Conferences in Research and Practice in
Information Technology

Volume 146

Data Mining and Analytics 2013 (AusDM'13)

,

Data Mining and Analytics 2013

Proceedings of the
Eleventh Australasian Data Mining Conference
(AusDM'13), Canberra, Australia,
13 – 15 November 2013

Peter Christen, Paul Kennedy, Lin Liu, Kok-Leong Ong,
Andrew Stranieri and Yanchang Zhao, Eds.

Volume 146 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 2013. Proceedings of the Eleventh Australasian Data Mining Conference (AusDM'13), Canberra, Australia, 13 – 15 November 2013

Conferences in Research and Practice in Information Technology, Volume 146.

Editors:

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

Paul Kennedy
School of Software
Faculty of Engineering and Information Technology
University of Technology Sydney
P.O. Box 123, Broadway, NSW 2007, Australia
Email: Paul.Kennedy@uts.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

Andrew Stranieri
School of Engineering and Information Technology
Faculty of Science and Technology
Federation University
PO Box 663, Ballarat, Victoria 3353, Australia
Email: Paul.Kennedy@uts.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 146.
ISSN 1445-1336.
ISBN 978-1-921770-16-6.

Document engineering by CRPIT, November 2013.

Table of Contents

Message from the General Chairs

Dear AusDM authors, participants, and sponsors,

We would like to welcome you to the eleventh Australasian Data Mining and Analytics Conference (AusDM'13). We are proud to again host AusDM in Canberra, the birthplace, where the first AusDM was held as a workshop in 2002.

As in previous years, AusDM is again co-located with another conference in a complementary research area. This year, for the first time, we are co-located with the Asian Conference on Machine Learning (ACML), and we hope participants of both conferences will find this co-location inspiring.

We are excited to have two local keynote speaker of high calibre. Dr Paul Wong from the Australian National University's Office of Research Excellence will talk about how network analytics is used both by academia and governments to inform and direct research strategies and investments. Klaus Felsche from the Department of Immigration and Border Protection will enlighten us about what happens, when you next hop on an international flight back to Australia (or for our international visitors, what happened before you boarded your flight to Australia).

This year's conference for the first time had paper submissions in two categories, research and industry, and we are pleased to have received a good number of high quality industry submissions which show the breadth and depth of data mining and analytics that happens in Australian industry and government organisations. Of the 54 submitted papers, 20 will be presented. Of these 13 are research track papers and 7 are industry papers.

To enrich the practical aspects of our conference program, we are pleased to also offer a practical tutorial on the use of R and Rattle to be given by Dr Graham Williams, Director and Senior Data Miner at the Australian Taxation Office, and author of the Springer book Data Mining with Rattle and R.

There are many people and organisations who have supported this year's conference. We would like to 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 volunteers who support the running of the conference, and the organisers for ACML to support AusDM by sharing catering and venues.

There are many people and organisations who have supported this year's conference. We would like to 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 volunteers who support the running of the conference, and the organisers for ACML to support AusDM by sharing catering and venues.

Finally, we would like to thank our sponsors for their support: SAS (Platinum Sponsor), Oracle (Gold Sponsor), the Australian National University, the University of South Australia, Togaware, and the Australian Computer Society.

We hope you enjoy AusDM'13 and your stay in Canberra.

Yours Sincerely,

Peter Christen
The Australia National University, Canberra

Paul Kennedy
University of Technology, Sydney

November 2015

Message from the Program Chairs

Welcome to the 11th Australasian Data Mining and Analytics Conference, in Canberra, Australia.

A total of fifty four (54) papers were submitted to the two conference tracks (research and industry). From these, each paper was rigorously reviewed by at least two reviewers and up to a maximum of four reviewers took part in providing an assessment of the papers' merits. After careful consideration, twenty-three (23) papers were selected for inclusion in the final conference program, of which three unfortunately had to withdraw due to funding and security approval reasons.

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

Beyond the technical program, the conference has been enriched by many other items. These include the co-location with the 5th Asian Conference on Machine Learning and the availability of keynote speakers from both conferences. We trust these programmes will provide insightful new research ideas and directions.

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

Yours Sincerely,

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 2013

Conference Organisation

General Chairs

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

Program Chairs (Research)

Kok-Leong Ong, Deakin University
Lin Liu, University of South Australia

Program Chair (Industry)

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

Sponsorship Chair

Andrew Stranieri, University of Ballarat

Steering Committee Chairs

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

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

Program Committee

Industry Track

Chris Barnes, Australian Institute of Sport
Rohan Baxter, ATO
Neil Brittliff, Australian Crime Commission
Shane Butler, Telstra
Adriel Cheng, Defence Science and Technology Organization
Ross Farrelly, Teradata ANZ
Klaus Felsche, Department of Immigration and Border Protection
Richard Gao, Department of Agriculture, Fisheries and Forestry
Ross Gayler, Veda
Warwick Graco, ATO
Lifang Gu, ATO
Greg Hood, Department of Agriculture, Fisheries and Forestry
Yingsong Hu, Department of Human Services
Warren Jin, CSIRO
Edward Kang, Australian Customs and Border Protection Service
Luke Lake, Department of Immigration and Border Protection
Clinton Larson, DIGIVIZER
Jin Li, Geoscience
Ray Lindsay, ATO
Chao Luo, Department of Human Services
Kee Siong Ng, EMC Greenplum
Hong Ooi, Greenplum
Cecile Paris, CSIRO
Clifton Phua, SAS Institute
Wilson Pok, Westpac
Bill Ross, Australian Customs and Border Protection Service
Graham Williams, Togaware and ATO
Rory Winston, ANZ
Andrew Wyer, Department of Immigration and Border Protection
Debbie Zhang, ATO
Ke Zhang, Department of Health and Ageing
Sam Zhao, Department of Agriculture, Fisheries and Forestry

Research Track

Md Anisur Rahman, Charles Sturt University, Australia
Xiaohui Tao, The University of Southern Queensland, Australia
Paul Kwan, University of New England, Australia
Guandong Xu, University of Technology Sydney, Australia
Raj Gopalan, Curtin University, Australia
Yun Sing Koh, University of Auckland, Australia
Tom Osborn, Chief Scientist, Brandscreen; UTS (Adjunct Prof), Australia
Muhammad Marwan Muhammad Fuad, Norwegian University of Science and Technology
Gang Li, Deakin University, Australia
Adil Bagirov, University of Ballarat, Australia
Sitalakshmi Venkatraman, University of Ballarat, Australia
Xuan-Hong Dang, Aarhus University, Denmark
Francois Poulet, IRISA, France
Brad Malin, Vanderbilt University, The Nederlands
Ping Guo, Image Processing and Pattern Recognition Laboratory, Beijing Normal University
John Yearwood, University of Ballarat, Australia
Ting Yu, University of Sydney, Australia
Christine O'Keefe, CSIRO Mathematics, Informatics and Statistics, Australia
Huizhi Liang, The Australian National University
Md Zahidul Islam, Charles Sturt University, Australia
Siddhivinayak Kulkarni, University of Ballarat, Australia
Peter Vamplew, University of Ballarat, Australia

Yee Ling Boo, Deakin University, Australia
Bing Liu, University of New South Wales, Australia
Sumith Matharage, Deakin University, Australia
Bing-Yu Sun, University of Science and Technology, China
Mengjie Zhang, Victoria University of Wellington, New Zealand
Shafiq Alam, University of Auckland, New Zealand
Robert Layton, University of Ballarat, Australia
Michael Hecker, Curtin University, Australia
Russel Pears, Auckland University of Technology, New Zealand

Additional Reviewers, Research Track

Yongli Ren
Zongda Wu
Md. Nasim Adnan
Liang Hu
Tianqing Zhu
Huy Quan Vu
Fangfang Li
Jian Yu
Yang Sun
Md. Geaur Rahman

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://http://www.sas.com

University of South Australia

http://www.unisa.edu.au/

http://www.oracle.com/au/index.html/

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

Keynotes

Predictive Network Analytics for National Research Investment

Paul Wong

Office of Research Excellence, Research Services
HC Coombs Policy Forum, Crawford School of Public Policy
The Australian National University
Canberra ACT 0200
Email: paul.wong@anu.edu.au

Abstract

Research is a risky business. The starting point of research is ignorance: if we already have answers to our questions or simply undertaking routine works to get answers, we wouldn't be undertaking research in the first instance.

Australia spends approximately 2.2% GDP (or $27.7B AUD) in research and development. So we are taking considerable risks as a country. Fortunately, some research areas are less risky than others. They have well-established theoretical foundations and experimental methodologies, proper access to infrastructures and equipment, and above all a critical mass of researchers to advance the state of knowledge. In "emerging" areas of research however the risks are considerably higher - there may not be an established theory, methodology, or even a critical mass of researchers available.

Finding the right approach to fund emerging research is a serious policy challenge. The European Research Council and the National Science Foundation (in the U.S.) have both independently initiated works in developing approaches to identify and fund emerging research in recent years. If we accept the suggestion that research investment is akin to portfolio investment (to maximize the expected return while minimize risk over an entire investment portfolio), then investing in emerging research amounts to investing in high risk options with high expected return. But how do we pick "winners" from "imposers"? How can we tell we are not picking "one hit wonders"? How can we spot "sleeping beauties" which may take years to mature? The availability of large scale global bibliographic (and other relevant) data from both open and commercial sources presents an intriguing opportunity for data miners and machine learners to contribute to these debates.

In this presentation, we shall examine the general shape of the problem definition - to look at the "why" and "what" instead of the "how". Our aim is to present and engage the Australasian data mining and machine learning communities in a conversation about an intellectually challenging and exciting problem that can have wide spread impact on how governments, funding agencies and industries make strategic decisions in R&D investment.

Harnessing the Power of Data in Government through Analytics

Klaus Felsche

Intent Management and Analytics
Department of Immigration and Border Protection
Belconnen ACT 2617
Email: klaus.felsche@immi.gov.au

Abstract

The Australian government collects and uses a large volume of data from its clients. Much of these data are used to facilitate the provision of services. Data custodians maintain the integrity of records as they record entitlements, decisions and liabilities. Strict regimes exist to protect privacy, commercial sensitivities and potential misuse of such data. Much of these data holdings are in data-stovepipes. While new project management practices (such as Agile and Lean Startup) are gaining support in the private sector, there are few instances of these being applied in government. This environment makes the job of teams seeking to leverage new and emerging analytics capabilities challenging. New and innovative approaches are needed to maximise the value of analytics capabilities. This presentation will offer some thoughts on approaches that can deliver positive outcomes based on recent developments in the Department of Immigration and Border Protection. The departments experience demonstrates that large organisation can develop and deploy affordable, new analytics capabilities into its core systems infrastructure in a relatively short period.

Contributed Papers

Extraction of Essential Region in Gastric Area for Diagnosing Gastric Cancer Using Double Contrast X-ray Images

Koji Abe [1] **Hideaki Nakagawa** [1] **Masahide Minami** [2] **Haiyan Tian** [3]

[1] Interdisciplinary Graduate School of Science and Engineering
Kinki University
3-4-1 Kowakae, Higashi-Osaka, Osaka 577-8502, Japan
Email: koji@info.kindai.ac.jp, nakagawa0623nara@gmail.com

[2] Graduate School of Medicine
the University of Tokyo
7-3-1, Hongo, Bunkyo-ku, Tokyo 113-0033, Japan
Email: maminami@dream.com

[3] Ministry of Education
Chongqing University
Chongqing 400044, China

Abstract

In a mass screening for gastric cancer, diagnosticians currently read several hundred stomach X-ray pictures at a time. To lessen the number of reading the pictures or to inform lesions to diagnosticians, computer-aided diagnosis systems for the cancer have been proposed. However, in every system, the gastric area or its part for the diagnosis has been manually extracted in extracting characteristics of figure patterns in the area. To deign full automatic computer-aided diagnosis for the cancer, this paper proposes a method for extracting the essential region in the gastric area of double contrast X-ray images. In the proposed method, considering characteristics of density distributions around objects of the barium-pool, the spinal column, and edge of the gastric area, the essential region is fixed by recognizing the bottom of the barium-pool, the right side line of spinal column, and the edge from right side to bottom of gastric area. Experimental results for the proposed method by conducting an existing system of discriminating normal and abnormal cases using 43 images including 11 abnormal cases have shown that there is no significant difference between both of the results by the existing system which extracts the region manually and with the proposal.

Keywords: medical image processing, computer-aided diagnosis, X-ray image, gastric cancer.

1 Introduction

In mass screenings for gastric cancer, due to financial reasons, double contrast X-ray pictures are generally used on behalf of CT, MRI or photogastroscope. In the screenings, diagnosticians always need hard labor because they read several hundred X-ray pictures at a time. Besides, since accuracy of the reading is strongly depended on experience of diagnosticians, it is hard for inexperienced doctors to read them well. Especially, importance of accurate reading and its education have been increased in recent years from activity the Japanese Government exports the studio car which equips the camera for taking stomach X-ray pictures to Asian countries. For the reasons, computer-aided diagnosis (CAD) systems for gastric cancer in X-ray images have been required as a second opinion for diagnosticians (Kita 1996, Maeda et al. 1998, Hasegawa et al. 1991, 1992, Yoshinaga et al. 1999). In addition, a CAD system for discriminating normal stomachs to lessen the number of the readings has been reported (Abe et al. 2011). In every of the systems, characteristics of normal cases or lesions are extracted as features by analyzing figure patterns of the gastric area or a part of the area in double contrast X-ray images. However, although every system needs to extract gastric area or region of interests (ROI), all of them extract the areas manually. The reason why the systems avoid recognizing the areas would be because it is very difficult to extract only the areas from the stomach X-ray images, where shades of several 3D objects such as ribs, the spinal column, barium-pools overlap each other on 2D space.

For the sake of the essential pre-processing in CAD systems of gastric cancer using double contrast stomach X-ray images, this paper proposes a method for extracting the essential region for the diagnosis in the gastric area. When diagnosticians read the images, they check the pattern of fold shades appeared in the gastric area. Hence, the essential region should at least include the area where the shades are appeared and results of diagnosis by the CAD systems which equip the proposed pre-processing should be very similar to ones without it.

In this paper, after the essential region is defined, the proposed method for extracting the region is presented. Then, performance of the proposed method is examined by comparison between discrimination re-

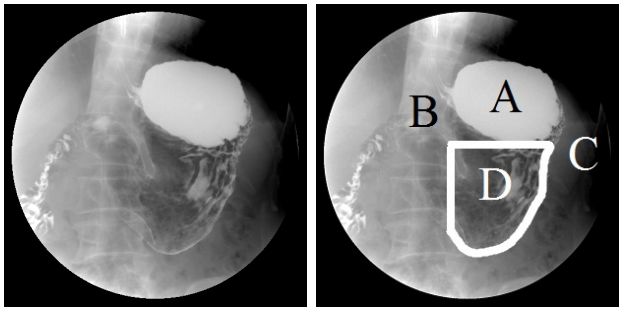


Figure 1: A double contrast stomach X-ray image and the essential region for the diagnosis (the area "D").

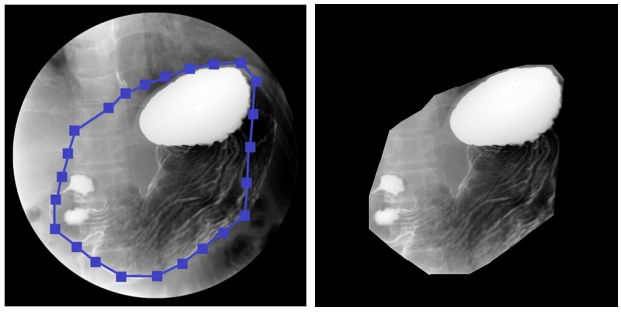


Figure 2: A result of the gastric area extracted by the snake algorithm.

sults of normal cases using the CAD system (Abe et al. 2011) and ones using the same system which equips the proposed method.

2 Double Contrast Stomach X-ray Images and the Essential Region for the Diagnosis of Gastric Cancer

In radiography to obtain contrast stomach X-ray pictures, participants drink barium and radiologists take eight X-ray pictures changing direction of their body. Diagnosticians first choose the head-on double contrast X-ray picture shown in Figure 1 among the eight pictures and diagnose the stomach reading it. When it is difficult to diagnose the stomach due to an uncertain or a doubtful case, the other pictures are used as the second material. Therefore, all the CAD systems of diagnosing stomach diagnose the head-on double contrast X-ray images. Hence, in this paper, the proposed method extracts the essential area for the diagnosis from the images as well.

The right image in Figure 1 is the copy of the left one, where "A" is the barium pool, "B" is the spinal column, "C" is the contour curve from the right side to the bottom of the gastric area, and "D" is the essential region. In the diagnosis, diagnosticians read the pattern of folds which mirror the shade of gastric rugae in the area D, which is located at the right side of B and under A in the images. Therefore, the essential region D is defined as the area which is enclosed by the right side line of B, the horizontal line through the bottom of the area A, and the curve C.

The active contour model (called as *Snakes*) (Kass et al. 1988) is often applied as a means for extracting the contour of an object (Fukushima et al. 2000). Figure 2 shows the contour of the gastric area extracted by applying the snake algorithm to one of the X-ray image, where the left shows a result of the contour recognition for the gastric area and the right shows the result of extracted gastric area. As shown in Figure 2 (right), the barium pool and a part of the spinal column have been included in the extracted area. Besides, some control points have invaded into the gastric area and the contour has chipped the right side part of the area. In radiography, since the human body, which is a 3D object, is mirrored into 2D space, 3D multi-organs are appeared to overlap each other in the 2D X-ray picture. Due to the overlaps, it is very hard for the snake algorithm to catch the correct contour of gastric area in the X-ray images.

Thus, if the target area for the diagnosis lacks roughly or includes other objects, the CAD systems could not output correct results. Therefore, it is necessary to propose another way for extracting the essential region which does not include other organs and keeps the pattern of folds appeared in gastric area.

Throughout this paper, all the double contrast X-ray pictures are digitalized as images by CR (Computed Radiography). The size of the images is 1024 $\times$ 1024 pixels with 256 gray levels.

3 Proposed Method

3.1 Extraction of the barium pool

In order to obtain the head location of the essential region, the barium pool is extracted by the following processing. In the X-ray images, pixel values of the barium pool are certainly between 150 and 255, and the contour of barium pool appears remarkably. Considering these characteristics, first, all the pixels whose value is no more than 150 are converted into 0 (i.e., their color becomes the complete black.). Second, the smoothing processes by the moving average and the selective local averaging (Rosenfeld et al. 1982), edge enhancement by the Kirsh filtering (Parker 2010) are applied to the image in order. Third, the image is binarized into the black and white image by discriminant analysis and the thinning is applied to the image. Since the barium pool is basically located at the upper right in the image, the line segments obtained by the thinning whose part is not included in the quarter area at the upper right side in the image are removed. Figure 3(2) shows the image obtained by these processes from the original image of Figure 3(1).

Next, all the pixels in the quarter area at the upper right side of the original image are binarized by the threshold of pixel value 200. Figure 3(3) shows the cetroid point in the area extracted from Figure 3(1) and it is regarded as the central point of the barium pool. In radiating a straight line from the central point, the nearest intersection between the line and one of the line segments (shown in Figure 3(2)) is obtained as shown in Figure 3(4). The straight line radiates from the point at intervals of an angle of 7.5 degree, i.e., the number of the extracted intersections becomes 48 points. The contour of the barium pool

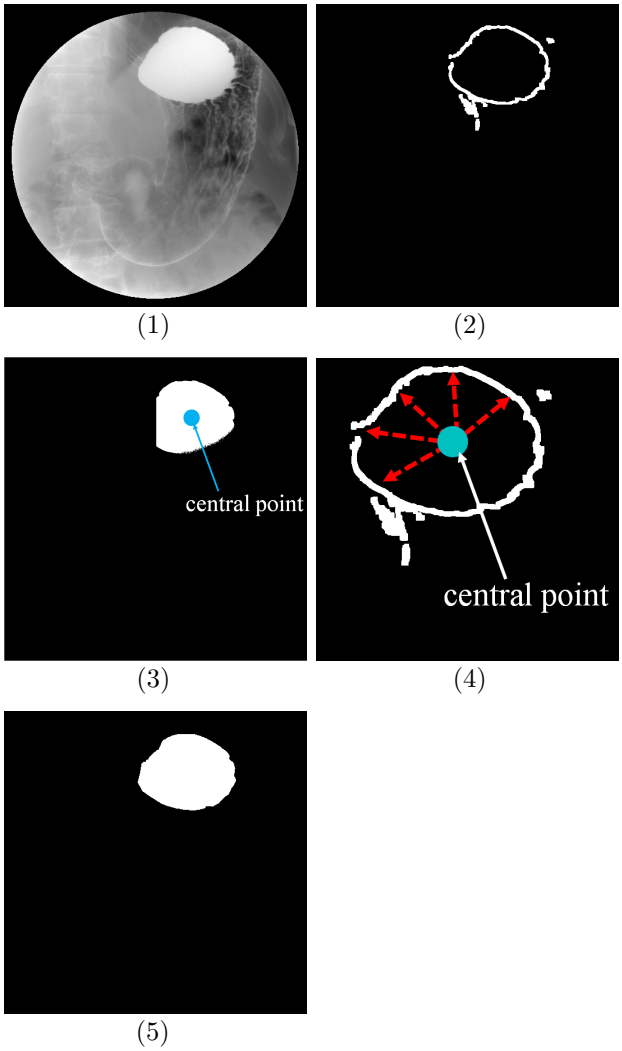


Figure 3: Processes for extracting the barium pool ((1)original image (2)line segments obtained as candidates of the contour of the barium pool (3)central point of the barium pool (4)the radiation of lines from the central point (5)extracted barium pool).

