \in U$. To further decrease the number of candidate neighbors and select a smaller set of nearest neighbors, we count the collision number of each user in all l blocks with user $u_i \in U$ to rank the neighbor users of $u_i \in U$. This is because the number of co-occurrences of a user u_x that appears together with u_i in the blocks reflects the similarity of the two users (Gan et al. 2012). The higher the number of co-occurrences is, the more similar the two users are. We set a threshold φ to select those users that appear at least φ times with the target user u_i together in blocks. Let g_{ix} denote the co-occurrence of user u_x and target user u_i. Let $\mathbf{N}_{u_i}$ denote the selected nearest neighbor record set of u_i. For each user $u_x \in \mathcal{N}_{u_i}$, we add u_x to $\mathbf{N}_{u_i}$ if $g_{ix} > \varphi$, $0 \leq \varphi \leq l$.

Thus, for each target user u_i, we can select top $|\mathbf{N}_{u_i}|$ nearest neighbor users and use a more sophisticated similarity measure approach (i.e., Cosine) to conduct pair-wise similarity calculation. As the time complexity of counting collision number is less than that of other similarity measure approach, the query time can be improved when we employ dynamic collision counting to rank candidate users.

For each target user u_i, a set of candidate items can be generated from the items of user u_i's neighbour users. Let $\mathbf{C}_{u_i}$ denote the the candidate items of user u_i. $\mathbf{C}_{u_i} = \{c_k | c_k \in C_{u_j}, c_k \notin C_{u_i}, u_j \in \mathbf{N}_{u_i}\}$, where C_{u_j} is the items of u_j. Let U_{c_x} denote the user set of item c_x, for each candidate item $c_x \in \mathbf{C}_{u_i}, \mathbf{N}_{u_i} \bigcap U_{c_x}$ is the subset of users in $\mathbf{N}_{u_i}$ who have used item c_x, the prediction score of how much u_i will be interested in $c_x \in \mathbf{C}_{u_i}$ is calculated by considering the similarities between user u_i and those users who are the neighbors of user u_i and have item c_x:

$$\mathcal{A}_u(u_i, c_x) = \sum_{u_j \in \mathbf{N}_{u_i} \bigcap U_{c_x}} \frac{1}{\sqrt{|\mathbf{N}_{u_i} \bigcap U_{c_x}|}} \cdot cosine(\vec{u_i}, \vec{u_j}) \quad (3)$$

The top N items with high prediction scores will be recommended to the target user u_i.

4.3.2 Item-based Recommendation

Similarly, we can generate the top K nearest neighbor items of each item c_j with item blocks. For a given target user $u_i \in U$, let C_{u_i} denote the item set of u_i, the similar items of each item $c_j \in C_{u_i}$ can be used as candidate items for user $u_i \in U$. The candidate item set of item $c_j \in C_{u_i}$ is denoted as $\mathcal{N}_{c_j}$. To further decrease the number of candidate items and select a smaller set of nearest neighbors for each item $c_j \in C_{u_i}$, we count the collision number of each item c_x in all l blocks with item $c_j \in C_{u_i}$ to rank the neighbor items of c_j. Let g_{jx} denote the co-occurrence of item c_x and the item $c_j \in C_{u_i}$ of user u_i. Let $\mathbf{N}_{c_j}$ denote the selected nearest neighbor item set of c_j. For each item $c_x \in \mathcal{N}_{c_j}$, we add c_x to $\mathbf{N}_{c_j}$ if $g_{jx} > \kappa$, $0 \leq \kappa \leq l$. Let $\mathfrak{C}_{u_i}$ denote the selected candidate item set of user u_i, $\mathfrak{C}_{u_i} = \{c_x | c_x \in \mathbf{N}_{c_j}, c_x \notin C_{u_i}, c_j \in C_{u_i}\}$

The prediction score of how much u_i will be interested in item $c_x \in \mathfrak{C}_{u_i}$ is calculated by considering the similarities between item c_x and each item c_j of user u_i:

$$\mathcal{A}_c(u_i, c_x) = \sum_{c_j \in C_{u_i}} \frac{1}{\sqrt{|C_{u_i}|}} \cdot cosine(\vec{c_j}, \vec{c_x}) \quad (4)$$

The top N items with high prediction scores will be recommended to the target user u_i.

5 Experimental Design

In the experiments, we recommend topics to users in a social media community. The experiments were conducted on a real-world dataset crawled from Twitter.com (Liang et al. 2012). After removing the stop words, we select the keywords that are at least used by 5 users as topics. To avoid sparseness, we only selected those users who have used at least 5 topics. The dataset has 2320 users and 3319 topics extracted from 1,214,604 tweets. The user set was split into training and test user sets. We randomly select 10% of users as test user set while the rest 90% users are used as training users. The training user set has 2088 users, the test user set has 232 users. We randomly select 50 items of each test user as the test/answer topic set of this user.

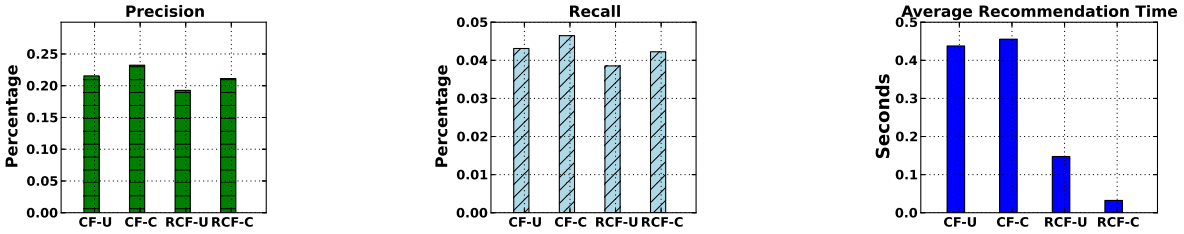
For each test user, the recommender system will generate a list of ordered topics that the test user has not used in his/her training set. The top N topics with high prediction scores will be recommended to the test user. If a topic in the recommendation list was in the test user's test/answer topic set, then this recommended topic was counted as a hit. We adopt Precision and Recall metrics to evaluate the accuracy of recommendations of the proposed approaches. Moreover, to evaluate the efficiency, the average online recommendation time over all test users is used.

6 Experimental Results

To evaluate the effectiveness of the proposed approaches, we have conducted comparison experiments of the following methods.

- ***CF-U***: The traditional user-based K-Nearest Neighborhood collaborative filtering approach.
- ***RCF-U***: The proposed real-time user-based collaborative filtering approach.
- ***CF-C***: The traditional item-based K-Nearest Neighborhood collaborative filtering approach.
- ***RCF-C***: The proposed real-time item-based collaborative filtering approach.

We set $K = 100$ and $N = 10$. For the proposed RCF-U and RCF-C, we set $d = 10$, $k = 4$, and $l = 8$ for hash family $\mathbf{H}$. $\varphi = 2$, $\kappa = 2$. The performance of the compared approaches are shown in Figure 3. The Precision and Recall values of all the approaches are



(a) Precision (b) Recall (c) Average Recommendation Time

Figure 3: The comparison results

low. This is because the dataset is sparse. The proposed real-time user-based and item-based collaborative filtering approach RCF-U and RCF-C achieved very close precision and recall results with the traditional collaborative filtering approaches CF-U and CF-C. RCF-U and RCF-C conducted much quicker recommendations. This can be explained that the proposed approaches largely decreased the number of pair-wise comparisons. They can incrementally and efficiently identify nearest neighbors when new updates occur in the community. Thus, they can be used for real-time recommendation making in online communities which requires quick responses for users' updates such as social media communities.

7 Conclusions

We discussed a real-time user-based and item-based collaborative filtering recommendation approach. To facilitate real-time recommendation, we adopt a LSH family that approximates Cosine distance/similarity and a LSH family that approximates Hamming distance to construct user and item blocks. Then, we discussed how to identify a set of nearest neighbors efficiently and how to rank candidate items quickly. As LSH techniques can be used for various types of item contents (e.g., text, image, numeric or binary weight values), this approach is applicable for various kinds of communities, especially those communities that have items with high dimensional content information and require quick recommendation responses to users updates. The experiments were conducted on a Twitter dataset and make real-time topic recommendations. The future work will consider the temporal effect of items and users.

References

Abbar, S., Amer-Yahia, S., Indyk, P. & Mahabadi, S. (2013), Real-time recommendation of diverse related articles, *in* 'WWW', pp. 1–12.

Adomavicius, G. & Tuzhilin, A. (2005), 'Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions', *TKDE* **17**(6), 734–749.

Anand, R. & Ullman, J. D. (2011), *Mining of massive datasets*, Cambridge University Press.

Bawa, M., Condie, T. & Ganesan, P. (2005), LSH forest: self-tuning indexes for similarity search, *in* 'WWW', pp. 651–660.

Chandramouli, B., Levandoski, J. J., Eldawy, A. & Mokbel, M. F. (2011), Streamrec: A real-time recommender system, *in* 'SIGMOD', pp. 1243–1246.

Christen, P. (2012), *Data Matching*, Data-Centric Systems and Appl., Springer.