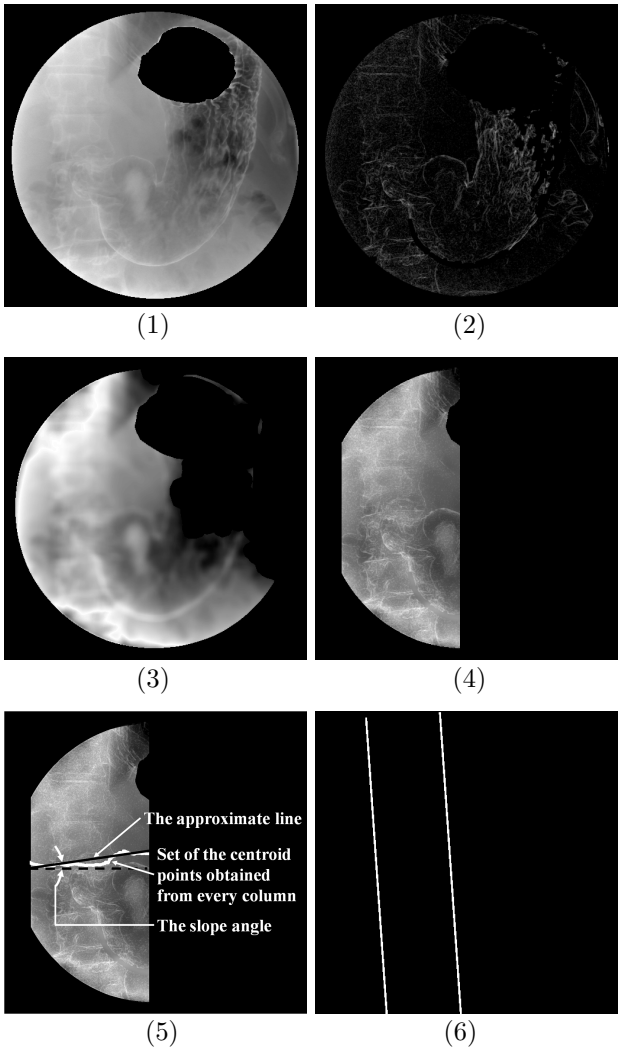


Figure 4: Processes for extracting the spinal column ((1)removal of the barium pool (2)image 1 (3)image 2 (4)candidate area for the spinal column (5)the slope angle of the spinal column (6)both sides of the spinal column).

is obtained by connecting neighbor intersections with the line segment between the intersections. Figure 3(5) shows area of the barium pool obtained from the contour.

3.2 Extraction of the spinal column

Before extracting the spinal column in the image, the area of barium pool is converted into black area in the image as shown in Figure 4(1).

Since difference of density between the spinal column and its background is not clear, Sobel filtering is applied to the image (Figure 4(1)) in order to enhance edge of the spinal column. Then, to reduce noises except the spinal column, pixels whose value is more then 50 and others which are located within a 3-pixels radius from them are removed. Figure 4(2) shows the result of this processing for Figure 4(1) and this image is called as *image 1* (Figure 4(2) is represented by threefold density of the actual one.). The frequency of folds' pattern appeared in the gastric area is higher than the spinal column. Hence, high-frequency components are removed from the image shown in Figure 4(1) by the band-pass filter of 3 $\sim$ 255-radius after Fourier transformation and the pixel value of all the pixels is converted into twice after inverse Fourier transformation. Figure 4(3) is the result of this processing for Figure 4(1) and the image is called as *image 2*.

Next, two of the image 2 is superposed on and the image 1 besides the total value of all the pixel values on each column of the superposed image is calculated in every column. Since pixel values are hold down except the spinal column, the total value should be significantly high at columns in the spinal column. Besides, the width of the spinal column is around 250. Hence, the column which has the largest total value is found and columns which are horizontally more than 250 away from the column are removed. Figure 4(4) shows the result of this processing for Figure 4(1) and this area is regarded as *candidate area for the spinal column*. Using pixels whose value is more than 100 in each column of the candidate area for the spinal column, g_{min} is found for obtaining the minimum value

$f(g_{min})$ of $f(g)$ shown in Eq.(1) and g_{min} is found from every column of the candidate area as the y-coordinate of the centroid point in the column.

$$f(g) = |\sum_{y=0}^{1023} f_y \times (g - y)| \quad (1)$$

where y is y-coordinate of a pixel whose value is more than 100 in a column, and f_y is its value. After that, an approximate line is obtained using the pixels obtained from each column. The angle between the approximate line and the horizontal line is regarded as the slope angle of the spinal column. Figure 4(5) shows the angle.

Finally, among all the columns in the candidate area, only higher rank half of them, which have larger total value of all the pixel values in the column, are chosen as the spinal column. Here, among the half, independent columns are removed because it would be impossible that an independent line forms the spinal column. The bundle of the chosen columns is rotated at the slope angle around the central point of the whole image. Figure 4(6) shows the two lines at both sides of the bundle obtained from Figure 4(1), and the area between them is regarded as the spinal column.

3.3 Extraction of the contour from the right to the bottom of the gastric area

Since the gastric contour in the X-ray images is not appeared clearly in most of cases, Sobel filtering is applied to the original image. And, since the left side area of the right side line of the spinal column extracted in 3.2 is not necessary for the diagnosis, the area is removed.

The shape of gastric area is always changing, hence the gastric contour is not uniform in the stomach. Although it would be difficult to catch characteristics of the shape, the contour has characteristics that the contour is clearer than shades of bowel and density of the contour is higher than its background in the X-ray images. Considering the characteristics, after the processes shown above are conducted to the image, the image is binarized according to the following three conditions:

[Conditions for the binarization of the contour]

1. the pixel whose value is more than 35 and difference of pixel value between the pixel and the first or second nearest right and left pixels from the pixel is more than 5
2. the pixel whose value is more than 35 and difference of pixel value between the pixel and the first or second nearest upper and lower pixels from the pixel is more than 5
3. the pixel whose value is more than 35 and difference of pixel value between the pixel and the nearest four pixels except pixels connected by the 4-neighbor connectivity among 8-neighborhoods of the pixel

This binarization produces one image to each condition (i.e., it produces three images.). After that, noises are removed by the combination of dilatation and erosion and the removal of isolated points in each of the three images, and the three images are converted into one image by OR composition. Figure 5 shows the results of the binarization to the image shown in Figure 3(1), where (1), (2), and (3) are the images binarized by the first, second and third conditions, respectively. And, Figure 5(4) shows the composite image of Figure 5(1), (2), and (3).

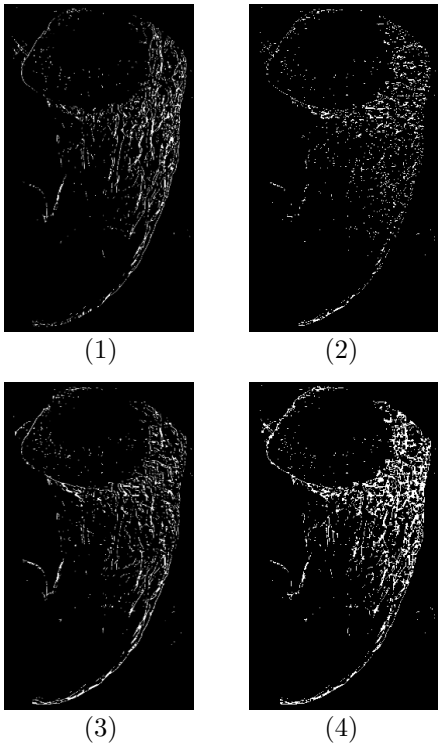


Figure 5: The binarization according to the conditions ((4) is the final output in the binarization.).

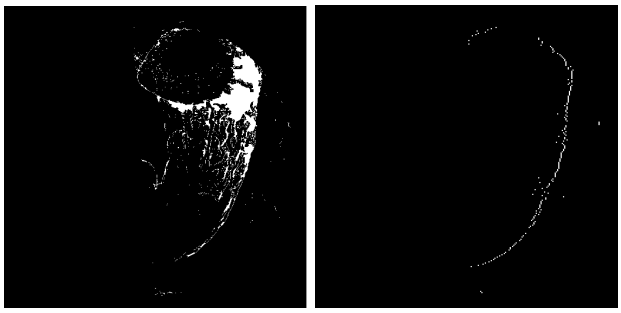
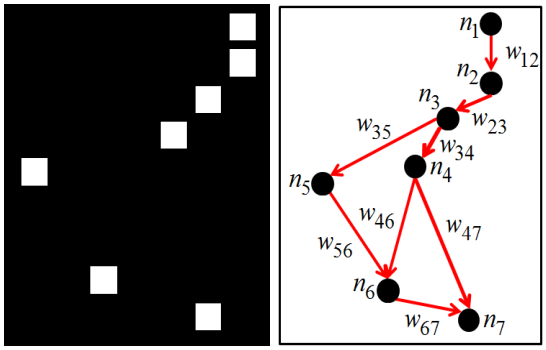
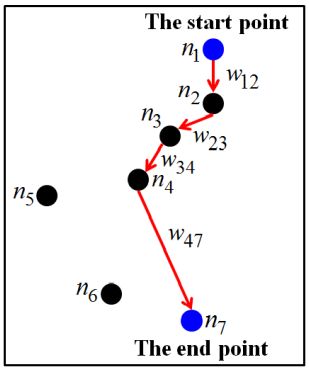


Figure 6: The process for obtaining the candidate points on the contour of the gastric area.

The contour curve from the right side to the bottom of the gastric area is extracted from the binary image obtained above as follows. First, all of closed black areas enclosed the white pixels are converted into the white color. Figure 6(1) shows the result of this processing for Figure 5(4). And then, dividing each of the vertical and horizontal sides into 256 in the image, a set of 4 × 4 pixels is regarded as 1 block, i.e., an image of 1024 × 1024 pixels is equal to 256 × 256 blocks. Second, scanning every block row



(1) (2)



(3)

Figure 7: The interpolation of the contour candidate by the DAG.

horizontally in the image, the right end among the blocks which at least have 6 white pixels in every row is obtained. Among the obtained blocks, if upper end of the block is lower than the horizontal line which includes the bottom of the extracted barium pool, the upper right pixel in the block is extracted as a candidate point p on the contour. Figure 6(2) shows the set of the candidate points extracted from Figure 6(1). And, if a couple of the points (p_i and p_j) is satisfied with Eq.(2), the couple is connected by a line segment where the points become the line ends; and the line segment has a weight w_{ij}, which is the distance between the two points. Conducting it to every combination of the couples, directed acyclic graphs (DAGs) (Jungnickel 2013) are obtained to the image. In the DAGs, the points and the line segments are regarded as nodes and directed arcs, respectively, besides all the arcs are directed downward in the graph. Figure 7(1) shows a part of Figure 6(2) and Figure 7(2) shows a DAG obtained from Figure 7(1). Choosing only the DAG which has the number of the nodes most in all the DAGs, regarding the node which is at the highest location in the image as the start point and the node which is at the lowest location as the end point, the shortest route from the start point to the end point is obtained (Jungnickel 2013) as shown in Figure 7(3), and the track of the route is regarded as the candidate of the contour. Figure 8 shows the extracted candidate of the contour from Figure 6(2), where the left is the candidate and the right is the image which places the left on its original image.

$$|d_x(i,j)| \leq 5 \quad \wedge \quad S(i,j) \leq 30 \qquad (2)$$

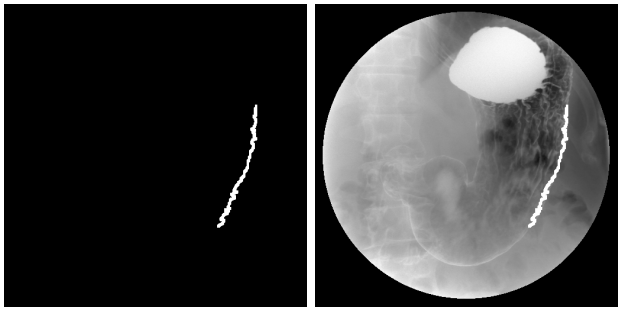
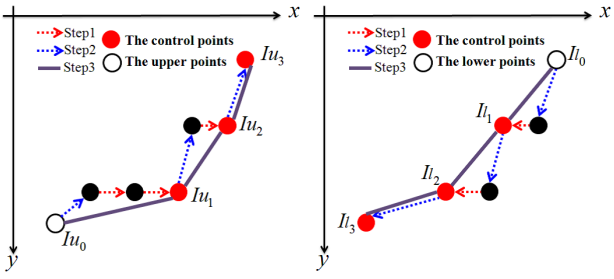


Figure 8: The candidate of the contour extracted from Figure 6(2).



(1) (2)

Figure 9: The tracking of the contour ((1)for the upper direction (2)for the lower direction).

where $d_x(i,j)$ and $S(i,j)$ are difference of x-coordinates and distance between p_i and p_j, respectively, i.e., $S(i,j)$ is $w(i,j)$ when a couple of p_i and p_j is satisfied with Eq.(2).

Next, individual white areas which are no more than 15 pixels are removed from the image that all of closed black areas enclosed the white pixels are converted into the white color obtained above (Figure 6(1)). The obtained image is called as *reference image*. Referring to the reference image, the contour is finally obtained by connecting n control points according to the tracking way shown below. Here, $I_u[i]$ and $I_l[i]$ are the i-th control point from the upper end $I_u[0]$ and the lower end $I_l[0]$ in the candidate of the contour, respectively. And then, n_u and n_l are the number of the control points obtained from $I_u[0]$ and $I_l[0]$, respectively (i.e., $n = n_u + n_l$).

[Flow of the interpolation]

1. The white pixels which are located at upper area from $I_u[i]$ (or, at lower area from $I_l[i]$) and satisfied with Eq.(2) in partnering $I_u[i]$ (or, $I_l[i]$) are searched in the reference image. Among them, the nearest pixel from $I_u[i]$ is set as $I_u[i+1]$ ($I_l[i+1]$ is set to the nearest pixel vertically and the farthest pixel horizontally from $I_l[i]$). If the suitable pixel is nothing, go to 3.

2. The white pixels which have y-coordinate of $I_u[i+1]$ (or, $I_l[i+1]$) and are satisfied with Eq.(2) in partnering $I_u[i+1]$ (or, $I_l[i+1]$) are searched in the reference image. $I_u[i+1]$ is moved to the pixel which has the largest x-coordinate. $I_l[i+1]$ is moved to the pixel which has the smallest x-coordinate. If nothing, go back to 1. increasing 1 to i.

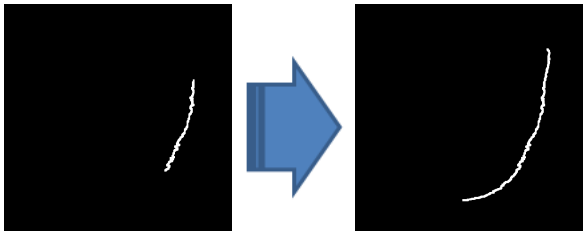
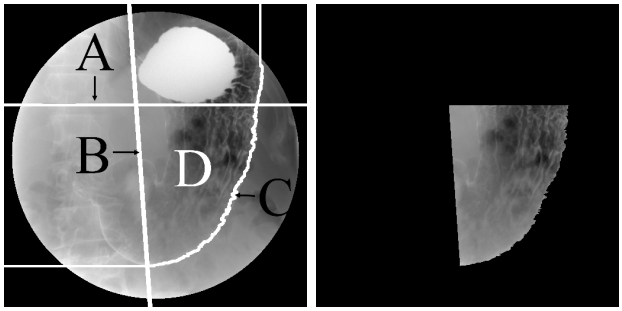


Figure 10: Extraction of the contour from the right side to the bottom of the gastric area (left: the candidate of the contour (Figure 8), right: the final output of the contour).



(1) (2)

Figure 11: Extraction of the essential region ((1)composition of the extracted objects (2)the extracted essential region).

3. Connect between $I_u[i]$ and $I_u[i+1]$ (or, $I_l[i]$ and $I_l[i+1]$) with a line. And, repeat it from $i = 0$ to $i = n_u$ (to $i = n_l$ in the case of $I_l[i]$).

As an example of the tracking, Figure 9 shows the tracking, where (1) shows the tracking from $I_u[0]$ to $I_u[3]$ and (2) shows from $I_l[0]$ to $I_l[3]$. Resulting this tracking, the contour is obtained finally. Figure 10 (right) shows the result of the contour obtained from the contour candidate for Figure 4(1).

3.4 Extraction of the Essential Region for the Diagnosis

The vertical line is drawn upward from the upper end of the contour extracted in 3.3. And, the horizontal line is drawn leftward from the lower end. The essential region is obtained by enclosing the horizontal line which includes the bottom point of the extracted barium pool obtained in 3.1 (A in Figure 11(1)), the right side line of the spinal column obtained in 3.2 (B in Figure 11(1)), and the curve which is the contour obtained in 3.3 plus the lines drawn here (C in Figure 11(1)). D in Figure 11(1) shows the essential region for the diagnosis. And, Figure 11(2) shows the final output of the essential region.

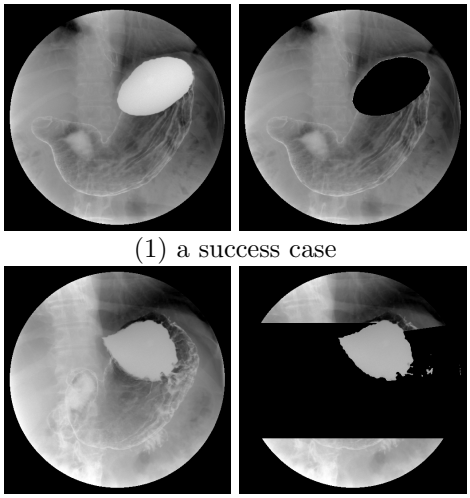
4 Experimental Results

4.1 The Extraction of the Essential Region

The proposed extractions for the three objects of 1)barium pool, 2)spinal column, and 3)the contour from the right side to the bottom of gastric area are applied in turn to 43 double contrast X-ray stomach images (32 normal cases, and 11 abnormal cases: all the abnormal cases had been diagnosed as gastric cancer by a medical doctor.). Performance of the extractions is evaluated by a medical examiner's eyes. After that, the essential region is extracted to only the images which got the extraction success in all the extractions of the three objects.

Table 1: Experimental results of the extractions for the three objects.

	success	failure
barium pool	42 cases (97.67%)	1 case
spinal column	42 cases (97.67%)	1 case
the outline	41 cases (95.35%)	2 cases



(1) a success case

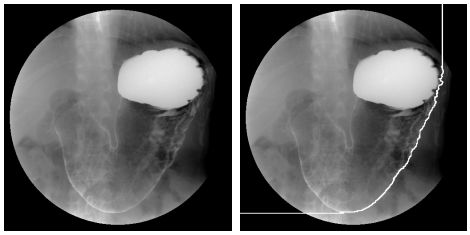
(2) the only failure case

Figure 12: Sample of results in the extraction of the barium pool.

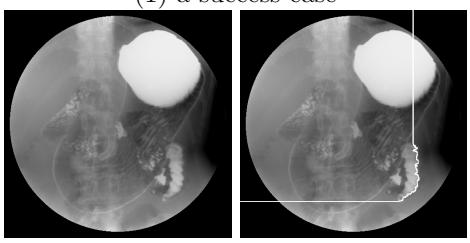
Table 1 shows experimental results of the extractions for the three objects. From the fact that the success ratios show more than 95 % in all the extractions, experimental results shown in Table 1 show that the proposed method has high performance in each of the extractions.

In the extraction of barium pool, according to the criterion of extraction success that at least 90 % of the pixels in the extracted area by the proposed method is included in the correct barium pool obtained by hands, only a case was failure. All the correct barium pools have been obtained by a nondoctor. Figure 12 (1) and (2) show a case of the extraction success and failure, respectively. The reason of the failure is because the central point was out of the correct area of barium pool. In this case, density of the barium pool was lower than the other X-ray images because some of barium ran into the intestine and barium decreased in the gastric area. Receiving this failure, it would be necessary to determine the threshold of the binarization considering the mean of density in the barium pool.

In the extraction of the spinal column, according to the criterion of extraction success that the extracted area includes at least part of the spinal column, only a case was failure and this failure case was



(1) a success case



(2) one of the two failure cases

Figure 13: Sample of results in the contour extraction.

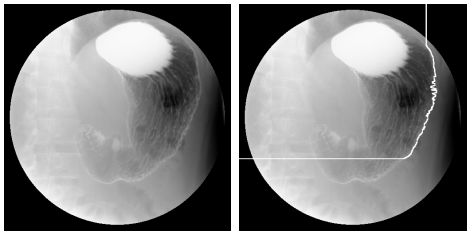
the same as the case which was the failure in the extraction of barium pool. The reason of the failure is because the barium pool was not extracted precisely. Thus, the extraction of the spinal column is depended on the location of the barium pool.

In the extraction of the contour, according to the criteria of extraction success that the extracted contour is put on the correct contour and extraction failure that the extracted contour is completely out of the actual contour or the fold pattern is remarkably lacked due to too short contour, two cases were failure. Figure 13 (1) and (2) show a case of the extraction success and failure, respectively. In Figure 13(1), we can confirm the extracted contour is completely put on the actual contour from the right side to the bottom of the gastric area. On the other hand, in Figure 13(2), we can confirm the tracking error has been occurred. The reason for the failure case is because a mass of barium appeared on the contour located at the lower right has been tracked on behalf of the correct contour. And, since the extraction of the contour is depended on the locations of the barium pool and the spinal column, the image shown in Figure 12(2) got the failure in the contour extraction as well.

In 16 cases among the 41 cases which got success in all the extractions, the contour was a little shorter than desirable length. Figure 14 shows a case of them. However, since every of their essential regions includes fold patterns for the diagnosis enough, they were regarded as the extraction success. The reason why their contour has been shorter is because the noise removal in making the reference image has removed part of indispensable area to track the contour. Therefore, it is necessary to improve enhance the extraction of the candidate points on the contour.

4.2 The Discrimination of Normal and Abnormal Cases for the Essential Regions

By using the CAD system of discriminating gastric cancer (Abe et al. 2011), performance of the essential regions extracted by the proposal are compared with



(1) (2)

Figure 14: A case when the extracted contour was shorter than desirable length ((1)original image (2)result of the contour extraction).

essential regions extracted by hands (used in Ref.(Abe et al. 2011)). All the essential regions extracted by hands have been obtained by a nondoctor. In the CAD system, image features proposed in Ref.(Abe et al. 2011) are extracted from the essential region and the discrimination is conducted by discrimination machines regarding the features as variables. As discrimination machines, linear discriminant analysis (LDA), the discriminant analysis by Mahalanobis' distance (MD), linear support vector machines (SVM) are applied for the discrimination. Before the image features for the diagnosis are extracted, the system extracts the folds in the essential region. The folds and the features are extracted as follows (Refer to (Abe et al. 2011) if necessary.).

The essential region is empirically binarized as follows. First, differences between every pixel value and each value of its 5 neighbors to each of right and left sides are measured, respectively. If the minimum difference among them is more than 16, the pixel is regarded as a pixel on a fold and its value is converted into 255. Otherwise, the value is into 0. Second, the binarization is conducted to every pixel in the vertical direction as well changing "5 neighbors" into "3 neighbors", and "right and left" into "upper and lower sides". Third, if the pixel value is 255 in one of the binarization images at least, the pixel value is fixed as 255. Otherwise, the value is 0. Finally, by conducting the thinning, the folds are extracted at last.

The features of parallelism f_1 and f_2 are extracted from the binary image of the folds as follows.

1) f_1 is defined as the number of pixels which have connection to at least three neighbors in 8-neighbor.
2) Removing all the points extracted in 1) from the binary image of the folds, all the folds are decomposed into line and curve segments. And then, one of the 8-direction codes is attached to every pixel on the folds in the image. The direction code d ($1 \leq d \leq 8$) is assigned to the angle of $(d-1) \times \pi/4$ rotating counterclockwise from the horizontal direction from left to right. f_2 is defined as $f_2 = sum(max_1 + max_2)$, where max_1 is the number of pixels which have the most code in the image, max_2 is the number of pixels which have the second most code, and sum is the number of pixels which have the other codes.

Performance of the discrimination is represented

Table 2: Experimental results of the diagnosis for the essential region.

	normal		abnormal	
tool	*Recall*	*Pecision*	*Recall*	*Precision*
LDA	90.32% (28/31)	93.33% (28/30)	80.00% (8/10)	72.73% (8/11)
MD	67.74% (21/31)	91.30% (21/23)	80.00% (8/10)	44.44% (8/18)
SVM	87.10% (27/31)	93.10% (27/29)	80.00% (8/10)	66.67% (8/12)

Table 3: Experimental results of the diagnosis for the area extracted by hand.

	normal		abnormal	
tool	*Recall*	*Pecision*	*Recall*	*Precision*
LDA	90.32% (28/31)	93.33% (28/30)	80.00% (8/10)	72.73% (8/11)
MD	80.65% (25/31)	92.59% (25/27)	80.00% (8/10)	57.14% (8/14)
SVM	87.10% (27/31)	93.10% (27/29)	80.00% (8/10)	66.67% (8/12)

by ratios of *Recall* and *Precision* defined as

$$Precision = \frac{|X_h \cap X_c|}{|X_c|} \times 100 \quad (3)$$

$$Recall = \frac{|X_h \cap X_c|}{|X_h|} \times 100 \quad (4)$$

where X_h is the set of the correct answers, X_c is a set of images discriminated by the proposed method, and $|X|$ is the number of images of a set X.

In these experiments, the 41 images (normal case: 31, abnormal case: 10) which were extraction success in all the three extractions have been used. Table 2 and Table 3 show discrimination results for the essential regions by the proposal and by hands, respectively, where the numbers in parentheses are the number of images used to calculate the ratios. Both of the tables show that there is no significant difference between results of the discrimination using essential regions extracted by the proposal and by hands. Therefore, we could confirm that the extraction of the essential region by the proposed method would be efficient enough in the diagnosis for mass screening of gastric cancer.

5 Conclusions

To design a computer-aided diagnosis for gastric cancer, this paper has presented a method for extracting the essential region in the gastric area of double contrast X-ray images. In the proposed method, the barium pool, the spinal column, and the contour from the right side to the bottom of the gastric area have been extracted. The essential region is fixed by connecting the two lines extracted from the barium pool and the spinal column and by the contour curve of the gastric area. Experimental results for the proposed method by the existing system of discriminating normal and abnormal cases have shown that there is no significant difference between both of the results by the existing system which extracts the region manually and by the proposal.

In the case when the angular incisur is included in the essential area, there is possibility that it becomes a noise and the CAD system leads to a discrimination error. As future works, it could be considered to improve the extractions of the central point in barium pools and the candidate points used to track the contour of the gastric area in addition to tracking the contour of the angular incisurs.

References

Y. Kita (1996), 'Elastic-model Driven Analysis of Several Views of a Deformable Cylindrical Object', *IEEE Trans. PAMI*, **18**(12), 1150–1162.

Y. Mekada, J. Hasegawa, J. Toriwaki, S. Nawano, and K. Miyagawa (1998), 'Automated Extraction of Cancer Lesions from Double Contrast X-ray Images of Stomach', *Proc. 1st International Workshop on Computer Aided Diagnosis*, Chicago, USA, 407–412.

J. Hasegawa, T. Tsutsui, and J. Toriwaki (1991), 'Automated Extraction of Cancer Lesions with Convergent Fold Patterns in Double Contrast X-ray Images of the Stomach', *Systems and Computers in Japan*, **22**(7), 51–62.

J. Hasegawa and J. Toriwaki (1992), 'A New Filter for Feature Extraction of Line Pattern Texture with Application to Cancer Detection', *Proc. 11th IAPR Int. Conf. on Pattern Recognition*, Hague, Netherlands, 352–355.

Y.Yoshinaga, H.Kobatake, and S. Fukushima (1999), 'The Detection and Feature Extraction Method of Curvilinear Convex Regions with Weak Contrast Using a Gradient Distribution Method', *Proc. ICIP 99*, Kobe, Japan, 715–719.

K. Abe, T. Nobuoka, and M. Minami (2011), 'Computer-Aided Diagnosis of Mass Screenings for Gastric Cancer Using Double Contrast X-ray Images', *Proc. IEEE Pacific Rim Conf on Communications, Computers and Signal Processing*, Victoria, Canada, 708–713.

M. Kass, A. Witkin, and D. Terzopoulos (1988), 'Snakes: Active contour models', *International J. of Computer Vision*, **1**(3), 321–331.

S. Fukushima, H. Uwai, and K. Yoshimoto (2000), 'Optimization-Based Recognition the Gastric Region from a Double-Contrast Radiogram', *IEICE Trans. (Japanese Edit.)*, **J83-D-II**(1), 154–164.

J.R.Parker (2010), 'Algorithms for Image Processing and Computer Vision', *Wiley Publishing*, 36–38.

A.Rosenfeld and A.C.Kak (1982), 'Digital Picture Processing, Second Edition, Volume 1', *Academic Press*.

D.Jungnickel (2013), 'Graphs, Networks and Algorithms', *Springer*, 49–52.

Features for Measuring the Congestive Extent of Internal Hemorrhoids in Endoscopic Images

Koji Abe [1] **Hidenori Takagi** [1] **Masahide Minami** [2] **Haiyan Tian** [3]

[1] Interdisciplinary Graduate School of Science and Engineering
Kinki University
3-4-1 Kowakae, Higashi-Osaka, Osaka 577-8502, Japan
Email: koji@info.kindai.ac.jp, tkghdnr2@gmail.com

[2] Graduate School of Medicine
The University of Tokyo
7-3-1, Hongo, Bunkyo-ku, Tokyo 113-0033, Japan
Email: maminami@dream.com

[3] Ministry of Education
Chongqing University
Chongqing 400044, China

Abstract

This paper presents a computer-aided diagnosis for internal hemorrhoids based on the congestive extent in endoscopic images. This system could be effective for young or even general practitioners as a second opinion because diagnosis of hemorrhoids needs much experience. Since the images are not always clear depending on situation of scenes, a pre-processing is conducted to the images to enhance saturation and contrast of congestive regions and blood vessels. Next, from characteristics of internal hemorrhoids, the proposed method measures degree of the congestion in the images and extracts the congestive region and features on the congestive extent. Experimental results of discriminations using the proposed abnormalities between normal and abnormal cases for 204 images including 108 abnormal cases have shown that the abnormalities are well effective to diagnose the congestion in internal hemorrhoids.

Keywords: medical image processing, computer-aided diagnosis, internal hemorrhoids, congestion, endoscopic image.

1 Introduction

Hemorrhoids come from the congestion of the blood caused by standing or sitting for long time at work, baby care, etc. With labor circumstances in the times, at least 70–80 % of population in developed countries could potentially have symptoms of hemorrhoids. However, since there is no nerve in the rectum, most of people do not notice the symptoms, and the symptoms have already become much worse in most cases when they notice the symptoms and go to a clinic. The congestion of the blood gradually changes into other worse symptoms of hemorrhoids. Therefore, the congestion is the most fundamental symptom in hemorrhoids.

When diagnosticians check internal hemorrhoids, they insert the endoscope from the anus into the rectum. Scenes of the rectum taken by the endoscope are shown to them via a monitor. At that time, if diagnosticians confirm a lesion of internal hemorrhoids, they save several endoscopic images of the lesion in order to compare with its future condition in addition to the record for the lesion. However, since it is difficult for even experts on hemorrhoids to diagnose internal hemorrhoids, disagreement between diagnosticians is often happened. For the reasons, a computer-aided diagnosis (CAD) system for evaluating internal hemorrhoids is required as a second opinion for diagnosticians. Measuring abnormality of the congestion in internal hemorrhoids could be useful for an assisted system for new doctors and non-experts.

Diagnosticians judge the congestive extent of the blood in internal hemorrhoids based on the color and the size of the congestive part. Although there is no report on extracting abnormalities for the congestive region in medical images, as similar trials in the field of medical image processing, recognition of lesions(Li et al. 2009, Kim et al. 2006), segmentations of organs or objects such as bleeding regions(Tjoa et al. 2001, Xiao et al. 2005, Tian et al. 2001, Vilarino et al. 2010), etc. have been reported. However, all of them are effective to only cases when contours of recognition parts are clear in images. Since the contour of the congestion is not clear in many cases due to shade around the contour, it would be difficult to apply them to the extraction of the congestion area. And, even if another technique to recognize an object in a gradation area like landscape images were existed, it would be hard to recognize the congestive region in endoscopic images because boundary between the region and its background is very dark in most cases due to shade, i.e., characteristics of color are hardly appeared around the boundary.

To design a CAD for internal hemorrhoids, this pa-

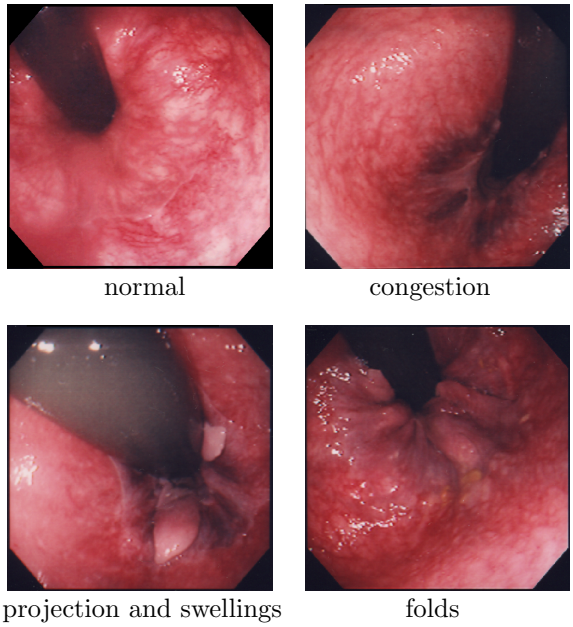
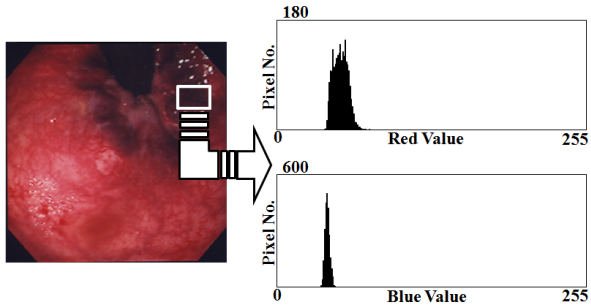
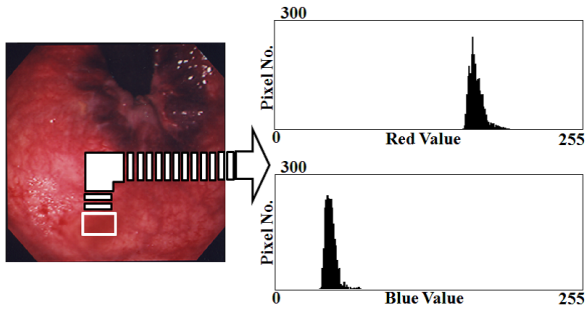


Figure 1: Typical cases of internal hemorrhoids.



(a) congestion area



(b) normal area

Figure 2: Difference of color between the congestive region anf normal area.

per proposes a method for extracting the congestive extent of internal hemorrhoids as abnormalities from endoscopic images. In addition, using the abnormalities, the proposed method diagnoses the images by discriminating between normal and abnormal cases as a prototype of the CAD. The abnormalities are measured by extracting characteristics of density distribution in the images. And then, this paper examines performance of the abnormalities from experimental results obtained by the discriminations.

2 Endoscopic Images of Internal Hemorrhoids and the Congestion of the Blood

Symptoms of internal hemorrhoids are broadly divided into three of 1)the congestive region appears around the tube or its tip, 2)projections or swellings appear around the tube with the anal canal opened, and 3)the folds appear around the tube due to hard projections and swellings; and when one of them appears, the object is diagnosed as internal hemorrhoids. Figure 1 shows an example of each case, where the upper left is a normal case and the other three are abnormal cases. And, the black cylindrical object in every image shown in Figure 1 is the tube of the endoscope. In the normal case shown in Figure 1, the whole color of internal rectum looks pink and there is no significant difference of color between the whole color and the area around the tube. On the other hand, as shown in the image at the upper right in Figure 1, we can see the color is changed into blackish purple around the tube in the congestion case. This part is the congestive region of the blood and a main factor to diagnose internal hemorrhoids caused by the congestion.

To endoscopic images of the rectum, this paper proposes features for measuring the congestion extent such cases shown at the upper right in Figure 1. Figure 2 shows difference of the histogram on each of red and blue values in RGB color system between a normal part and a congestive part, where the size of both of the two rectangles in the image is 40×60 and the histograms have been measured in each of the areas. In addition, Table 1 shows the mean values and the standard deviations for the histograms in Figure 2. From Figure 2 and Table 1, we can confirm that difference between red and blue values in the congestive part is much smaller than normal part, and red values in the congestive part are lower than normal part.

Table 1: The mean value and the standard deviation (s.d.) of the histograms shown in Figure 2.

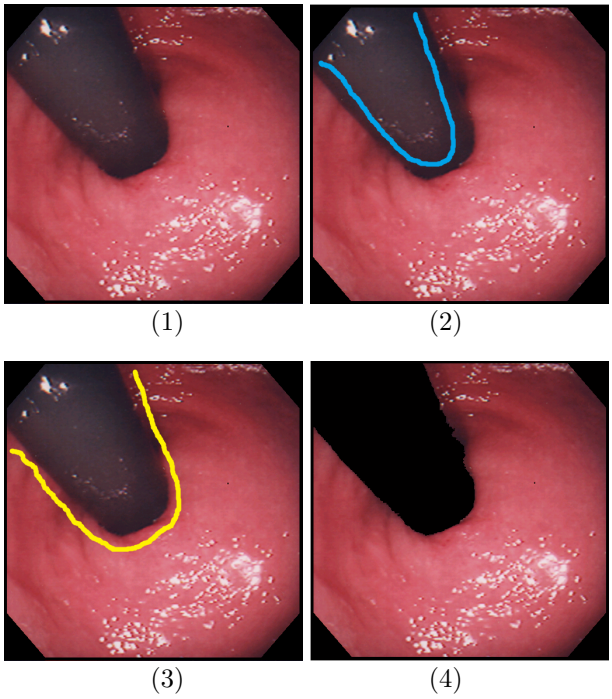
	mean value	s.d.
red value in (a)	55.4	32.7
blue value in (a)	44.0	57.3
red value in (b)	174.6	36.4
blue value in (b)	48.8	41.2

Throughout this paper, the size of all the endoscopic images is 512×512 pixels with 24-bit full color.

3 Proposed Method

3.1 Pre-processing

First, since the color of the tube region in the image is nonrelative in recognition of the congestive region and feature extractions of the congestive extent besides there is possibility the region could be a noise due to much halation, the tube region is extracted from the original image by the Lazy Snapping (LS)(Li et al. 2004) and all the pixel values in the region are converted into black color, i.e., the tube region is removed from the original image. LS is a method for separating an image into the target area and its back-



(1) (2)

(3) (4)

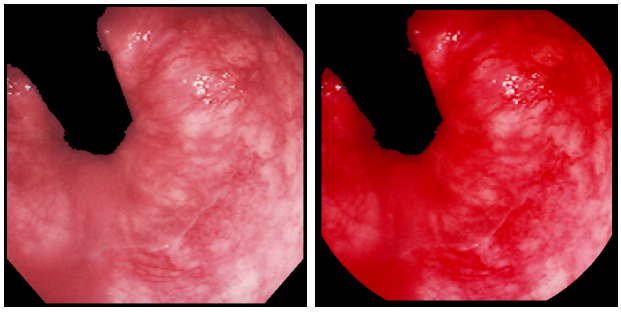
Figure 3: Extraction of the tube region by the Lazy Snapping.

ground in the image by roughly drawing the boundary between them with an interface, hence the user manually cuts the target drawing a boundary between the two areas with a mouse or a tablet-pan on the monitor. Figure 3 shows a case when LS is applied to one of the endoscopic images, where (1) is the original image, (2) is a curve drawn in the inside of the tube region along the tube contour, (3) is a curve drawn in the outside of the tube region along the contour, and (4) is the final output. As shown in Figure 3(4), only the color of tube region is converted into black color by using LS.

Next, to the image conducted the tube removal, pixel values in RGB space are converted into HSV space and the linear stretch with saturation is conducted. In the stretch, the histogram for saturation value S is stretched at the range [0, 1]. After the stretch, pixel values are reconverted into values in RGB space. Figure 4 shows a case when the stretch is conducted, where the left is an image before the stretch and the right is the converted image. By this conversion, the congestion area and blood vessels could be brighter. RGB values are converted into HSV values H (hue), S (saturation), and V (value) as follows:

$$H = 60 \times \begin{cases} \frac{G-B}{max-min} & if\ max\ is\ R \\ \frac{B-R}{max-min} + 120 & if\ max\ is\ G \\ \frac{R-G}{max-min} + 240 & if\ max\ is\ B \end{cases} \quad (1)$$

$$S = \frac{max - min}{max} \quad (2)$$



before after

Figure 4: The linear stretch with saturation.

$$V = max \quad (3)$$

where R, G, and B are color values of red, green, and blue in the pixel, respectively; and max and min are the highest and the lowest values among R, G, and B.

3.2 Exraction of an Abnormality on Congestion: f_1

As shown in Figure 2, red values of the blackish purple color appeared in the congestive region are basically higher than blue values. However, supposing features of the congestive extent were extracted considering only this characteristic, there could be possibility that halation and light-colored area become strong noises in the extraction. Besides, the red values are generally lower than red values in normal area. Considering them, an abnormality on congestion f_1 is given by

$$f_1 = \sum_{x=0}^{512} \sum_{y=0}^{512} P(x, y) \quad (4)$$

$$P(x, y) = \frac{B(x, y)}{R(x, y)} \times w_R(x, y) \quad (5)$$

where $w_R (0 \leq w_R \leq 1)$ is the weight determined by red value R and blue value B in the pixel (x, y).

Thus, f_1 would be an abnormality considering the the size of the congestive region and its color because f_1 measures the shade of the blackish purple with the ratio of B for R to all the pixels in the image.

3.2.1 Derivation of the Weight $w_R(x, y)$

The weight $w_R(x, y)$ in the pixel (x, y) is determined by employing the following sigmoid function:

$$w_R(x, y) = \frac{1}{1 + e^{-a(-127 + R(x,y) - axis)}} \quad (6)$$

where a is the gain value of the function, $R(x, y)$ is red value in the pixel (x, y), and $axis$ is defined as x-coordinate at the axis of symmetry in the sigmoid curve. The reason why the sigmoid function is employed is because various weights could be brought by changing the gain and x-coordinate at the axis of

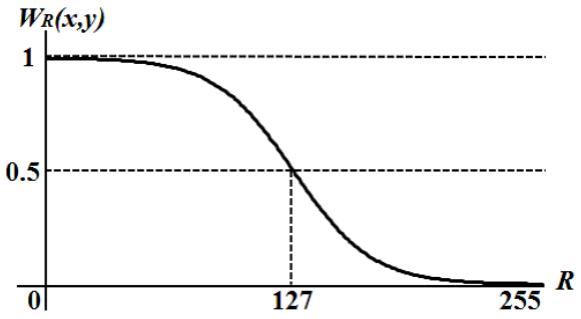


Figure 5: Sigmoid curve for Eq.(6) (a: 0.5, $axis$: 0).

symmetry. Therefore, the more $R(x, y)$ is small, the more w_R is large (i.e., w_R approaches 1), hence f_1 is going to the ratio of B for R shown in Eq.(5). On the other hand, the more $R(x, y)$ is large, the more w_R is small (i.e., w_R approaches 0), hence the ratio is revised low and f_1 is held down. As an example of the sigmoid function given by Eq.(6), Figure 5 shows a sigmoid curve for Eq.(6) where a and $axis$ are 0.5 and 0, respectively.