Diaz-Aviles, E., Drumond, L., Schmidt-Thieme, L. & Nejdl, W. (2012), Real-time top-n recommendation in social streams, *in* 'RecSys', pp. 59–66.

Dong, W., Wang, Z., Josephson, W., Charikar, M. & Li, K. (n.d.), Modeling lsh for performance tuning, *in* 'CIKM'08', ACM, pp. 669–678.

Gan, J., Feng, J., Fang, Q. & Ng, W. (2012), Locality-sensitive hashing scheme based on dynamic collision counting, *in* 'SIGMOD', pp. 541–552.

Gionis, A., Indyk, P., Motwani, R. et al. (1999), Similarity search in high dimensions via hashing, *in* 'VLDB', pp. 518–529.

Kim, H.-S. & Lee, D. (n.d.), HARRA: fast iterative hashed record linkage for large-scale data collections, *in* 'EDBT'10', ACM, pp. 525–536.

Li, D., Lv, Q., Xie, X., Shang, L., Xia, H., Lu, T. & Gu, N. (2012), 'Interest-based real-time content recommendation in online social communities', *Knowledge-Based System* **28**, 1–12.

Li, L., Wang, D., Li, T., Knox, D. & Padmanabhan, B. (2011), Scene: a scalable two-stage personalized news recommendation system., *in* 'SIGIR', pp. 125–134.

Li, S., Liang, H. & Ramadan, B. (2013), Two stage similarity-aware indexing for large-scale real-time entity resolution, *in* 'AusDM 2013', CRPIT, Vol. 146.

Liang, H., Wang, Y., Christen, P. & Gayler, R. W. (2014), Noise-tolerant approximate blocking for dynamic real-time entity resolution, *in* 'PAKDD', pp. 449–460.

Liang, H., Xu, Y., Tjondronegoro, D. & Christen, P. (2012), Time-aware topic recommendation based on micro-blogs, *in* 'CIKM', pp. 1657–1661.

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 124 - Database Technologies 2012
Edited by Rui Zhang, The University of Melbourne, Australia and Yanchun Zhang, Victoria University, Australia. January 2012. 978-1-920682-95-8.
Contains the proceedings of the Twenty-Third Australasian Database Conference (ADC 2012), Melbourne, Australia, 30 January – 3 February 2012.

Volume 125 - Information Security 2012
Edited by Josef Pieprzyk, Macquarie University, Australia and Clark Thomborson, The University of Auckland, New Zealand. January 2012. 978-1-921770-06-7.
Contains the proceedings of the Tenth Australasian Information Security Conference (AISC 2012), Melbourne, Australia, 30 January – 3 February 2012.

Volume 126 - User Interfaces 2012
Edited by Haifeng Shen, Flinders University, Australia and Ross T. Smith, University of South Australia, Australia. January 2012. 978-1-921770-07-4.
Contains the proceedings of the Thirteenth Australasian User Interface Conference (AUIC2012), Melbourne, Australia, 30 January – 3 February 2012.

Volume 127 - Parallel and Distributed Computing 2012
Edited by Jinjun Chen, University of Technology, Sydney, Australia and Rajiv Ranjan, CSIRO ICT Centre, Australia. January 2012. 978-1-921770-08-1.
Contains the proceedings of the Tenth Australasian Symposium on Parallel and Distributed Computing (AusPDC 2012), Melbourne, Australia, 30 January – 3 February 2012.

Volume 128 - Theory of Computing 2012
Edited by Julián Mestre, University of Sydney, Australia. January 2012. 978-1-921770-09-8.
Contains the proceedings of the Eighteenth Computing: The Australasian Theory Symposium (CATS 2012), Melbourne, Australia, 30 January – 3 February 2012.

Volume 129 - Health Informatics and Knowledge Management 2012
Edited by Kerryn Butler-Henderson, Curtin University, Australia and Kathleen Gray, University of Melbourne, Australia. January 2012. 978-1-921770-10-4.
Contains the proceedings of the Fifth Australasian Workshop on Health Informatics and Knowledge Management (HIKM 2012), Melbourne, Australia, 30 January – 3 February 2012.

Volume 130 - Conceptual Modelling 2012
Edited by Aditya Ghose, University of Wollongong, Australia and Flavio Ferrarotti, Victoria University of Wellington, New Zealand. January 2012. 978-1-921770-11-1.
Contains the proceedings of the Eighth Asia-Pacific Conference on Conceptual Modelling (APCCM 2012), Melbourne, Australia, 31 January – 3 February 2012.