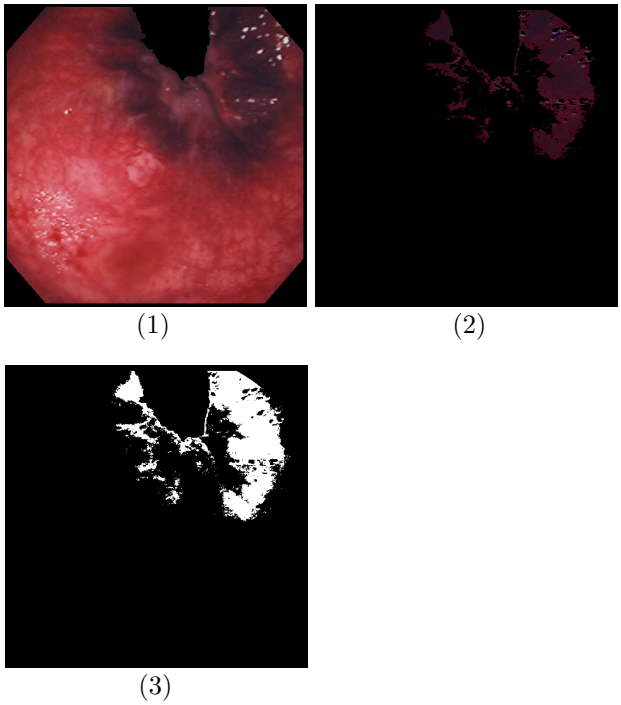
3.2.2 Determination of the Parameters a and $axis$ in the Weight $w_R(x, y)$

a and $axis$ are determined by using samples selected by principal component analysis. The weight $w_R(x, y)$ should be introduced in Eq.(5) to enhance difference of red value between the normal and abnormal cases on the congestion. Considering this, first, in order to obtain the standard samples in each dataset of normal and abnormal cases, principal component analysis is applied to all the images in each case separately, where the variables for the analysis are regarded as the three color values in RGB. By the analysis, the images are represented as vectors which have three coordinates of the first, second, and third principal components in a 3D space composed of the components. Putting all the images in each class in the 3D space, the image which is the nearest vector from the mean vector in each of the 8 quadrants in the 3D space is selected, i.e., 8 images are selected as the standard samples from each dataset of normal and abnormal cases. Second, using f_1 extracted from the standard samples, degree of separation between the couple of datasets composed of the standard samples in each case is calculated. The degree of separation (Sep) is given by

$$Sep = \frac{\sigma_b^2}{\sigma_w^2} \tag{7}$$

$$\sigma_b^2 = \frac{N_n N_a (m_n - m_a)^2}{(N_n + N_a)^2} \tag{8}$$

$$\sigma_w^2 = \frac{N_n \sigma_n^2 + N_a \sigma_a^2}{N_n + N_a} \tag{9}$$



(1) (2) (3)

Figure 6: Extraction of the congestive region for a case.

where N_n and N_a are the number of the standard samples in each of normal and abnormal cases ($N_n = N_a = 8$), m_n and m_a are the mean value of f_1 extracted from the standard samples in each, σ_n^2 and σ_a^2 are variance of f_1 in each, respectively. Third, changing both values of $axis$ in the range [-127, 127] (pitch: 1) and a in the range [0.1, 10.0] (pitch: 0.1), Sep is calculated each time. And then, the couple of a and $axis$ used in the case when Sep obtains the largest value is determined as their optimum values at last. $axis$ is the x-coordinate at the point of inflection in a sigmoid function, $w_R(x, y)$ has a value for $R(x, y)$ in Eq.(6), and the range of $R(x, y)$ is [0, 255]. Hence, the size of the domain for $axis$ should be the same as the domain for $R(x, y)$. Besides, considering $axis = 0$ in the case when $w_R(x, y)$=0.5, the domain for $axis$ becomes [-127, 127].

3.3 Extraction of the Congestive Region

The congestive region is extracted by the linear discriminant analysis with $P(x, y)$ shown in Eq.(5). The discriminant analysis is applied to every pixel in the image and their discriminant score decides whether each pixel joins in the congestive region or its background. If the score for a pixel is more than 0, the pixel joins in the congestive region. Figure 6 shows the extracted congestive region for a case, where (1) is the original image, (2) is the extracted congestive region, and (3) is the black and white image of (2).

3.4 Extraction of abnormalities f_2, f_3, and f_4

In addition to f_1 proposed in **3.2**, the other abnormalities on the congestive extent f_2, f_3, and f_4 are extracted as features from the extracted congestive region obtained in **3.3**. f_2 represents the size of the

congestive region. f_3 represents the mean value of red values in the congestive region. And, f_4 represents the mean value of the ratio of the blue values for the red values in the congestive region. Now, when the number of pixels in the congestive region is $size$ and one of them is represented by c_i $(1 \leq i \leq size)$, the abnormalities are given by

$$f_2 = size \tag{10}$$

$$f_3 = \frac{1}{size}\sum_{i=1}^{size} r_i \tag{11}$$

$$f_4 = \frac{1}{size}\sum_{i=1}^{size} \frac{b_i}{r_i} \tag{12}$$

where r_i and b_i are the red and the blue values of the pixel c_i, respectively.

4 Experimental Results

To examine performances of the proposed abnormalities, linear discriminant analysis (LDA), discriminant analysis by Mahalanobis distance (MD), neural network (NN), linear support vector machine (L-SVM), and nonlinear support vector machine (N-SVM) were applied to the discrimination of the endoscopic images into normal or abnormal (i.e., internal hemorrhoids) cases by regarding the abnormalities as variables, respectively. Each of the abnormalities has been normalized to the dataset that the mean value is 0 and the variance is 1. NN was designed as a three-layered perceptron (input layer: 3, hidden layer: 5, output layer: 3), where sigmoid function and back propagation (learning rate: 0.3, weight decay rate: 0.1) were used. As a kernel function in N-SVM, the Gaussian kernel was used. Selecting the standard samples for each of normal and abnormal cases from this dataset, the parameters a and $axis$ in Eq.(6) have been obtained. Figure 7 shows Sep vs. the parameters a and $axis$. From the result shown in Figure 7, the parameters a and $axis$ were obtained as 4.1 and -2, respectively. The number of the images is 204 (normal case: 96, abnormal cases: 108), where the 16 standard samples selected to determine a and $axis$ are not included.

The training dataset and test dataset were chosen by the cross validation (Mosteller 1948) in all the discriminations. In fact, every discrimination was conducted as the following procedure:

(1) Choose 1 image from all the images as test data, and use the other images as training dataset.

(2) Discriminate the test data between the class of "normal" and "abnormal".

(3) Repeat the procedure from (1) to (2) to every image changing the test data.

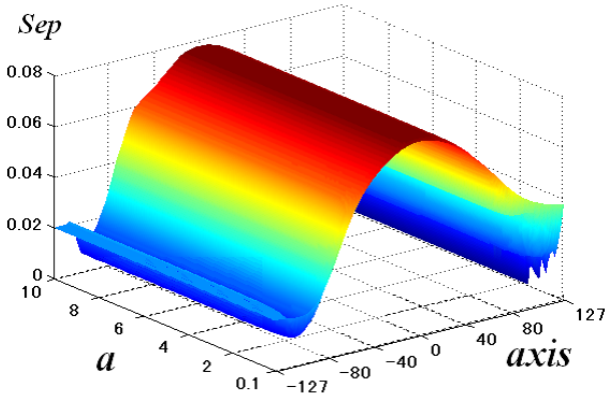


Figure 7: Sep vs. a and $axis$.

Table 2: Discrimination results of abnormal cases.

tool	*Pecision*	*Recall*
LDA	81.5% (88/108)	86.3% (88/102)
MD	89.8% (97/108)	81.5% (97/119)
NN	83.3% (90/108)	88.3% (90/102)
L-SVM	81.5% (88/108)	86.3% (88/102)

Table 3: Discrimination results of normal cases.

tool	*Pecision*	*Recall*
LDA	85.4% (82/96)	80.4% (82/102)
MD	77.1% (74/96)	87.1% (74/85)
NN	87.5% (84/96)	82.3% (84/102)
L-SVM	85.4% (82/96)	80.4% (82/102)

Table 2 and Table 3 show discrimination results for abnormal and normal cases respectively obtained by the four discrimination machines of LDA, MD, NN, and L-SVM, where *Precision* and *Recall* represent the discrimination ratio; and the numbers in parentheses are the number of images used for their calculations, which are defined as

$$Pecision = \frac{|X_h \cap X_c|}{|X_c|} \times 100 \tag{13}$$

$$Recall = \frac{|X_h \cap X_c|}{|X_h|} \times 100 \tag{14}$$

where X_h is the set of the correct answers, X_c is a set of images discriminated by the proposed method, and $|X|$ is the number of images of a set X. In addition, Figure 8 shows the discrimination ratios against

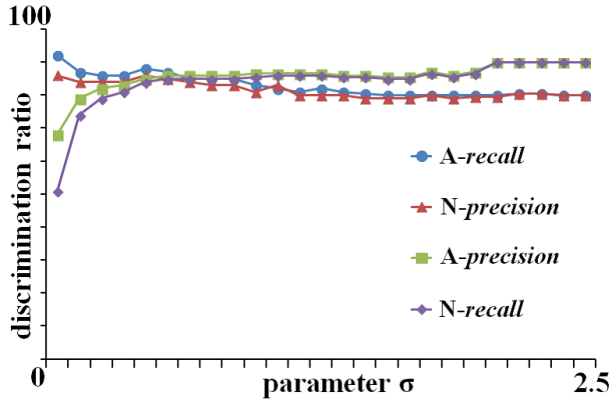


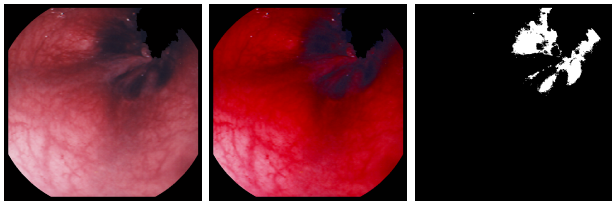
Figure 8: The discrimination ratios vs. σ of the Gaussian kernel in N-SVM.

Table 4: Mean values of the abnormalites in each of correct abnormal and normal datasets (The number in the parenthesis is the standard deviation.).

abnormality	abnormal	normal
f_1	0.55 (1.10)	-0.61 (0.17)
f_2	0.57 (1.03)	-0.64 (0.39)
f_3	-0.55 (0.80)	0.62 (0.81)
f_4	0.59 (0.99)	-0.66 (0.42)

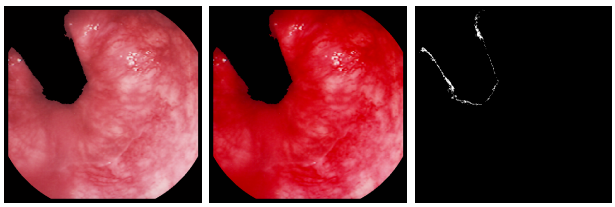
change of the parameter σ in the Gaussian kernel used in N-SVM, where A and N means abnormal and normal cases, respectively. From the fact that the discrimination ratios for each of normal and abnormal cases show more than 80% except the precision of normal case in MD, experimental results shown in Table 2, Table 3, and Figure 8 show the proposed method has high performance in the discrimination. Besides, to pass the film reading test of mammography in Japan, medical doctors have to correctly read at least 80% in each of cancer cases and normal cases. It means the discrimination ratio in the proposed method is enough high from clinical standpoint. Thus, the experimental results have shown that the proposed abnormalities are appropriately extracted for the discrimination of internal hemorrhoids caused by the congestion. Table 4 shows the mean value of each abnormality extracted from correct dataset in each case, where the number in the parenthesis is the standard deviation.

Figure 9 shows an example of discrimination success for the abnormal case in all the discrimination ways. In Figure 9, (1) is the image where the tube has been removed from the original image, (2) is the image converted by the stretch with saturation for (1), and (3) is the black and white image of congestive region extracted from (2) and the region is shown as the white color. In addition, Figure 10 shows an example of discrimination success for the normal case in all the discrimination ways, where (1), (2), and (3) are the same as Figure 9, respectively.



(1) (2) (3)

Figure 9: An abnormal case of discrimination success ($f_1 = 0.100$, $f_2 = 0.199$, $f_3 = -0.741$, $f_4 = 0.174$).



(1) (2) (3)

Figure 10: A normal case of discrimination success ($f_1 = -0.766$, $f_2 = -1.003$, $f_3 = 1.574$, $f_4 = -1.273$).

5 Discussions

In the experimental results, the number of images which were discrimination failure in all the discrimination machines is 15, where 7 normal cases were diagnosed as abnormal cases and 8 abnormal cases were done as normal cases. Figure 11 – 14 show examples of the discrimination failures, where (1), (2), and (3) are the same as Figure 9, respectively.

First, normal cases which were diagnosed as abnormal cases can be divided broadly into two kinds. Figure 11 and Figure 12 show a result of each kind. In the case of Figure 11, we could confirm there is the congestive region though this case is a normal case, besides $f_1 = -0.226$ and $f_2 = 0.446$, i.e., the features are closer to the mean value of them in abnormal case shown in Table 4. Actually, this case was internal hemorrhoids before and it is going to the recovery in progress; and a diagnostician concluded this case has already gone out of abnormal cases. The other 3 normal cases were the same as this. In the case of Figure 12, we can confirm that there is shade such as shadow appeared in a tunnel. Since the proposed method cannot distinguish between the congestive region and the shade, the proposed method cannot extract the features precisely in this kind of cases when deep shade is appeared around the tube. In fact, f_3 and f_4 are -0.301 and 0.067, respectively, and they are closer to the mean values of them in abnormal case than normal case shown in Table 4. The other 2 normal cases were the same as this. In the future, it would be necessary to improve the features considering characteristics of distributions of the congestive region and the shade in the 2D image.

Second, abnormal cases which were diagnosed as normal cases can be also divided broadly into two kinds. Figure 13 and Figure 14 show a result of each kind. In the case of Figure 13, we can con-

(1) (2) (3)

Figure 11: Failure case 1 (by the proposal: abnormal, correct answer: normal)

(1) (2) (3)

Figure 12: Failure case 2 (by the proposal: abnormal, correct answer: normal)

(1) (2) (3)

Figure 13: Failure case 3 (by the proposal: normal, correct answer: abnormal)

(1) (2) (3)

Figure 14: Failure case 4 (by the proposal: normal, correct answer: abnormal)

firm there are some projections and swellings. Since the proposed abnormalities are designed by considering only the congestion, the abnormalities cannot apply to these symptoms. Regarding the image shown in Figure 13, the abnormalities are $f_1 = -0.633$, $f_2 = -0.669$, $f_3 = 0.412$, and $f_4 = -0.486$, respectively, and all of them are closer to the mean values of normal cases shown in Table 4. The other 3 abnormal cases were the same as this. In the future, it is necessary to propose a method for recognize the projection and swelling parts in order to diagnose these symptoms separately. In the case of Figure 14, due to the lightning of the endoscope, a change of hue has been occurred hence the abnormal case was discriminated as a normal case. The other 3 abnormal cases were the same as this. Regarding the image shown in Figure 14, the abnormalities are $f_1 = -0.755$, $f_2 = -1.018$, $f_3 = 0.885$, and $f_4 = -0.208$, respectively, and all of them are closer to the mean values of normal cases shown in Table 4. The lightning sometimes brings halation or a change of hue into endoscopic images. If the location of the light source could be obtained, there could be possibility that the color of the image is reproducted into the original color. Since this problem can be also occurred in other general images and it is still not solved, in order to cope with this case, it might be necessary to consider preparing a manual on taking the desirable pictures for diagnosticians.

6 Conclusions

Aiming at showing a second opinion to medical doctors and supporting non-experts in internal hemorrhoids, this paper has presented abnormalities for measuring the congestive extent in endoscopic images of internal hemorrhoids. The proposed abnormalities provide degree of congestive color density and size of the congestive region as features besides they could be used for a computer-aided diagnosis of internal hemorrhoids. Regarding the abnormalities as variables for discrimination of internal hemorrhoids, this paper has examined performance of the abnormalities by discriminating between normal and abnormal cases with discrimination machines. Experimental results of the discriminant trials have shown that the discrimination ratios for the proposed method have become more than 80% in almost of the trials. In addition, we have confirmed the proposed method could extract the congestive region well. However, the proposed method cannot cope with abnormal cases caused by the other symptoms, yet. And also, it is difficult to diagnose internal hemorrhoids to the images where hue have been changed due to the light of the endoscope.

Therefore, as future works, it is necessary to 1)design the features considering the distribution of the congestion region and shades in the image, 2)propose a method for recognize the projection and swelling parts, and 3)design a manual on taking the desirable pictures for medical doctors.

References

B. Li and M.Q. Meng (2009), 'Computer-Aided Detection of Bleeding Regions for Capsule Endoscopy Images', *IEEE Trans. Biomed. Eng.*, **56**(4), 1032–1039.

K-B Kim, S. Kim, and G-H Kim (2006), 'Analysis System of Endoscopic Image of Early Gastric Cancer', *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, **E89-A**(10), 2662–2669.

M.P. Tjoa, S.M. Krishinan, C. Kugean, P. Wang, and R. Doraiswami (2001), 'Segmentation of Clinical Endoscopic Image Based on Homogeneity and Hue', *Proc. Annual Int. Conf. IEEE Eng. Med. Biol. Soc.*, **3**, 2665–2668.

M. Xiao, S. Xia, and S. Wang (2005), 'Geometric Active Contour Model with Color and Intensity Priors

for Medical Image Segmentation', *Proc. 27th Annual Int. Conf. IEEE Eng. Meg. Biol. Soc.*, 6496–6499.

H. Tian, T. Srikanthan, and K.V. Asari (2001), 'Automatic segmentation algorithm for the lumen region and boundary from endoscopic images', *Med. Biol. Eng. Comput.*, **39**(1), 8–14.

F. Vilarino, P. Spyridonos, F. Deiorio, J. Vitria, F. Azpiroz, and P. Radeva (2010), 'Intestinal Motility Assessment With Video Capsule Endoscopy : Automatic of Phasic Intestinal Contractions', *IEEE Trans Med Imaging*, **29**(2), 246–259.

Y. Li, J. Sun, C-K Tang, and H-Y Shum (2004), 'Lazy Snapping', *ACM Trans. on Graphics*, **23**(3), 303–308.

F. Mosteller (1948), 'A k-sample slippage test for an extreme population', *The Annals of Mathematical Statistics*, **19**(1), 58–65.

Evaluating Surgical Performance in Real Time Using Data Mining

Yun Zhou[1] **Ioanna Ioannou** [2] **James Bailey** [1]
Gregor Kennedy [3] **Stephen O'Leary** [2]

[1] Department of Computing and Information Systems
University of Melbourne,
Email: yuzhou@student.unimelb.edu.au, baileyj@unimelb.edu.au
[2] Department of Otolaryngology
University of Melbourne,
Email: ioannoui@unimelb.edu.au, sjoleary@unimelb.edu.au
[3] Centre for the Study of Higher Education
University of Melbourne
Email: gek@unimelb.edu.au

Abstract

Virtual reality simulators are becoming increasingly popular as adjuncts to traditional surgical training methods, but most simulators do not have the ability to evaluate performance on-the-fly and provide advice to trainees as they practice. Timely feedback on performance is a critical component of surgical training, therefore the ability to provide such evaluation is necessary if simulation is to be effective as a platform for self-guided learning. We propose an evaluation framework to automatically assess performance within a temporal bone simulator in real time. This evaluation framework uses data mining techniques to assess performance at different granularities. Drilling technique is analysed to deliver detailed short-term evaluation, while hidden markov models are used to evaluate the completion of small surgical subtasks and provide medium-term assessment. Finally, an analysis of drilled bone shape is used to evaluate performance at the completion of each stage of a surgical procedure. We demonstrate the effectiveness of the proposed methods by validating them on an existing simulation dataset.

Keywords: surgical simulation, online evaluation

1 Introduction

Immersive virtual reality (VR) simulators with haptic capabilities are increasingly seen as convenient, cost-effective and valid training tools in surgical training programs (Agus et al. 2003, Bryan et al. 2001, Kerwin et al. 2009). These VR simulators use techniques such as 3D illusion and haptic feedback to simulate surgical interactions on virtual anatomical models. These virtual platforms have attracted much attention in the field of surgical education, since they have the potential to provide repeatable practice on a variety of surgical cases with varying difficulty, at the trainee's convenience.

Timely performance evaluation plays a critical and essential role in the development of surgical expertise through deliberate practice (Ericsson 2004). Past work has demonstrated that trainees are not able to improve their skills without feedback and suggestions (Darzi & Mackay 2002). Timely performance evaluation - provided during the execution of a task - is a critical component that is currently lacking in existing VR surgical training environments. In most cases, human expert instructors are still required to provide appropriate feedback as tasks are undertaken by trainees. The limited availability of expert supervision and the subjective nature of surgical skill assessment is an impediment in the wide adoption of VR simulators for surgical training. There is great potential to improve surgical training if VR simulators can be used as unsupervised self-guided learning tools by providing trainees with valid and reliable evaluation of their performance.

Previous work on automated performance evaluation within surgical simulators has focussed on offline assessment (Murphy et al. 2003, Rosen et al. 2001, Sewell et al. 2007, 2008). There are several disadvantages to this approach. First, there is a large delay between actions and the provision of feedback on those actions; trainees may not be able to recall the mistake which prompted the feedback. Second, score values may be difficult to interpret and trainees may not understand what they need to change in order to improve. Third, the assessment provided is summative in nature and pertains to the surgical task as a whole, rather than providing detailed evaluation of specific sub-tasks. A trainee may perform well on some sub-tasks and badly on some others, so it is highly beneficial to provide evaluation of appropriate granularity.

It is intrinsically more difficult to evaluate surgical performance online than offline. One difficulty lays in determining how much information should be collected during a surgical procedure in order to provide an accurate performance evaluation. As a trainee progresses through the procedure, the information available to expertise prediction models increases, as does the accuracy of assessment. However, accuracy comes at the expense of timeliness. To provide real time evaluation, it is necessary to develop expertise models which are capable of providing reasonably accurate predictions using short segments of surgical procedure.

Another challenge in providing online evaluation is that different stages or sub-tasks within a surgical procedure may require different techniques. Therefore, feedback should be context-aware, and the provision of such evaluation is predicated on the ability to determine the current stage within a surgical

procedure. Automatic online stage prediction is a non-trivial task, even for a procedure such as cortical mastoidectomy, with has pre-defined steps to expose a series of anatomical structures. Each surgeon may perform the steps with slight variations, such as interleaving them or returning to a previous step at a later stage.

In this paper, we study the important problem of delivering online performance evaluation within a VR surgery simulator. The procedure chosen as the basis for this investigation was cortical mastoidectomy, a temporal bone surgery procedure. Designing an evaluation system for this type of surgery is a complex undertaking. To provide automated evaluation that mimics the advise of a human expert, the system must be able to assess various aspects of performance - such as drilling technique and outcome - in real time and provide timely and accurate evaluation. Thus, our main aims in developing the evaluation system was to provide:

- Online surgical performance evaluation without sacrificing prediction accuracy;
- Multi-level evaluation of surgical technique and end-product.

We propose an event-based framework to deliver different levels of performance evaluation during ongoing simulator training. The rest of the paper is organized as follows. We first review recent work on automated performance evaluation within surgical simulators in section 2. We briefly introduce our temporal bone simulator in section 3, and propose a general framework to deliver different levels of online evaluation in section 4. We then explain each component of this framework and report the corresponding experimental results in sections 5, 6 and 7. In section 8, we summarize our contributions and discuss future work.

2 Related Work

In the field of temporal bone surgery simulation, the majority of existing work on automated performance evaluation has focussed on offline assessment of surgical outcomes, usually by means of a statistical analysis of measures recorded during simulator training.

Sewell et al. (2007, 2008) evaluated performance within a temporal bone simulator by determining whether a specified objective was met while avoiding damage to sensitive anatomical structures. They recognised that expert and novice surgeons drill different regions of the bone, and selected the top 1000 informative voxels as features to train a Naive Bayes model to evaluate expertise. To provide offline performance evaluation, they used green dots to mark bone voxels that had been removed using good surgical technique while red dots represented improperly removed voxels (Sewell et al. 2008). They showed that when this feedback was provided, trainees removed more voxels correctly compared to a control group. Kerwin et al. (2012) constructed a decision tree to automatically evaluate surgical performance from voxel-related features. While these approaches were shown to be effective, none of them provide online performance evaluation.

Zhou et al. (2013) used random forest models to predict surgical expertise and generate meaningful automated real time feedback on surgical technique within a VR temporal bone simulator. One limitation of this approach is that the training of random forest models assumed that experts use only good technique and that trainees use only bad technique. This assumption is useful in labelling the training data, but is not generally true in practice, and it limits the accuracy of the models. Furthermore, feedback was limited to advice on surgical technique, while no attempt was made to advise trainees as to where they should drill.

In the field of minimally invasive surgery, Hidden Markov Models (HMMs) have been applied in various ways to evaluate surgical skill. Rosen et al. (2001) built expert and novice HMM models with hidden states representing surgical gestures and force/torque measurements as observations. Their approach achieved 87.5% accuracy in the classification of expert and novice performance. Instead of using HMM to represent different skill levels, Murphy et al. (2003) built HMMs for each type of basic gesture. Their results showed that expert surgeons use fewer motions overall as well as less wasted motions compared to novices. Lin et al. (2006) used a combination of linear discriminant analysis and Baysian classification to segment and label sequences of laparoscopic surgical suturing gestures. They achieved 90% prediction accuracy in labelling the preprocessed suturing gestures, which suggests that surgical gestures can be extracted to produce highly accurate skill evaluation. These approaches achieved good results, but they still provided only offline evaluation. Furthermore, they may not be applicable to surgical procedures that are not composed of a distinct set of surgical gestures, such as temporal bone surgery.

In view of the limitations of the methods discussed above, we propose a new multi-level evaluation framework to deliver short-term, medium-term and long-term assessment of drilling technique and surgical outcomes.

3 Temporal Bone Simulator

The temporal bone simulator referenced in this study is the University of Melbourne/CSIRO simulator (Hutchins et al. 2005, O'Leary et al. 2008). In this section, we briefly describe the simulator and the collection of the training data used in the development of evaluation models.

3.1 Simulator Environment

The simulator presents users with a 3D temporal bone model constructed from CT data. Figure 1 shows a semi-transparent view of the temporal bone with underlying anatomical structures, as viewed within the simulator. Users interact with the virtual temporal bone using two pen-like haptic devices that provide force feedback. The first haptic device represents a surgical drill while the second haptic device represents the irrigator, which lubricates the drill and prevents over-heating of the bone.

3.2 Data Collection

Cortical mastoidectomy was chosen as the surgical procedure to be investigated in this work. Cortical mastoidectomy is the preparatory step of many otological operations and is the most basic of all mastoid procedures. The aim of mastoidectomy is to locate a series of anatomical structures by removing the overlaying bone, while avoiding damage to those structures. A mastoidectomy is performed in a series of steps, during which the surgeon identifies a series of anatomical landmarks in a specific order, so that these structures can be protected, and the surrounding bone removed in a safe manner.

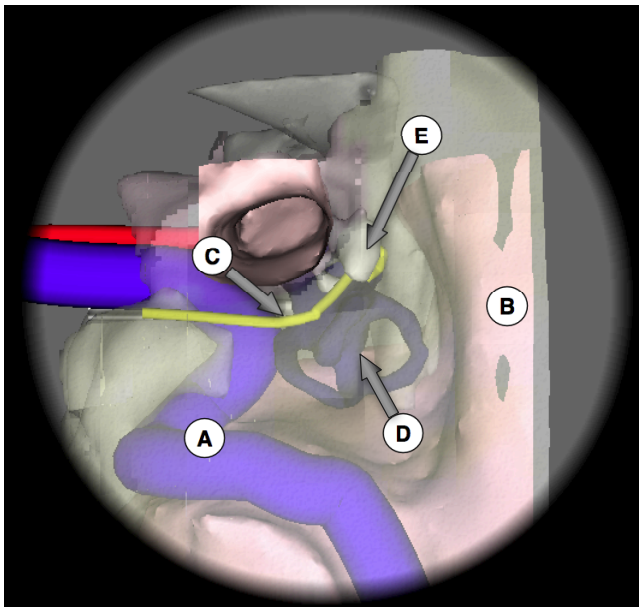


Figure 1: Semi-transparent view of the simulated temporal bone, showing important anatomical structures.
A: sigmoid sinus, B: dura, C: facial nerve, D: lateral semi-circular, E: incus

Table 1: Low-level metrics recorded by the temporal bone simulator

Time stamp
Tool position, orientation and force metrics
Current force applied by drill tool (X,Y,Z)
Current position of drill tool (X,Y,Z)
Current orientation of drill tool (X,Y,Z,Angle)
Current position of suction tool (X,Y,Z)
Current orientation of suction tool (X,Y,Z,Angle)
Simulator settings
Current burr spinning speed
Current burr radius
Anatomical structure metrics
Number of drilled voxels
Distance from the drill tip to the closest point of the structure surface
Distance from the suction tip to the closest point of the structure surface

We recruited six participants to perform 62 trials on the simulator. Participants were divided into an expert group and a novice group. The expert group consisted of three qualified Ear, Nose and Throat (ENT) surgeons, while the novice group consisted of three university students with no medical or surgical training. The functions and operation of the simulator were explained to all participants and they were given time to familiarise themselves with it. Participants in the novice group were provided with a tutorial video showing the full procedure performed on the simulator by an expert surgeon. Participants were asked to carry out the each mastoidectomy stage sequentially and indicate to the researchers whenever they completed the stage.

The majority of novices in our dataset encountered difficulties in completing the latter stages of the procedure, so this study focuses on the first three stages of mastoidectomy, as they were completed by all participants. The stages were: 1) removing the outer layer of thick (cortical) bone, 2) finding and exposing the sigmoid sinus, 3) finding and exposing the dura.

During each trial, the simulator recorded a variety of measures at a rate of approximately 15 Hz. Table 1 lists the low-level metrics recorded by the simulator. Anatomical structure metrics were collected for the dura, sigmoid sinus and facial nerve.

4 Evaluation Framework

We approach the problem of performance evaluation by proposing an event-based framework that provides online assessment with different temporal granularities: short-term, medium-term, and long-term. Short-term and medium-term evaluation focus on drilling technique. Short-term evaluation provides detailed assessment of immediate stroke technique, while medium-term evaluation assesses technique based on a longer time frame. Finally, long-term evaluation provides an assessment of end product quality upon completion of each stage, using bone shape analysis. The rationale in using three different levels of evaluation is to provide timely, appropriate and relevant feedback to the trainee without overwhelming them with information.

Evaluation is triggered by three event types in the simulator: the end of each stroke, a motion pause, or the completion of a stage. The end of a stroke triggers short-term evaluation, a motion pause generates medium-term evaluation, and the completion of a stage cues long-term evaluation. The raw metrics generated by the simulator (Table 1) are analysed at each time point to determine whether any of the three event types has occurred.

Short-term evaluation is provided by a mathematical evaluation of immediate drilling technique. Medium-term evaluation is generated once sufficient data has been collected, by means of a HMM model representing different types of drilling. Long-term evaluation is produced by an analysis of drilled voxels. Figure 2 demonstrates the work-flow of the proposed event-based evaluation system.

5 Short-Term Evaluation

The aim of short-term evaluation is to assess the surgical technique of trainees as they practice on the simulator. Instead of assessing the entire surgical procedure, this type of evaluation focusses on individual units of motion, which we refer to as "strokes". We define a stroke as a continuous drilling motion without significant change in direction.

Strokes are identified using an online segmentation of the trajectory formed by the position coordinates of the virtual drill. A pause in "bone" removal or an abrupt change in direction signifies the end of a stroke. In the case of a change in direction, the end point of the stroke is identified using a k-cos algorithm adapted from Hall et al. (2008). The end of a stroke is considered to be a good time to evaluate the quality of the preceding stroke and provide short-term feedback if necessary.

Sections 5.1 and 5.2 describe the metrics used to assess the quality of drilling technique and report the experimental results of the validation performed on these metrics.

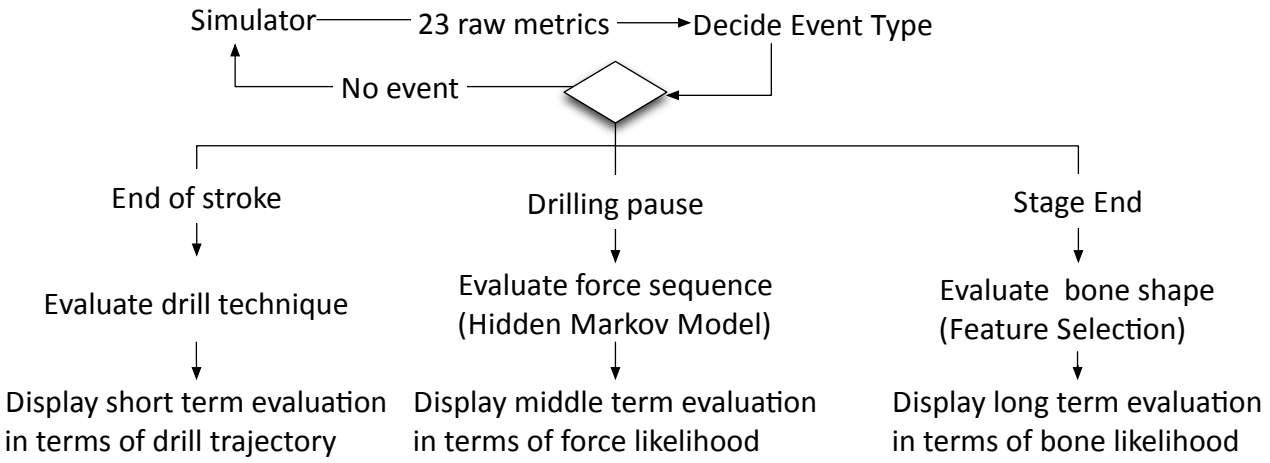


Figure 2: Event driven evaluation system work flow

5.1 Drilling Technique Analysis

The metrics used to assess the quality of drilling technique were derived with the guidance of expert surgeons, who provided a set of principles which characterise good technique in the context of temporal bone surgery.

The first principle is that drilling should be parallel to the surface of underlying anatomical structures. This technique minimises the risk of damaging a structure before it is recognised by the surgeon. Figure 3 demonstrates the associated technique. To quantify and measure adherence to this principle using simulator metrics, we define a parallel score (PS) as the cosine value between a drilling stroke and a related anatomical structure. We use vectors to approximate the shape of the stroke and the surface of the anatomical structure. So PS is defined as $\frac{\vec{s}\cdot\vec{a}}{|\vec{s}|\cdot|\vec{a}|}$, where $\vec{s}$ is a vector representing the stroke and $\vec{a}$ is a simplified vector representing the anatomical structure of interest.

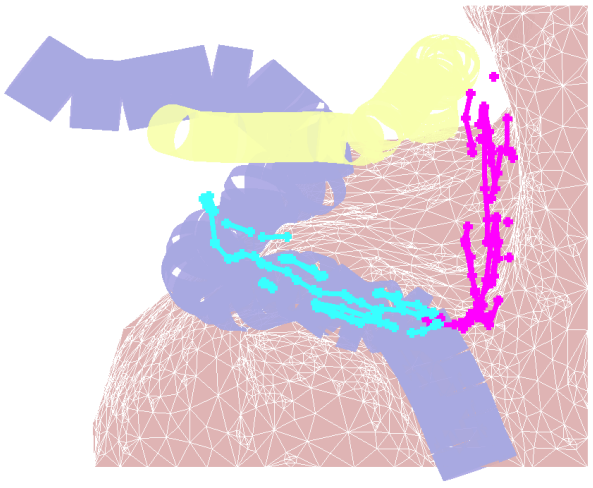


Figure 3: Example of parallel drilling technique. The blue structure represents the sigmoid sinus, the brown structure represents the dura and the yellow structure represents the facial nerve. Cyan trajectories demonstrate drilling parallel to the sigmoid sinus. Magenta trajectories demonstrate drilling parallel to the dura.

The second principle of good technique suggests that the drilling motion should be long and smooth, akin to a painter's brush stroke. This type of motion is considered to be safer than a 'poking' drilling style, as it affords better control and visibility. To capture this principle, we derived the measure of angular score, which we define as $AS = \frac{L^2}{d}$, where L is the width of a stroke and d is the depth of a stroke. Long, flat strokes will have higher angular score compared to shorter or deeper strokes, as shown in Figure 4.

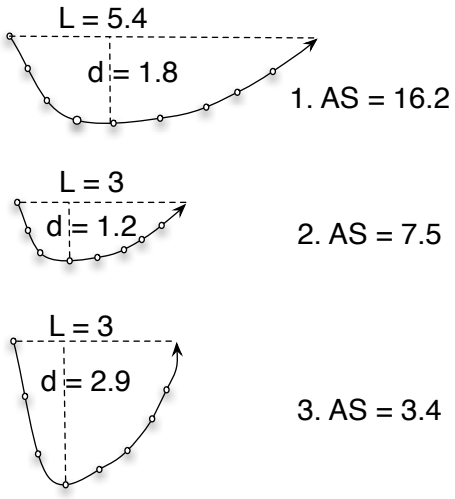


Figure 4: Example of strokes and angular scores. Stroke 1 is better than stroke 3 because it is flatter. Stroke 2 and stroke 1 have a similar smoothness, but stroke 2 is has a lower score due to shorter length.

The third metric we use to measure the smoothness of motion is jerk magnitude, which is defined as the derivative of acceleration. Jerk has been used in the past to measure the gracefulness of robotic movement (Hogan 1987). Since force is proportional to acceleration, we use the force measures provided by the simulator to derive the jerk magnitude (formally defined as yank). The jerk magnitude is defined as $j \propto \frac{\Delta F}{\Delta t}$ where ΔF is the change in force between measurement points and Δt is the elapsed time. The mean squared jerk magnitude for the stroke is computed as the sum of each point's jerk value and divided by stroke duration as $J = \frac{\sum j}{\sum \Delta t}$.

Table 2: AUC of short-term drilling technique metrics across three stages. Stroke length is the baseline metric

Stage	Stroke Length (SL)	Angluar (A)	Parallel (P)	Jerk (J)	SL+A +P+J
1	0.772	0.668	0.581	0.784	**0.801**
2	**0.722**	0.376	0.534	0.569	0.484
3	0.665	**0.795**	0.593	0.409	0.777

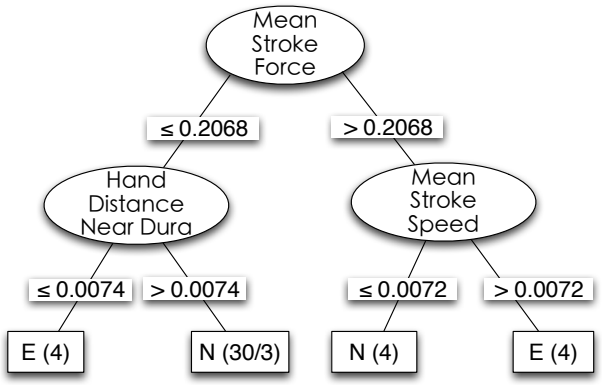


Figure 5: Decision tree for stage 1. E represents experts, N represents novices.

5.2 Drilling Technique Assessment Results

In order to validate the three metrics presented above, we evaluated the ability of each metric to distinguish between expert and novice strokes. To carry out this evaluation we calculated the value of each technique metric for all strokes in the training data. Each stroke was labelled as expert or novice according to the experience of the surgeon undertaking the simulator trial. Each stroke metric and the associated labels were used to build a logistic regression model. We obtained the 10 cross-validation Area-Under-Curve (AUC) value from the logistic regression model of each metric.

For comparison purposes, we chose the average stroke distance as a baseline metric, since it has been shown to be predictive of expertise (Grober et al. 2003). Table 2 shows the AUC values for the baseline and the three metrics proposed in section 5.1, as well as a combination of those three metrics. For stage 1, the combined metrics achieved the best result. However, in stage 2 stroke distance metric achieved a higher AUC than all our proposed metrics. This may be due to oversimplification of the anatomical structure shape and stroke motion. Another possible explanation is that stroke length is simply more important in stage 2. In stage 3, angular score was the most predictive metric, followed closely by the combined metrics. The results show that AUC values vary across the three stages, which suggests that the usefulness of these metrics is highly stage-dependent. There is no 'gold standard' measure we can choose to evaluate short-term drilling technique across all stages. The results suggest that predicting the current stage online is critical, as it determines which short-term metric will provide the most accurate evaluation. Stage prediction is explained later in this paper.

6 Medium-Term Evaluation

The aim of medium-term evaluation is to provide drilling technique assessment over a longer period than the immediate drill motions. This type of assessment may be delivered when there is a pause in drilling. A drilling pause is defined as a time when force magnitude is around zero or the burr is not spinning.

In section 6.1 we investigate how a hidden markov model (HMM) can be used to provide medium-term evaluation. Then in section 6.2 we compare the prediction accuracy of HMM and other classification algorithms.

6.1 Hidden Markov Model

A rule-based approach has been previously used to automatically evaluate performance within a temporal bone simulator (Kerwin et al. 2012). A simple approach is to establish a set of rules against which trainee behaviour is assessed. These rules can either be obtained from human experts or derived from labelled data. Obtaining rules from human experts can be difficult, as they may not be able to fully articulate their knowledge, which is often tacit in nature. It may also be difficult to translate expertise into quantitative rules, such as how much force to apply, how fast to drill, or what a perfect stroke should look like. Given these difficulties, we chose to use a data-driven approach to derive rules. To accomplish this, we computed a set of high-level statistical features for each simulator trial, from low-level simulator metrics. These high-level features included a variety of measures, such as average stroke force, average stroke speed, number of strokes per second, etc. We then used these statistical features to build a decision tree to predict expertise, as shown in Figure 5. Despite its simplicity, the decision tree achieved 82.3% cross-validation accuracy for stage 1. The decision tree shows that the vast majority of novices have average stroke force below 0.2068 newtons, which suggests that stroke force may be a good discriminator of expertise.

The decision tree shown in Figure 5 is accurate where stage summary data is concerned, but in order to provide online assessment, we need a model that can work with a sequence of low-level metrics. Rosen et al. (2001) used an HMM to evaluate laparoscopic surgery skills at the surgery level offline. Here we combine a sliding window with an HMM to deliver medium-term evaluation online.

An HMM consists of two parts: hidden states and observation metrics. Our previous analysis suggests that force may be a suitable observation metric for an HMM. In the context of surgical simulation, the sequence of forces applied by surgeons may reflect surgical techniques that are difficult to observe directly. Although unknown, these surgical techniques may be highly predictive of expertise. The hidden states of an HMM are able to model such unknown surgical techniques. Unlike laparoscopic surgery, temporal bone surgery does not have clear definitions of surgical gestures, hence the number of HMM hidden states cannot be derived directly from knowledge of the procedure. Instead we tried different numbers of hidden states, ranging from two to ten, and chose the number that achieved the best prediction accuracy for each stage.

HMM assumes that observations within each hidden state follow a distribution, which must be estimated. An inaccurate estimate of the distribution of observations will adversely affect the underlying mod-

el. To fit the training data to known distribution, we discretise the values to fit a binomial distribution. To do this we convert the continuous force magnitude values into two categories of "low" and "high" using the threshold of 0.2068 newtons, which is the split value of the decision tree in Figure 5.

An intuitive way to model expert behaviour is to create an HMM that represents an ideal surgical performance. However, deriving a model of ideal performance from collected data is difficult. Different simulation trials proceed at difference paces, thus creating a synchronisation problem. Furthermore, even expert surgeons use sub-optimal techniques occasionally. Instead of creating one HMM representing ideal performance, an expert HMM (E-HMM) and a novice HMM (N-HMM) were created from expert and novice force sequences respectively (using the Baum-Welch algorithm (Rabiner & Juang 1986)). The E-HMM and N-HMM models capture the differences between expert and novice technique. Once these two models were established, they could be used to evaluate the technique of an individual undertaking a new surgical procedure in the simulator. Put simply, if the technique of the individual carrying out a new surgical procedure is similar to the E-HMM, there is no need to change their behaviour. If however, their technique is closely aligned with the N-HMM model, then feedback should be provided to adjust the applied force.

However, providing this feedback is not simple. In order to provide real time feedback, the degree of expertise shown by an individual undertaking a new surgical procedure needs to be established over a period of time. Feedback should only be provided if the individual exhibits novice technique over a number of surgical strokes. Instead of using a fixed time duration for the purpose of HMM force sequence evaluation, we used drilling pauses to dynamically define the window size. These pauses could be easily identified by automatically determining when the force magnitude of the drill approached zero or the drill head was not spinning (i.e. it was not in use). Algorithm 1 explains the medium-term evaluation procedure.

```
Input: Force magnitude sequence;
while Not end of a surgery do
    while Drilling is not paused do
        F[i++]=Current force magnitude ;
    end
    eScore=likelihood(F, E-HMM);
    nScore=likelihood(F, N-HMM);
    F = null ;
    if nScore>=eScore then
        Display feedback message;
    end
end
```

Algorithm 1: Medium-term evaluation based on a sliding window of force magnitudes

The combination of HMM and drilling pauses provides dynamically assessment of expertise using at appropriate time intervals. If the force data stream within the sliding window accords with N-HMM, real time feedback is provided to the trainee. In our prototype system this feedback is presented in text form, letting the trainee know they can increase the force they are applying.

In this work we used only force as an observation metric for HMM medium-term evaluation, however the same approach can be applied to any other metric that is known to distinguish the technique of experts from that of novices.

6.2 HMM Evaluation Results

In order to validate the HMM evaluation model, we compared its prediction accuracy to other machine learning models, such as simple count (SC), logistic regression and OneR. The validation was carried out by passing trajectories to each model and obtaining the predicted classification for each trajectory. Validation was carried out in two ways, either by passing entire stage trajectories to each model, or by supplying sub-trajectories using the sliding window approach described in Algorithm 1. The simple count (SC) approach classifies the expertise of a sub-trajectory by counting the number of points with high force magnitude and low force magnitude.