Volume 133 - Australian System Safety Conference 2011
Edited by Tony Cant, Defence Science and Technology Organisation, Australia. April 2012. 978-1-921770-13-5.
Contains the proceedings of the Australian System Safety Conference (ASSC 2011), Melbourne, Australia, 25th – 27th May 2011.

Volume 134 - Data Mining and Analytics 2012
Edited by Yanchang Zhao, Department of Immigration and Citizenship, Australia, Jiuyong Li, University of South Australia, Paul J. Kennedy, University of Technology, Sydney, Australia and Peter Christen, Australian National University, Australia. December 2012. 978-1-921770-14-2.
Contains the proceedings of the Tenth Australasian Data Mining Conference (AusDM'12), Sydney, Australia, 5–7 December 2012.

Volume 135 - Computer Science 2013
Edited by Bruce Thomas, University of South Australia, Australia. January 2013. 978-1-921770-20-3.
Contains the proceedings of the Thirty-Sixth Australasian Computer Science Conference (ACSC 2013), Adelaide, Australia, 29 January – 1 February 2013.

Volume 136 - Computing Education 2013
Edited by Angela Carbone, Monash University, Australia and Jacqueline Whalley, AUT University, New Zealand. January 2013. 978-1-921770-21-0.
Contains the proceedings of the Fifteenth Australasian Computing Education Conference (ACE 2013), Adelaide, Australia, 29 January – 1 February 2013.

Volume 137 - Database Technologies 2013
Edited by Hua Wang, University of Southern Queensland, Australia and Rui Zhang, University of Melbourne, Australia. January 2013. 978-1-921770-22-7.
Contains the proceedings of the Twenty-Fourth Australasian Database Conference (ADC 2013), Adelaide, Australia, 29 January – 1 February 2013.

Volume 138 - Information Security 2013
Edited by Clark Thomborson, University of Auckland, New Zealand and Udaya Parampalli, University of Melbourne, Australia. January 2013. 978-1-921770-23-4.
Contains the proceedings of the Eleventh Australasian Information Security Conference (AISC 2013), Adelaide, Australia, 29 January – 1 February 2013.

Volume 139 - User Interfaces 2013
Edited by Ross T. Smith, University of South Australia, Australia and Burkhard C. Wünsche, University of Auckland, New Zealand. January 2013. 978-1-921770-24-1.
Contains the proceedings of the Fourteenth Australasian User Interface Conference (AUIC 2013), Adelaide, Australia, 29 January – 1 February 2013.

Volume 140 - Parallel and Distributed Computing 2013
Edited by Bahman Javadi, University of Western Sydney, Australia and Saurabh Kumar Garg, IBM Research, Australia. January 2013. 978-1-921770-25-8.
Contains the proceedings of the Eleventh Australasian Symposium on Parallel and Distributed Computing (AusPDC 2013), Adelaide, Australia, 29 January – 1 February 2013.

Volume 141 - Theory of Computing 2013
Edited by Anthony Wirth, University of Melbourne, Australia. January 2013. 978-1-921770-26-5.
Contains the proceedings of the Nineteenth Computing: The Australasian Theory Symposium (CATS 2013), Adelaide, Australia, 29 January – 1 February 2013.

Volume 142 - Health Informatics and Knowledge Management 2013
Edited by Kathleen Gray, University of Melbourne, Australia and Andy Koronios, University of South Australia, Australia. January 2013. 978-1-921770-27-2.
Contains the proceedings of the Sixth Australasian Workshop on Health Informatics and Knowledge Management (HIKM 2013), Adelaide, Australia, 29 January – 1 February 2013.

Volume 143 - Conceptual Modelling 2013
Edited by Flavio Ferrarotti, Victoria University of Wellington, New Zealand and Georg Grossmann, University of South Australia, Australia. January 2013. 978-1-921770-28-9.
Contains the proceedings of the Ninth Asia-Pacific Conference on Conceptual Modelling (APCCM 2013), Adelaide, Australia, 29 January – 1 February 2013.

Volume 144 - The Web 2013
Edited by Helen Ashman, University of South Australia, Australia, Quan Z. Sheng, University of Adelaide, Australia and Andrew Trotman, University of Otago, New Zealand. January 2013. 978-1-921770-15-9.
Contains the proceedings of the First Australasian Web Conference (AWC 2013), Adelaide, Australia, 29 January – 1 February 2013.

Volume 145 - Australian System Safety Conference 2012
Edited by Tony Cant, Defence Science and Technology Organisation, Australia. April 2013. 978-1-921770-13-5.
Contains the proceedings of the Australian System Safety Conference (ASSC 2012), Brisbane, Australia, 23rd – 25th May 2012.