Table 3 shows the results of the validation process. The accuracy of HMM approached 80% when supplied with the entire stage trajectory, and remained above 70% in the tests involving sub-trajectories. Overall, HMM achieved higher accuracy for all three stages compared to the other models. HMM may have achieved higher accuracy because it considers the order of force values in the given trajectory, whereas the other models do not.

The observed decrease in the accuracy of HMM in sub-trajectory tests compared to whole trajectory tests may be due to an assumption made in labelling the training data. To simplify the task of labelling, we labelled all sub-trajectories from expert trials as expert behaviour, while all sub-trajectories from novice trials were labelled as novice behaviour. However, in reality novices did not always perform badly and experts did not always perform perfectly. This simplification may have affected the accuracy of the assessment provided by the prediction models. Although the accuracy of HMM was lower in the sub-trajectory tests, it was still more reliable compared to the other approaches.

Stage 1 generally had higher accuracy than the other stages for all models. This result is consistent with the opinion of human experts, who advise that it is safe to apply relatively high force in this initial stage. Novices are generally more tentative, therefore they tend to apply lower force. This difference was not as pronounced in stages 2 and 3, where experts use lower force near anatomical structures such as the dura and sigmoid sinus.

Evaluating performance in real time is more difficult than delivering a summative evaluation at the end of each task. HMM achieved the highest accuracy for each stage when provided with the entire trajectory rather than a series of sub-trajectories. This highlights the trade-off between the accuracy and timeliness of the evaluation. Choosing an appropriate window size for evaluation is important in achieving optimal results. In previous work, window size was chosen either arbitrarily or experimentally (Murphy et al. 2003, Lin et al. 2006). In section 6.1 we proposed a dynamic approach to determine the optimal window size. This approach resulted in an average window size of 40 points when applied to our dataset, which corresponds to approximately 2.2 seconds. This is an acceptable delay for medium-term feedback.

7 Long-Term Evaluation

The aim of long-term evaluation is to assess the quality of the surgical end-product at the end of each stage. Sewell et al. (2007) demonstrated that the shape of the drilled bone is a good indicator of expertise in temporal bone surgery. Our contribution here is to deliver online assessment based on bone shape at the

Table 3: Medium-term expertise prediction accuracy

Stage	Trajectory	HMM	SC	Logistic	OneR
1	Whole	**87.09**	72.58	80.65	79.03
	Sub-traj	**77.69**	59.88	62.81	63.12
2	Whole	**79.03**	61.29	66.13	66.13
	Sub-traj	**72.32**	59.88	68.83	63.73
3	Whole	**79.03**	51.61	64.52	53.23
	Sub-traj	**71.33**	59.88	70.37	59.10

end of each stage to highlight incorrectly drilled regions of the bone. To conduct this type of assessment, we must be able to identify transitions from one stage to the next. Once a stage boundary has been detected, we can apply end-product analysis to evaluate performance as in Sewell et al. (2007). In this section, we begin by describing our stage prediction method and proceed to test its accuracy over the first three stages of mastoidectomy. Then we explain how long-term evaluation can be provided using bone shape analysis.

7.1 Stage Prediction

We applied a supervised approach combined with a continuous classification process to predict the current stage from the stream of low-level data provided by the simulator. The goal of this analysis was to derive informative features which can assist in stage prediction, and select an effective supervised classification model.

As discussed in section 3.2, participants were asked to inform the researchers whenever they completed a stage. This information provided the start time and end time for each stage, which were used to label each point in the training dataset. Stages were assumed to be sequential, so the end time of the current stage was also the start time of the next stage. In practice, the transition from one stage to the next was not always clear. For example, some surgeons preferred to fully expose the sigmoid sinus before they started exposing the dura, while others preferred to go back and forth between these structures. The task of labelling stages becomes difficult when stages are not sequential and have no clear boundaries. It is also harder to design models that can predict the current stage under these conditions.

Another challenge is that the amount of bone removed and the extent to which anatomical structures were skeletonised varied between simulator trials, even though the procedure was carried out on the same bone. This resulted in different bone shapes at the end of each stage, even for the same human expert. The lack of consistency in bone shape meant that it was difficult to apply a voxel-based approach to predict stage boundaries.

Appropriate feature selection is critical in achieving accurate stage prediction. One intuitive way to predict the current stage is bone shape analysis (i.e. a voxel-based approach). Surgeons remove different regions of the bone in each stage, therefore the record of removed voxels is an obvious candidate feature for stage prediction. We adopted a text classification approach, where different topics are assumed to have distinct word distributions which can be used to identify them (Berry 2004). Stages can be thought of as "topics" in the text classification approach, and voxel positions can be thought of as "words". A stage is assumed to have a distinct distribution of voxel positions associated with it.

Following the unigram assumption (Berry 2004) the probability of a stage given a vector of voxel positions can be computed by:

$$P(STG_i|Bone) \propto \prod_{j=1}^{m} P(Voxel_j|STG_i) \times P(STG_i) \quad (1)$$

where m is the length of the voxel position vector, and $Voxel_j$ is a boolean value indicating whether the jth voxel has been removed. STG_i represents the ith stage of the procedure, so the three stages of mastoidectomy are labelled as STG_1, STG_2, STG_3.

124674 unique voxels were removed during the 32 expert simulator trials in our dataset. Given the large number of unique voxels removed, it was not practical to build a classifier that uses each individual voxel as a feature. Hence, we needed to derive a smaller set of representative features that capture bone shape information, to reduce the dimensionality of the dataset.

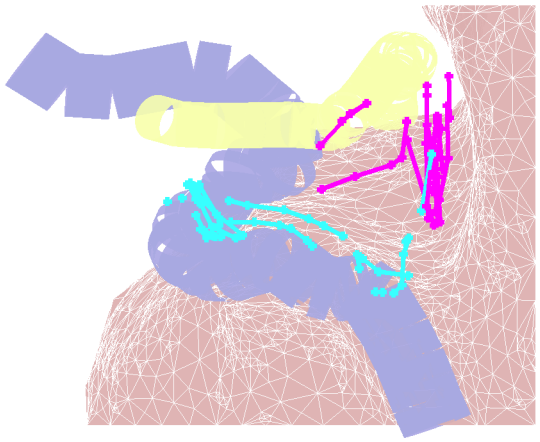
One common approach to reduce dimensionality is feature reduction. We applied information gain (Mitchell 1999) to select the top 12000 informative voxels in the dataset to be used as representative features, thus achieving around 90% reduction of the original feature space. This was the first of three feature selection approaches we examined.

The second feature selection approach we used involved compressing the historical information provided by the simulator and separating it from instantaneous drilling information. Instead of using the raw voxel positions as features, compressed historical information (HI) and instantaneous drilling information (IDI) were used to predict the current stage. Historical information included the total drilling time so far and the total number of voxels removed so far, because these measures provide an estimation of progress through the procedure. Instantaneous drilling information included the current X, Y, Z position of the drill, the X, Y, Z position of the irrigator, the force magnitude, and the distance of the drill and irrigator from each anatomical structure. These features suggest the current bone region being drilled, which is indicative of stage. Figures 6a and 6b show a sample of trajectories from two experts in stages 2 and 3. Although each surgeon's trajectories are slightly different, the bone regions being drilled in each stage are similar.

Finally, we chose a very simple feature selection approach as a baseline to compare the approaches discussed above. This approach used the depth of the current drill position as the only feature. This feature was thought to be an indicator of stage, because the depth of the drilled bone cavity increases across the stages of cortical mastoidectomy.

We carried out a comparison to determine which combination of feature selection and classifier yields the highest prediction accuracy. Different supervised models such as Navies Bayes, Decision Tree, Random Forest, SVM and KNN were applied on different feature sets to predict the stage. The accuracy of each model was evaluated using 10 cross-validation on the 32 expert trials in our dataset. Cross validation was performed using entire simulator trials rather than individual points, since training and testing within the same trial will result in over fitting.

Table 4 reports the prediction accuracy of each classifier in stages 1 to 3 using different feature selection approaches. All classifiers achieved highest accuracy on the historical/immediate drilling information features (HI&IDI) compared to the other feature selection approaches, with the exception of SVM-



(a) Expert 1

(b) Expert 2

Figure 6: Visualisation of drilling trajectories in stage 2 and 3 by two different surgeons. Cyan trajectories were performed in stage 2. Magenta trajectories were performed in stage 3.

Linear. Despite their simpliity, the HI&IDI features appear to provide a reliable approximation of stage. SVM-Linear may not have worked as well on these features due to possible violation of the linear separation assumption. 1NN and random forest provided the highest accuracy amongst the classifiers we tested (84.5% and 84.8% respectively). Although these models provided similar accuracy on the HI&IDI features, the random forest classifier is preferred for online performance evaluation due to faster prediction speed. Discriminative models like decision tree or random forest could not be practically applied to the voxel-based features, due to the high dimensionality of the voxel dataset. In addition, decision tree and random forest were not appropriate where a single feature was used, such as the drill depth.

Our results suggest that the HL&IDI feature selection approach combined with a random forest classifier is the best combination for online stage prediction. Furthermore, we observed that incorrect predictions occurred more frequently at stage boundaries. Stage boundaries may not always be clear even when a human expert is labelling them, therefore some prediction errors at stage boundaries are unavoidable. When providing long-term stage evaluation, it may be necessary to delay feedback to the user until sufficient evidence of a stage transition has been acquired.

Table 4: Stage prediction using different features and classifiers

Classifier	124674 drilled voxel	12999 drilled voxel	drill depth	HI&IDI
OneR	NA	69.6	53.1	59.7
Naive Bayes	78.0	69.3	38.1	78.3
1NN	NA	78.4	52.3	84.5
Decision Tree	NA	NA	NA	79.9
Random Forest	NA	NA	NA	**84.8**
SVM Linear	NA	NA	66.7	25.4

7.2 Surgical Expertise Classification

In the previous section, we showed how the current stage of cortical mastoidectomy can be predicted online. In this section, we explain how a voxel-based approach (Sewell et al. 2007) can be applied to evaluate the end product of each stage in our simulator. We assume that the drilled bone shape at the end of each stage differs between experts and novices. For each stage, we created an expert and a novice matrix of voxel positions and simulator trials, where each cell indicated whether a particular voxel was removed during a given trial. The rows (i.e. trials) of each matrix were instances and the columns (i.e. voxel positions) were features.

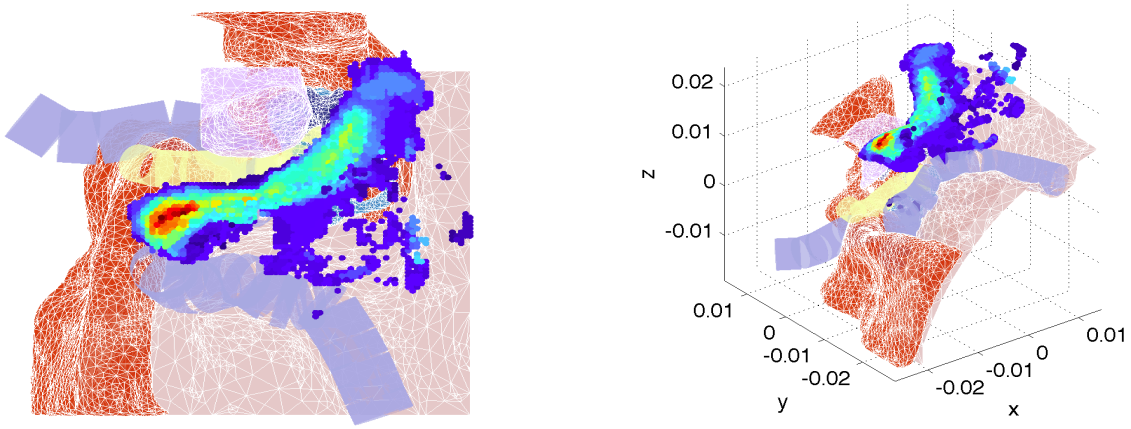
A large proportion of voxels was drilled by both experts and novices in most trials. These voxels are clearly not predictive of surgical expertise; if they were included in an expertise classification model they could adversely affect its accuracy. To select the most predictive voxel positions to use as features, we applied information gain (Mitchell 1999) to select the top 10% (around 8000) most informative voxels. We applied the simple One Rule (OneR) Classifier to our dataset and achieved an accuracy of 96.8%, which is slightly better than the accuracy of the Naive Bayes classifier used in Sewell et al. (2007). Given the high accuracy achieved by the OneR classifier, there was no need to pursue a more complex model.

Figure 7 plots the AUC values of the voxels removed during stage 1 based on the OneR classifier. The heat map shows the capacity of each voxel to differentiate expertise. Voxels that are highly predictive are those which were not removed by the majority of novices. We observe that novices usually failed to remove a sufficient amount of bone around the mastoid tip, as denoted by the deep red and orange voxels in Figure 7. Novices also tended to not thin the bone around the ear canal (shown as the green/light blue area) as much as experts. These results show that the proposed evaluation method can identify common differences in the bone regions drilled by experts and novices.

We have shown that our approach provides an accurate assessment of end product quality and identifies the most common bone regions neglected by novices. Providing novices with this information would bring their attention to these bone regions, thus allowing them to improve their performance.

8 Conclusion

We have proposed a framework to deliver three different levels of online performance evaluation in a temporal bone simulator: short-term, medium-term and long-term. We described each component of this



(a) X-Y view

(b) 3D view

Figure 7: Two views of stage 1 voxel predictive ability heat map in relation to anatomical structures. Dark red voxels are most predictive of expertise, while dark blue voxels are least predictive.

framework in detail and conducted experiments to validate the accuracy of our models. For short-term evaluation, we developed three metrics to assess immediate drilling technique. For medium-term evaluation, we described a process for choosing informative features from a range of low-level simulator data and showed how one such feature can be used to provide medium-term feedback on surgical technique. For long-term evaluation, we introduced methods to identify the current stage of a surgical procedure online, and we used voxel-based classification to assess the end product of each stage.

Many open questions remain to be explored in the quest to provide online performance feedback in a surgical simulator, such as how feedback should be presented to trainees, how should it be timed, and how much feedback should be provided.

In the future, we would like to investigate how the evaluation provided by the proposed framework can be presented to trainees as actionable constructive feedback.

References

Agus, M., Giachetti, A., Gobbetti, E., Zanetti, G. & Zorcolo, A. (2003), 'Real-time haptic and visual simulation of bone dissection', *Presence-Teleop Virt* **12**(1), 110–122.

Berry, M. W. (2004), *Survey of Text Mining I: Clustering, Classification, and Retrieval*, Vol. 1, Springer.

Bryan, J., Stredney, D., Wiet, G. & Sessanna, D. (2001), Virtual temporal bone dissection: a case study, *in* 'Proc. Conf. Visualization'01', IEEE Computer Society, pp. 497–500.

Darzi, A. & Mackay, S. (2002), 'Skills assessment of surgeons', *Surgery* **131**(2), 121.

Ericsson, K. A. (2004), 'Deliberate practice and the acquisition and maintenance of expert performance in medicine and related domains', *Acad Med* **79**(10), S70–S81.

Grober, E., Hamstra, S., Wanzel, K., Reznick, R., Matsumoto, E., Sidhu, R. & Jarvi, K. (2003), 'Validation of novel and objective measures of microsurgical skill: Hand-motion analysis and stereoscopic visual acuity', *Microsurgery* **23**(4), 317–322.

Hall, R., Rathod, H., Maiorca, M., Ioannou, I., Kazmierczak, E., O'Leary, S. & Harris, P. (2008), Towards haptic performance analysis using k-metrics, *in* 'Proc. HAID'08', pp. 50–59.

Hogan, N. (1987), 'Moving gracefully: quantitative theories of motor coordination', *Trends in Neurosciences* **10**(4), 170–174.

Hutchins, M., O'Leary, S., Stevenson, D., Gunn, C. & Krumpholz, A. (2005), 'A networked haptic virtual environment for teaching temporal bone surgery', *Studies in Health Technology and Informatics* **111**, 204–207.

Kerwin, T., Shen, H.-W. & Stredney, D. (2009), 'Enhancing realism of wet surfaces in temporal bone surgical simulation', *IEEE T Vis Comput Gr* **15**(5), 747–758.

Kerwin, T., Wiet, G., Stredney, D. & Shen, H.-W. (2012), 'Automatic scoring of virtual mastoidectomies using expert examples', *IJCARS* **7**(1), 1–11.

Lin, H., Shafran, I., Yuh, D. & Hager, G. (2006), 'Towards automatic skill evaluation: Detection and segmentation of robot-assisted surgical motions', *Comput Aided Surg* **11**(5), 220–230.

Mitchell, T. M. (1999), 'Machine learning and data mining', *Communications of the ACM* **42**(11), 30–36.

Murphy, T. E., Vignes, C. M., Yuh, D. D. & Okamura, A. M. (2003), 'Automatic motion recognition and skill evaluation for dynamic tasks', *in Eurohaptics* **2003**, 363–373.

O'Leary, S. J., Hutchins, M. A., Stevenson, D. R., Gunn, C., Krumpholz, A., Kennedy, G., Tykocinski, M., Dahm, M. & Pyman, B. (2008), 'Validation of a networked virtual reality simulation of temporal bone surgery', *The Laryngoscope* **118**(6), 1040–1046.

Rabiner, L. R. & Juang, B. H. (1986), 'An introduction to hidden markov models', *IEEE ASSp Magazine* .

Rosen, J., Hannaford, B., Richards, C. G. & Sinanan, M. N. (2001), 'Markov modeling of minimally invasive surgery based on tool/tissue interaction and

force/torque signatures for evaluating surgical skills', *IEEE Transactions on Biomedical Engineering* **48**(5), 579–591.

Sewell, C., Morris, D., Blevins, N., Barbagli, F. & Salisbury, K. (2007), 'Evaluating drilling and suctioning technique in a mastoidectomy simulator', *St Heal T* **125**, 427–432.

Sewell, C., Morris, D., Blevins, N., Dutta, S., Agrawal, S., Barbagli, F. & Salisbury, K. (2008), 'Providing metrics and performance feedback in a surgical simulator', *Comput Aided Surg* **13**(2), 63–81.

Zhou, Y., Bailey, J., Ioannou, I., Wijewickrema, S., O'Leary, S. & Kennedy, G. (2013), Constructive real time feedback for a temporal bone simulator, *in* 'Proc. MICCAI'13'.

sRADAR: A Complex Event Processing and Visual Analytics System for Maritime Intelligence

Naveen Nandan **Baljeet Malhotra*** **Daniel Dahlmeier**

SAP Research and Innovation, CREATE #14
University Town, 1 Create Way, Singapore 138602

naveen.nandan@sap.com baljeet.malhotra@sap.com d.dahlmeier@sap.com

Abstract

Maritime Intelligence is about empowering users in a port ecosystem with data and visual analytics to increase the efficiency and effectiveness of maritime operations. In this context, discovery and visualization of *ship domain violations* based on the analysis of trajectories generated by ships could serve important navigational and business purposes. Finding patterns of domain violations in a large trajectory database, however, is a non-trivial task, primarily due to the combinatorial nature of the problem. In this paper, we present a system, sRADAR, which models such trajectories, applies complex event processing on the data streams to identify such domain violations, performs analytics to derive useful insights into the recorded data, and helps visualizing the result of such geo-spatial analytics. To evaluate our proposal, we setup an Automatic Identification System for collecting real trajectories of ships arriving at the port of Singapore. We discuss some preliminary results on domain violations and the efficiency of our system using the real-time data collected by our system.

Keywords: Anomaly detection, Complex event processing, Maritime Intelligence, Visual analytics.

1 Introduction

Interactions between ships are an important area of research in marine navigation science and traffic engineering. According to the International Maritime Organization, 90% of the global trade is transported by sea, and as the global trade is increasing, ship collision avoidance will become more important than ever before. In this context, *ship safety domain* is a key concept that essentially prescribes an area around a ship that must not be intruded by any other ships for safe navigation. Ship safety domain is very critical, not only to enhance the navigation safety, but also to protect the lives of crew members and the serenity of marine environment. When a ship enters into the safety domain of another ship, it is generally called *ship brushing* or *domain violation*. Apart from jeopardizing human and financial losses, ship brushing may have serious implications on international relations.

*Corresponding Author.

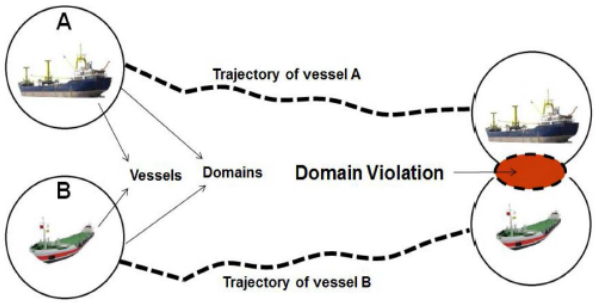


Figure 1: An illustration of a domain violation between two vessels (ships) named A and B.

A scenario of ship domain violation is depicted in Figure 1. The severity of ship brushing depends on the speed, size, and the orientation of the ships involved. Though domain violations are unavoidable in some situations, e.g., in narrow water channels where space is limited and may not always lead to accidents, domain violations can be serious and serve as an indicator of navigation safety. For instance, one can ask the following questions which are not only important for port authorities, but also for insurance companies and the vessel owners:

1. What types of vessels cause more violations than others?
2. In which geographical regions (that are bottlenecks) are violations more frequent?
3. Are there any particular seasons when the violations occur more frequent than others?

Answers to the above questions could be useful for situational awareness, port capacity planning, vessel profiling, and insurance claims.

In this paper, we describe sRADAR, a system that aims at automatically identifying complex events related to such questions and alerting various stake holders, such as port authorities, port operators, shipping companies, and insurances, of such domain violations. In order to provide further insight into the detected events, we perform complex data analysis on the collected geo-spatial and temporal data. Visualization and interaction can bridge the gap between computational data analysis methods, human reasoning, and decision-making processes, combining the strengths of both worlds (Riveiro 2011, Riveiro & Falkman 2011). On one hand, we take advantage of intelligent algorithms and the vast computational power of modern technology, such as in-memory databases, and on the other hand, we integrate human ability to comprehend information using intuitive methods of visualization for the derived

knowledge. To track the position of ships, we rely on the Automatic Identification System (IMO 2001) or AIS for short, which provides a rich source of data on ships' identification, trajectories, navigation status and others. The AIS technology is explained in more detail in Section 3.

The rest of the paper is organized as follows. Section 2 presents a discussion on the related work in the context of trajectory data mining and analytics. In section 3, we present an overview of the system and the details of our experimental setup. In the following section 4, we discuss the architecture and other technical details of the sRADAR system. Section 5 presents the details of the analytics visualization and reporting component of the system. The paper is concluded in section 6.

This paper discusses the preliminary approach taken towards building a solution for the maritime scenario. In building this prototype, we use and exploit some of the capabilities of in-memory database technology and front-end technology, HANA© and UI5© respectively, offered by SAP.

2 Related Work

Many researchers have extensively used maritime data specifically generated by the AIS for trajectory data mining (Li et al. 2010). The AIS data has also been used to study a spectrum of multidisciplinary problems such as maritime emission control (Perez et al. 2009), anomaly detection and risk assessments (Ristic et al. 2008, Laxhammar et al. 2009, Jakob et al. 2010), complex network analysis (Kaluza & et. al. 2010), and others (Malhotra et al. 2011).

Trajectory data mining is an emerging and rapidly developing topic in the area of data mining that aims at discovering patterns of trajectories based on their proximity in either a spatial or a spatio-temporal sense. As ships keep moving and continuously generate trajectory data, mining their trajectories plays an important role in maritime data management (Alvares et al. 2007, Andrienko et al. 2007, de Vries et al. 2010, Giannotti et al. 2007, Lee et al. 2008, Li et al. 2010). For instance, at a commercial port where hundreds of vessels may enter or leave the port or wait to do so, collision avoidance is of utmost importance (Statheros et al. 2008).

Trajectory data mining methods can also be employed to discover mobility, traffic and congestion patterns which can then be used for situational awareness (Alvares et al. 2007). Based on the movement patterns, trajectories (and the vessels spanning them) can be clustered into groups to access the interactions between them and their collision risks (Li et al. 2010, de Vries & van Someren 2008). Furthermore, models can be built based on the discovered patterns to engineer monitoring systems such as the one proposed in (Piciarelli & Foresti 2006), which can then be used to detect anomalies (e.g., the trajectory of a particular ship that is not adhering to the guidelines) in real time to warn the authorities immediately. In (Perez et al. 2009), the authors discussed the challenges of data management, analysis, and the problems of missing data in the AIS datasets while proposing potential methods for addressing the limitations. Yet another study (Malhotra et al. 2011), discusses the management of the AIS data streams from the perspective of privacy and access control.

The purpose of this paper is not to conduct an in-depth survey of works that deal with the above data mining techniques or to focus on trajectory analysis based on the AIS data. Rather we pay attention to the particular problem of domain violations that could serve various purposes. We also focus on visualization and reporting mechanisms that are important for analyzing the brushing incidents (in particular in port waters) for various stake holders of a port ecosystem as mentioned previously.

Field	**Description**
MMSI	Mobile Marine Service Identifier. 9 digit identifier for a vessel's AIS.
Navigation Status	For example, *under way using engine*, *at anchor*, *engaged in fishing*.
Rate of Turn	Turning rate in degrees per minute.
Speed	Speed of vessel in knots.
Longitude	Longitude position of vessel.
Latitude	Latitude position of vessel.

Table 1: Data fields in AIS message types 1 to 3.

3 AIS Overview

The Automatic Identification System (AIS) (IMO 2001)) is an automatic identification and tracking system for maritime vessels. The primary purpose of AIS is to improve navigation safety and avoid collisions between vessels. It allows ships and stations to broadcast messages that contain information about a ship's navigational status, position course and speed, among many others. The primary purpose for AIS is navigation safety and traffic control. The International Maritime Organization (IMO) requires all international voyaging ships with a gross tonnage of 300 or more tons and passenger ships to have an Automatic Identification System (AIS) installed.

AIS messages are broadcasted through VHS radio equipment. AIS messages can be received by other ships and by vessel traffic services stations in the vicinity (typically within 40 nautical miles). As a matter of fact, AIS messages can be received by anyone in range using an appropriate receiver and decoding hardware and software. The AIS protocol specifies 27 different message types which carry different information, for example ship navigation information, base station reports, information about ship size and dimensions, and search-and-rescue aircraft reports. In practice, we observe that AIS message types 1 to 3, which are position reports for navigation purposes, are used more frequently than the other message types. Table 1 shows some of the important fields contained in AIS message types 1 to 3. It is worth noting that AIS does not infer the position or navigational status of the ship automatically. AIS is merely a radio technology for broadcasting a ship's information, which has to be provided by other sensors on-board that ship, for example its GPS device.

3.1 Experimental Setup

The AIS data can be collected from an AIS communication network while using a multi-layer system typically consisting of a Complex Event (Stream) Processing Engine (CEP) (Arasu et al. 2003, Abadi et al. 2003) and a database system for processing, storing

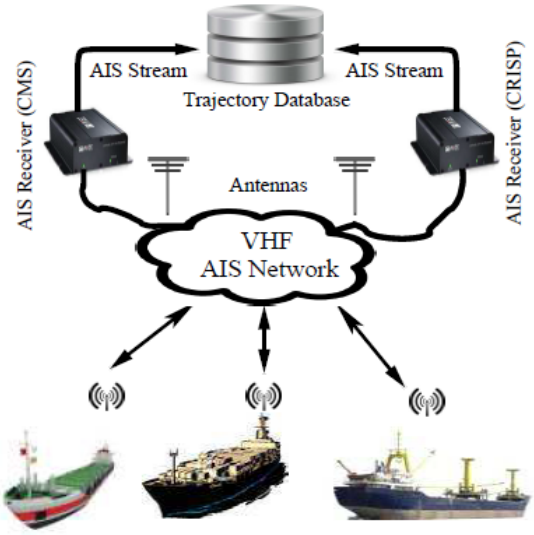


Figure 2: Experimental setup for data collection.

and analysis of the AIS data. At SAP Research, we have setup an AIS station to collect data from the ships arriving at the port of Singapore. The overall infrastructure of the setup is shown in Figure 2.

The captured AIS data is being processed and analyzed using a specialized complex processing engine (CEP). CEPs usually do not store data permanently, however, they allow access to traditional databases such as Oracle© and SQL Server© for data storage and processing purposes. We interfaced our CEP with HANA©, which is SAP's in-memory database appliance. Next, we describe the overall system architecture of the sRADAR system that we built for complex event processing based on AIS data and visual analytics for the purpose of maritime intelligence.

4 sRADAR Architecture

The proposed system consists of the following four main components:

1. AIS decoder,
2. rule engine,
3. real-time database,
4. visualization and reports.

Figure 3 shows the overall architecture of the system. We describe each component in turn.

4.1 AIS Decoder

The AIS messages are encoded in a binary format and hence, need to be decoded for further processing to identify information about the ships and their location. For this, we stream the AIS messages to an AIS decoder on the application server via UDP. The AIS decoder acts as a listener and as each message arrives, applies a decoding algorithm and stores the information to the database for future analysis. The decoder also forwards the ship information to the rule engine in order to perform on-the-fly computations, such as event detection.

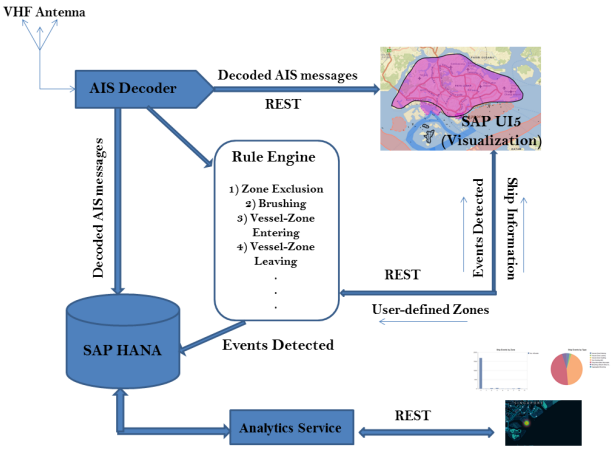


Figure 3: Overall architecture of the system.

4.2 Rule Engine

The rule engine has predefined rules, such as vessel brushing detection, vessel zone entering, vessel zone leaving, etc. As the decoded AIS messages arrive at the rule engine, each of the rules is applied to every message and if it matches the defined constraint, an event is fired that contains the relevant information about the detected event, for example the location, the identifier of the ships involved, and the time. The detected events are stored in the database which is used for further analytics as well as reporting and visualization.

4.3 Real-Time Database

The database layer for the system is implemented in SAP HANA© which is highly effective for temporal analysis of geo-spatial data. The database acts as both data store for the event detection phase and as the analytical engine for complex queries which are described in the following section. The initial system prototype makes use of JDBC calls to interact with the database, but going forward we are investigating on exploiting the geo-spatial capabilities of HANA©.

4.4 Visualization and Reports

The visualization layers for the prototype are built using a mix of SAP UI5© for the real-time ship movements, event reports and drawing zones for geo-fencing, CVOM©, bundled within UI5©, for analytical charts and Leaflet© for the event heatmap and derived event trajectory visualization. We considered visualization to be one of the most important components of the sRADAR system as visual analytics are not only important for reporting incidents, but also for situational awareness. To that end, we discuss the Visual Analytics in detail in the following section.

5 Visual Analytics

The system allows users to visualize data in different modes.

1. Real-time Ship Movement
2. Event Reports
3. Analytical Charts
4. Event Heatmap

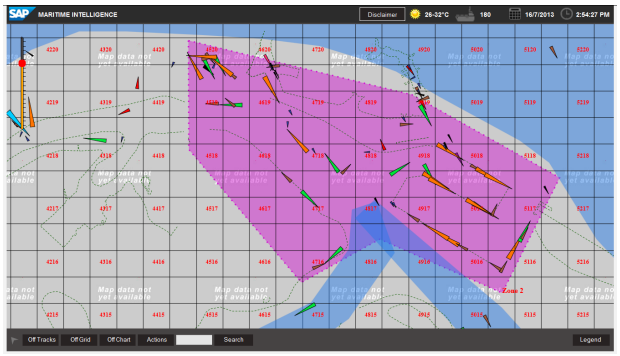


Figure 4: Real-time visualization of ships and interface to perform various actions such as drawing of zones.

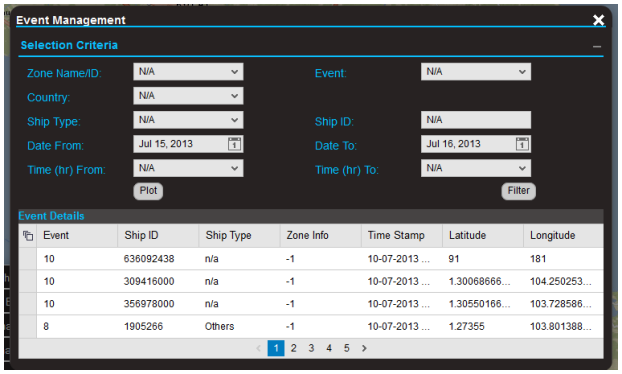


Figure 5: User interface for the selection of detected events based on various criteria such as ship/zone types, timeline, country of origin and so on.

5. Event Spatial Clustering
6. Brushing Event Trajectory

5.1 Real-time Ship Movement

The decoded AIS messages contain information about the position of ships. As a new AIS message arrives, the system identifies the source and checks if that specific ship has been identified before. If the ship has not been detected before, we store the information in the database along with its latest position. In case the ship already exists in the database, a comparison is made with its previous reported position and if this has changed, the latest position is updated. In both cases, the updated position is returned to the frontend which in turn plots the ship on a map. The mapping API used for this real-time plot is provided by ESRI©. Apart from plotting objects on the map, the front-end can also be used to draw and define zones for geo-fencing. This is illustrated in Figure 4.

5.2 Event Reports

The events triggered are stored in the database with information on time of occurrence, event type, position of the ship, details of other ships involved in the event, zone information, if any, and so on. These are presented to the user in the form of tabulated reports. This service is available on demand i.e., the query is triggered when user requests through the UI and populated in tabular format accordingly. An example of the event report UI is shown in Figure 5.

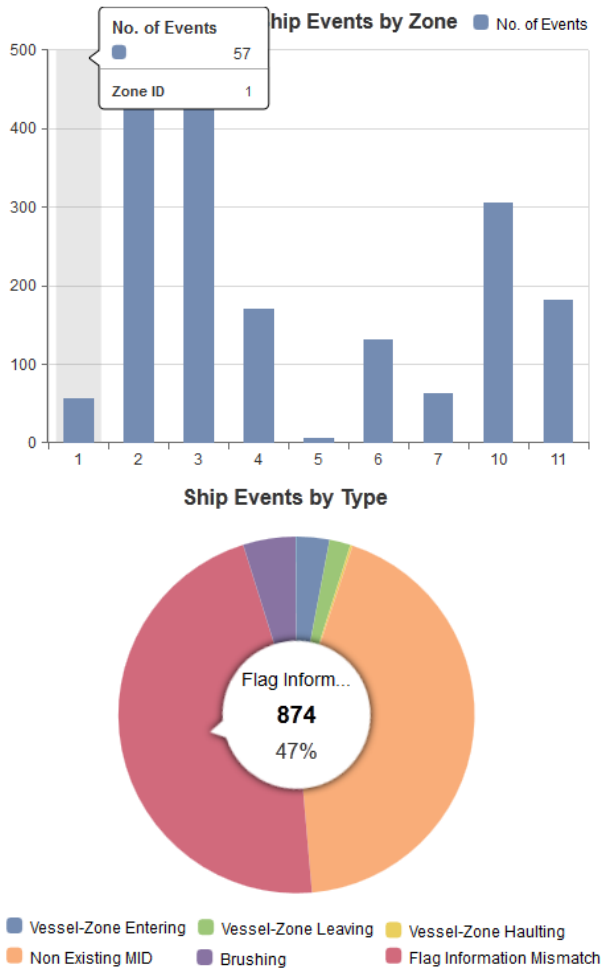


Figure 6: Analytical charts that present ship events by zone and event types.

5.3 Analytical Charts

As the decoded AIS data grows in volume, the amount of events detected also increase proportionally. This makes it difficult to query for all the events and represent them in a report format. For this reason, we make use of the interactive charting library CVOM© to present such information in a more intuitive fashion. The queries that populate the charts aggregate events by ship type, event type, etc. and help the users to get a broader insight into the recorded data. This would in turn help the user to identify outliers and move into investigating further based on the knowledge gained.

The frontend makes use of REST calls to the backend that fire the query every few seconds, thereby, ensuring real-time updates. The data is populated on the charts only if there is a change in state of the result, making the process of calculation and retrieval quite efficient. Figure 6 shows examples of analytical charts from our system.

5.4 Event Heatmap

When a large number of data points are involved that are geo-spatially distributed, plotting each of them was found to be inefficient. After investigating further, using heatmaps was found to be a common method of presenting the density of an aggregate of spatially distributed points. We make use of the heatmap.js javascript library which is used as

Figure 7: Visualization of a heatmap based on detected events.

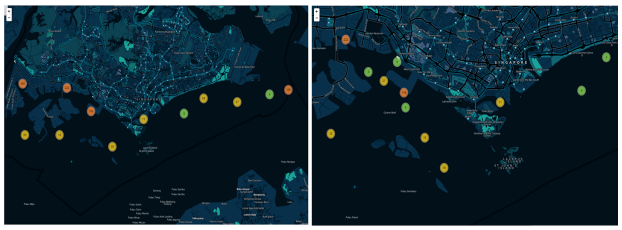


Figure 8: Visualization of spatially clustered events based on various zoom levels.

a layer atop leaflet.js, a mapping solution that integrates with OpenStreet© maps.

The user is allowed to select a specific event type, on which a REST call is made to query the database and retrieve a distribution of all points in space of that specific event. The returned points are plotted as a heatmap with the color intensity depicting the density of points returned. An example heatmap is shown in Figure 7.

5.5 Event Spatial Clustering

Another method to visualize a large number of spatially distributed points was to make use of the marker clustering API of Leaflet. By using this, we are able to cluster the number of events and display them as a collective point with an aggregated count, which in turn is controlled by the zoom level of the map. When the user zooms in, the clusters breakdown and render into individual points or smaller clusters as shown in Figure 8.

5.6 Brushing Event Trajectory

Brushing between ships was identified to be one of the major concerns at busy ports such as Singapore. We define a prior rule to detect brushing events between ships which is triggered when two ships violate the space constraint. The constraint is usually defined as the minimum distance that is to be maintained between any two ships. This minimum distance varies from ship to ship and takes into account the type and dimensions of the ship. An event is triggered and stored in the database for violation of such constraints with information on the ships involved, the time when the event occurs, the information on the zone in which the event occurred, if any, and the position of the ship. For every brushing event, two records are generated, one for each ship. Also, based on the duration of brushing there could be multiple records of the same event that is being recorded with different timestamp and position logs. Initially we present to the user a report of all such logs in tabular format, which was useful when the number of brushing events detected were quite low. But as more and more of such events were detected, the number of such records in the database grew large and representing them in tabular format was not always useful. For instance,

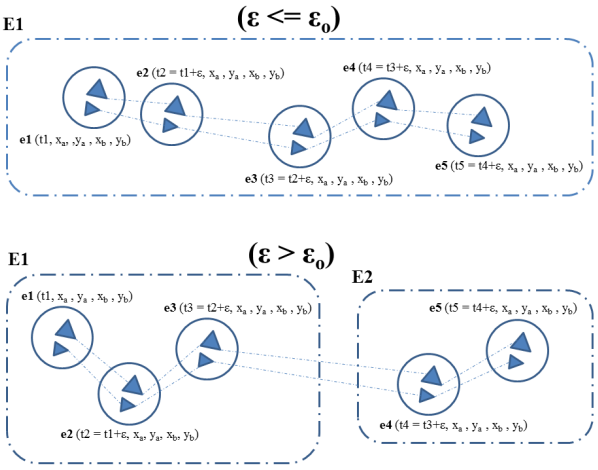


Figure 9: A scenario depicting multiple brushing events and a method to aggregate them based on time windows.

if our system detects brushing events between ship A and ship B for a period of 1 hour, based on the frequency of AIS messages received from these ships, we would have multiple entries of the same event with a variation in timestamp and position of the ships. Although this was accurate and represented the actual situation, from the user's perspective, it would not be useful to know that this event occurred multiple times.

To tackle this problem, and present the information to the user in a more intuitive form, we developed an algorithm to aggregate such events based on a time interval. A query is generated to aggregate all events of this type for each ship based on a time window. The definition of the interval for the time window is set based on studying the frequency of such events. Figure 9 represents the definition of the threshold between consecutive events. Here, *e1, e2, e3, e4, e5* are the event records as detected by the rule engine and stored in the database. The algorithm aggregates these events as a single brushing incident E1 or as two different brushing incidents *E1* and *E2* based on the variation of the threshold ϵ_0. In this way, multiple entries of the same brushing event that are recorded can be reported to the user as an aggregated event. One of the methods to present this information to the user is as before, using tabulated reports with the following attributes: *Ship1, Ship2, Start_Time, End_Time, Brush_Duration, Start_Position, End_Position.* Another method is to directly visualize the trajectory of the incident on a map. This is done by returning the result in GeoJSON standard format to the frontend Leaflet layer.

6 Conclusion

The system developed serves as a research prototype for maritime intelligence using AIS broadcast messages recorded from the ships around the port of Singapore, one of the busiest ports in the world. The capabilities of technologies such as in-memory databases can be further exploited for analytics on such large data volumes to detect anomalies in maritime traffic. In the future, the system will run analytics on multiple other data sources that provide accurate position information such as radar, surveillance cameras, etc. As the system for collecting and analyzing AIS messages is in place, further research focus

would be in the direction of applying and developing learning algorithms to detect domain violations automatically, rather than handcrafting them. Also, the developed prototype is currently being evaluated by various stakeholders and further changes would be made based on the feedback.

7 Acknowledgment

We would like to thank the Economic Development Board of Singapore and National Research Foundation of Singapore for partially supporting this research. Our thanks are also due to SAP COIL and SAP HANA Solutions teams in Singapore for providing infrastructure support.

References

Abadi, D. J., Carney, D., Çetintemel, U., Cherniack, M., Convey, C., Lee, S., Stonebraker, M., Tatbul, N. & Zdonik, S. B. (2003), 'Aurora: a new model and architecture for data stream management', *The VLDB Journal* **12**(2), 120–139.

Alvares, L. O., Bogorny, V., de Macedo, J. A. F., Moelans, B. & Spaccapietra, S. (2007), Dynamic modeling of trajectory patterns using data mining and reverse engineering, *in* 'Proc. of the Int. Conf. on Conceptual Modeling (ER2007)', pp. 149–154.

Andrienko, G., Andrienko, N. & Wrobel, S. (2007), 'Visual analytics tools for analysis of movement data', *ACM SIGKDD Explorations* **9**(2), 38–46.

Arasu, A., Babcock, B., Babu, S., Datar, M., Ito, K., Nishizawa, I., Rosenstein, J. & Widom, J. (2003), Stream: The stanford stream data manager, *in* 'Proc. of SIGMOD'.

de Vries, G. K. D., van Hage, W. R. & van Someren, M. (2010), Comparing vessel trajectories using geographical domain knowledge and alignments, *in* 'Proc. of the ICDM Int. Workshop on Spatial and Spatiotemporal Data Mining (SSTDM)', pp. 209–216.

de Vries, G. & van Someren, M. (2008), Unsupervised ship trajectory modeling and prediction using compression and clustering, *in* 'Proc. of the Belgian-Netherlands Conf. on Artificial Intelligence', pp. 7–12.

Giannotti, F., Nanni, M., Pinelli, F. & Pedreschi, D. (2007), Trajectory pattern mining, *in* 'Proc. of the ACM Int. Conf. on Knowledge Discovery and Data Mining (KDD)', pp. 330–339.

IMO (2001), Guidelines for the onboard operational use of shipborne automatic identification systems (ais), as amended by itu-1371. resolution a.917(22), Technical report, International Maritime Organization.

Jakob, M., Vanek, O., Urban, S., Benda, P. & Pechoucek, M. (2010), Employing agents to improve the security of international maritime transport, *in* 'Proc. of the Int. Conf. on Autonomous Agents and Multiagent Systems', pp. 29–38.

Kaluza, P. & et. al. (2010), 'The complex network of global cargo ship movements', *The Journal of the Royal Society Interface* **7**, 1093–1103.

Laxhammar, R., Falkman, G. & Sviestins, E. (2009), Anomaly detection in sea traffic - a comparison of the gaussian mixture model and the kernel density estimator, *in* 'Proc. of the 12th Int. Conf. on Information Fusion', pp. 756–763.

Lee, J., Han, J. & Li, X. (2008), Trajectory outlier detection: A partition-and-detect framework, *in* 'Proc. of the Int. Conf. on Data Engineering (ICDE)', pp. 140–149.

Li, Z., Lee, J.-G., Li, X. & Han, J. (2010), Incremental clustering for trajectories, *in* 'Proc. of Int. Conf. on Database Systems for Advanced Applications', pp. 32–46.

Malhotra, B., Tan, W.-J., Cao, J., Kister, T., Bressan, S. & Tan, K.-L. (2011), Assist: Access controlled ship identification streams, *in* 'Proc. of the 19th ACM SIGSPATIAL Int. Symposium on Advances in Geographic Information Systems (GIS)', pp. 485–488.

Perez, H., Chang, R. & Billings, R. (2009), Automatic identification systems (AIS) data use in marine vessel emission estimation, *in* 'Proc. of the 18th Annual Int. Emission Inventory Conference'.

Piciarelli, C. & Foresti, G. L. (2006), 'On-line trajectory clustering for anomalous events detection', *Pattern Recognition Letters* pp. 1835–1842.

Ristic, B., Scala, B. L., Morelande, M. & Gordon, N. (2008), Statistical analysis of motion patterns in AIS data : Anomaly detection and motion prediction, *in* 'Proc. of the 11th Int. Conf. on Information Fusion', pp. 40–46.

Riveiro, M. & Falkman, G. (2011), The role of visualization and interaction in maritime anomaly detection, *in* 'IS&T/SPIE Electronic Imaging', International Society for Optics and Photonics, pp. 78680M–78680M.

Riveiro, M. J. (2011), Visual analytics for maritime anomaly detection, PhD thesis, Örebro University.

Statheros, T., Howells, G. & McDonald-Maier, K. (2008), 'Autonomous ship collision avoidance navigation concepts, technologies and techniques', *The Journal of Navigation* **61**, 129–142.

Analysing Twitter Data with Text Mining and Social Network Analysis

Yanchang Zhao

Intent Management and Analytics Section,
Risk Analysis and Monitoring Branch,
Department of Immigration and Citizenship, Australia
Email: yanchang.zhao@immi.gov.au

Abstract

Twitter, as one of major social media platforms, provides huge volume of information. A research project was performed recently in the Analytics Research Weeks in the Australian Department of Immigration and Citizenship (DIAC) to analyse Twitter data and study the feasibility of using them to better understand and improve DIAC business. It studied the official DIAC Twitter accounts in two ways. First, DIAC tweets are studied with text mining techniques to identify topics and their variations over time. And then, DIAC followers are analysed with social network analysis to find out how tweets spread over Twitter network. The methodology and techniques used in this work are general and can be easily applied to analysis of other Twitter accounts.

Keywords: Twitter, social media, text mining, topic modelling, social network analysis

1 Introduction

Twitter[1] is one of the most popular social media websites and has been growing rapidly since its creation in March 2006. As of March 2013, there were over 200 million active users, creating over 400 million tweets every day (Twitter Blog 2013). An advantage of Twitter is that it is real time and information can reach a large number of users in a very short time. As a result, there has been an increasing trend to analyse Twitter data in past years. One very early work on Twitter data analysis was published in 2007, which studied the topological and geographical properties of Twitter's social network and analysed user intentions at a community level (Java et al. 2007). It is followed by Kwak's work that analysed Twitter's follower-following topology, user ranking and top trending topics (Kwak et al. 2010). There were a lot of other publications on this topic recently (Bakshy et al. 2011, Poblete et al. 2011, Szomszor et al. 2011, Zubiaga et al. 2011, Bae & Lee 2012, Lehmann et al. 2012, Lu et al. 2012, Pennacchiotti et al. 2012, Stringhini et al. 2012, Tao et al. 2012, Chang et al. 2013).

However, there is little work reported on social media data analysis in government agencies. To analyse social media data and study the feasibility of using them to better understand and improve the business of the Australian Department of Immigration and Citizenship (DIAC)[2], a research project was performed recently in the Analytics Research Weeks in DIAC, with Twitter as a start point. This work studies the official DIAC Twitter accounts in two ways. At first, DIAC tweets are analysed with text mining techniques to identify topics and their variations over time. And then, DIAC followers are studied with social network analysis to find out how tweets spread over Twitter network. All analysis in this work was done with R[3] (R Core Team 2013) and several R packages.

The rest of this paper is organised as below. Section 2 introduces the Twitter data used in this work and also shows how to get data from Twitter. Tweets are then analysed with text mining and topic modelling in section 3. In section 4, DIAC followers are investigated with social network analysis and it demonstrates how tweets spread over Twitter network. Conclusions and discussions are provided in the last section.

2 Twitter Data

2.1 DIAC Twitter Accounts

There are two official Twitter accounts owned by DIAC, *@SandiHLogan* and *@DIACAustralia*. *@SandiHLogan* is an account of the DIAC National Communications Manager and used to be official account of DIAC. Its tweets were used in the analysis of text mining in section 3. In December 2012, a new dedicated account, *@DIACAustralia*, was created, and its data were used in social network analysis in section 4.

2.2 Getting Twitter Data

To pull data from Twitter, the *TwitterR* package[4] (Gentry 2013) was used, which provides an interface to the Twitter web API. In addition, the Twitter API v1's *"GET statuses/:id/retweeted_by/ids"*[5] was also used, together with the *RCurl* package (Lang 2013), to find out how tweets were retweeted. Because Twitter API v1 stopped in June 2013 and was replaced with v1.1, readers need to use *"GET statuses/retweeters/ids"* or *"GET statuses/retweets/:id"* provided in Twitter API v1.1[6] to trace the path on which a tweet was retweeted.

[1] http://www.twitter.com

[2] http://www.immi.gov.au

[3] http://www.r-project.org

[4] http://cran.r-project.org/web/packages/twitteR

[5] https://dev.twitter.com/docs/api/1

[6] https://dev.twitter.com/docs/api/1.1

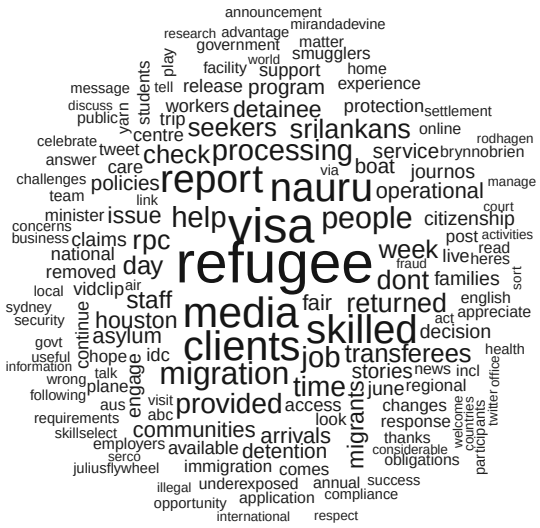


Figure 1: Word Cloud

3 Text Mining of Tweets

Tweets of *@SandiHLogan* extracted on 18 December 2012 were used in this analysis. At that time, it had over 6,000 Tweets and 5,493 followers, and followed 173 users. In this analysis, there were 1,409 tweets from 31 March to 9 December 2012, which were collected and analysed with text mining and topic modelling techniques to find topics and events over time. After that, topics and events were aligned with time series of visa applicants to find relationship between them.

The tweets were pulled from the Twitter website with R and the *twitteR* package (Gentry 2013), and then were processed and analysed with the *tm* package (Feinerer & Hornik 2013) and the *topicmodels* package (Grün & Hornik 2011), by following an example on text mining of Twitter data (Zhao 2013). At first, the text were cleaned by removing punctuations, numbers, hyperlinks and stop words, followed by stemming and stem completion. In addition to common English stop words, some other words, such as "Australia", "Aussie", and "DIAC", which appeared in most tweets, were also removed. After that, a term-document matrix was bulit and used for modelling. The results of text mining are shown in Figures 1, 2 and 3.

3.1 Frequent Terms and Term Network

Based on the term-document matrix, the frequency of terms was derived and plotted as a word cloud (see Figure 1), using the *wordcloud* package (Fellows 2013). In the word cloud, the more tweets a term appeared in, the bigger the term is shown. The figure shows that there were many tweets on refugee and skilled migration, and also some tweets on Sri Lankan and the Nauru Regional Process Centre (RPC).

Still based on the term-document matrix, a network of terms were built according to their co-occurrences in tweets, using the *Rgraphviz* package (Gentry et al. 2013). The result is shown in Figure 2. The vertices stand for terms, and the connections for the co-occurrences of terms in same tweets. A thick line indicates that the two corresponding terms appeared together in many tweets. The figure indicates that there are some tweets on transferees and the Nauru RPC, some on asylum seekers, some on a job fair in Houston, and some on a refugee week in June.

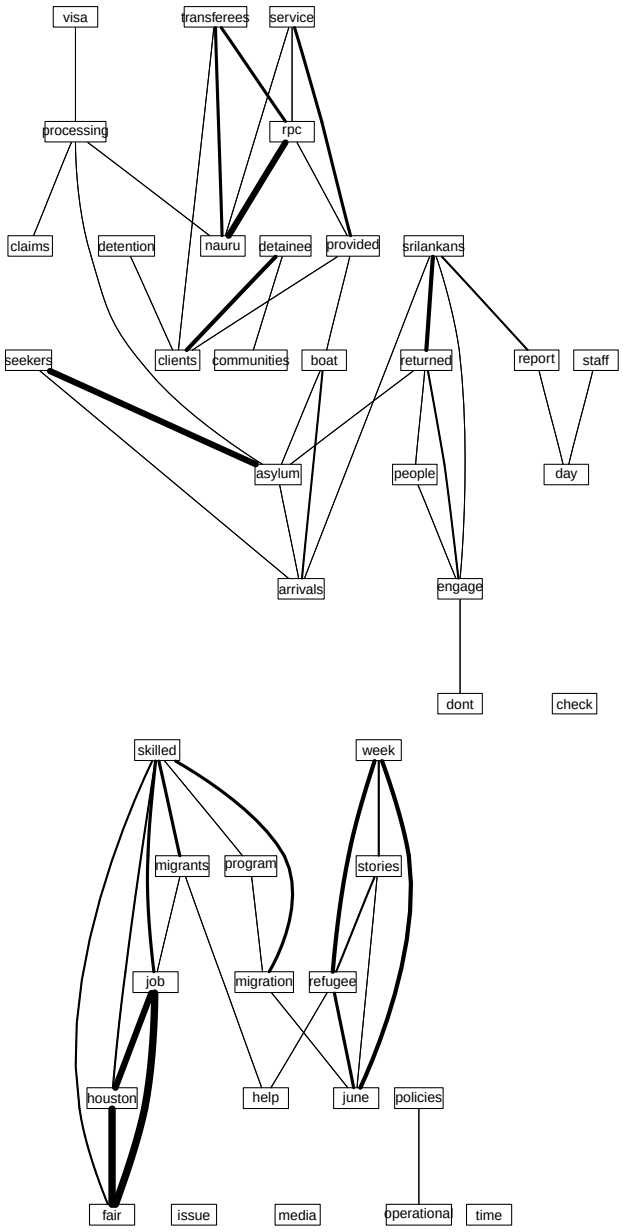


Figure 2: Term Network

3.2 Topic Modelling

After the above text mining of frequent terms and their connections, topics in tweets were studied. Topics were identified from tweets with the LDA (Latent Dirichlet Allocation) model (Blei et al. 2003) provided in the *topicmodels* package (Grün & Hornik 2011). Then the number of tweets in each topic was counted and plotted as a stream graph to show temporal variation of topics (see Figure 3).

Figure 3 can be taken as a stacked density plot of count of tweets on every topic, where the volume of tweets on a topic is shown with band width. Note that in the stream graph, the topics in legend are in the reverse order of those in the graph itself. That is, the first band stands for topic *staff, visa, media, changes* (the last one in legend), the second shows *skilled, job, fair, dont* (the 2nd last in legend), and so

Figure 4: Twitter Follower Map

on. The figure shows that there were many tweets on refugee (see the 4th & 5th topics) in May and June 2012. In addition, the 2nd topic from the bottom in the stream graph shows many discussions on the Nauru RPC, transferee and Sri Lankan in November 2012.

A possible application is to align the stream graph with time series, such as the number of visa applications, to find out any relationship between them, and even further to predict the trend in the number of visa applications based on changes in topics and produce alerts for significant events and emerging topics.

4 Social Network Analysis of Twitter Followers and Retweeting

Following text mining of tweets in last section, this section studies DIAC Twitter account in the approach of social network analysis. This analysis focused on who the followers of *@DIACAustralia* were and how its tweets were retweeted by them and spread over the Twitter network. More specifically, its followers were investigated and shown on a word map, top retweeted messages were identified, and the spread of the above tweets and their potential impact was studied.

The Twitter data of *@DIACAustralia* and its followers were used in this social network analysis. This account started from December 2012, and data of it on 24 May 2013 were extracted. On that day, it had 118 Tweets and 1,493 followers and followed 91 users. The techniques involved are geomap, social network analysis and text mining, and the tools used are R, packages *twitteR* (Gentry 2013) and *igraph* (Csardi & Nepusz 2006), and the Twitter API. More details about how to extract Twitter data are provided in section 2.2.

4.1 Followers

Locations of followers were first checked. With the location information of Twitter accounts, a map of Twitter followers (see Figure 4) was produced using a *twitterMap* function[7] authored by Jeff Leek. The lines in the figures show the connections between DIAC (in Canberra) and its followers. Note that it shows only followers who have provided locations in their Twitter account descriptions.

Next, followers were categorised based on descriptions provided in their Twitter account, which give a short piece of information about owner of the account. Alternative ways to categorise followers are categorising based on their tweets, followers or friends (i.e., users that they follow). With text mining again, a term-document matrix was built for user descriptions of followers and then plotted as a term network (see

[7] http://biostat.jhsph.edu/~jleek/code/twitterMap.R

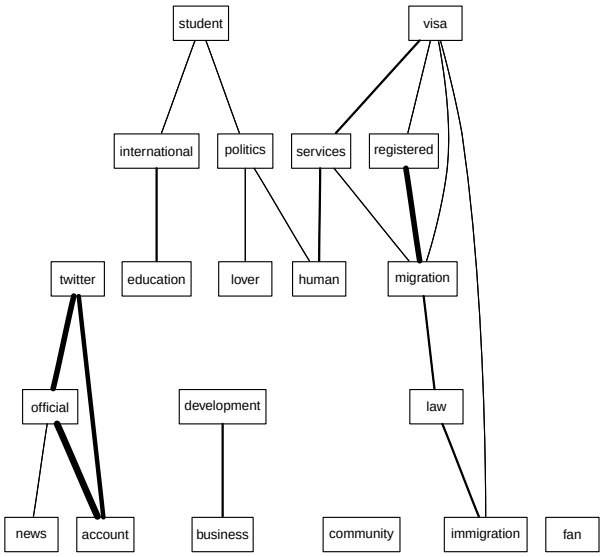


Figure 5: Term Network of Follower Descriptions

Figure 5). The figure shows some categories of followers. The subgraph on the bottom-left corner indicates some followers are official Twitter account of government departments or organisations. The top-left part shows that there are some followers who focus on international students and education. Another group of followers, shown in the right part of the figure, are registered migration agents, who provide visa and legal services.

After that, active and influential users among DIAC followers were inspected. Information of all followers of *@DIACAustralia* were collected, including when the accounts were created, how many tweets they had, how many users (i.e., followers) followed them, and how many users (i.e., friends) they were following. For every follower, the ratio of number of followers to number of friends was calculated, because an influential user tends to have a lot of followers but does not follow too many users. The average number of tweets of every follower per day is also calculated, which shows how active a user is. Based on the above numbers, a scatter plot of top followers were produced as Figure 6. Note that personal names are anonymised for privacy reasons. The figure shows that, the most influential and active followers largely fall into two categories.

- Media and correspondents: *7News Yahoo!7*, *The Morning Show* (on Channel 7) and *lia281* (an ABC Papua New Guinea Correspondent); and
- Government agencies and officials: *lat250* (quarrelling quandaries of question time, Parliament House Canberra), *ACPET* (national industry association for education and training), *AEC* (Australia Electoral Commission), *DFAT* (Department of Foreign Affairs and Trade, Australia), *Australian Customs*, *Sandi Logan* (DIAC National Communications Manager), *Aus745* (Australian Ambassador to US), etc.

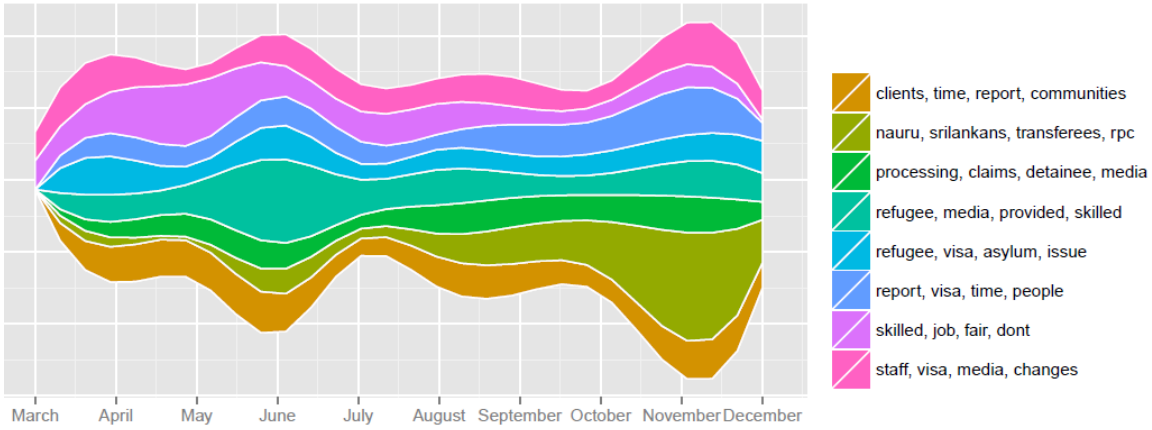


Figure 3: Stream Graph of Topics

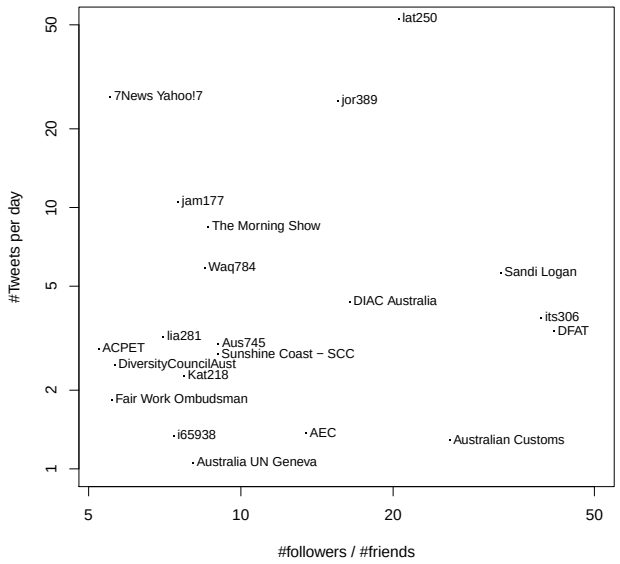


Figure 6: Top Influential and Active Followers

4.2 Tweets Most Retweeted and Their Spread Over Twitter Network

After studying who and where the followers were, this analysis presents what the most influential tweets were about and how they were retweeted via the Twitter network.

Figure 7 shows tweets that have been retweeted more than 10 times. The horizontal axis stands for time and the vertical for the number of times that a tweet was retweeted. The most retweeted one was a tweet on 9 January 2013 about Australia's low unemployment rate of migrants, and it was retweeted over 40 times.

The second most retweeted message was tweeted on 11 January 2013: "Are you an international student in Australia? New post-study work arrangements are being introduced http://t.co/g8c4yPlT". The URL in the tweet links to a post on the DIAC Migration Blog[8]. This tweet was investigated further to find out how it spread on Twitter network and how many users it reached. The retweeting data were extracted with the Twitter API mentioned in section 2.2, and the analysis were preformed with the *igraph* package (Csardi & Nepusz 2006).

Figures 8 and 9 show how the message were retweeted by DIAC followers and spread over Twitter network. Similar to Figure 6, personal names are anonymised and moreover, personal photos are replaced with an egg icon. Figure 8 shows a network of Twitter users who have retweeted the above message. Based on Figure 8, Figure 9 shows followers of those users and illustrates how many users the message might have reached. The message was retweeted by *DFAT*, who had 14,662 followers at that time, and then retweeted again by its followers, such as *deb338* (an Editor of ABC News Melbourne, 396 followers), *Austraining Int.* (352 followers) and *mym278* (a Policy Analyst at Chamber of Commerce & Industry, Queensland, 344 followers). The message was also retweeted by *Australia in UK* (Australian High Commission in UK, 1,129 followers) and then by *Dub706* (Australian Ambassador to Indonesia, 3,586 followers), who passed it on to his followers. In addition, it was also retweeted by other immediate followers of *@DIACAustralia*, such as *Ohj787* and *Sma346*, who were Immigration Officers at universities and education organisations. The above analysis shows that the message has potentially reached over 23,000 Twitter users.

5 Conclusions and Future Work

This paper presents a preliminary research on analysing DIAC Twitter data in approaches of text mining and social network analysis. With text mining of tweets, topics and their variations over time have been identified. Twitter followers have been analysed and the spread of tweets over Twitter network has been studied. With some initial interesting results identified, this research will be further studied in future research projects.

The methodology used in this work is general, the tools used are open-source software, and the data used in this work are publicly available on Twitter. Therefore, readers can easily replicate the analysis and apply it to Twitter accounts that they are interested in.

This research can be extended by analysing text from more Twitter accounts, analysing social network between them and their followers, and developing an effective method to find relationship between topics/events and variations in time series, e.g., the number of visa applicants, approvals, arrivals, departures and visa processing time.

It can also be extended further to investigate how messages spread, estimate their impacts and generate alerts. It would also be interesting to analyse tweets

[8] http://migrationblog.immi.gov.au

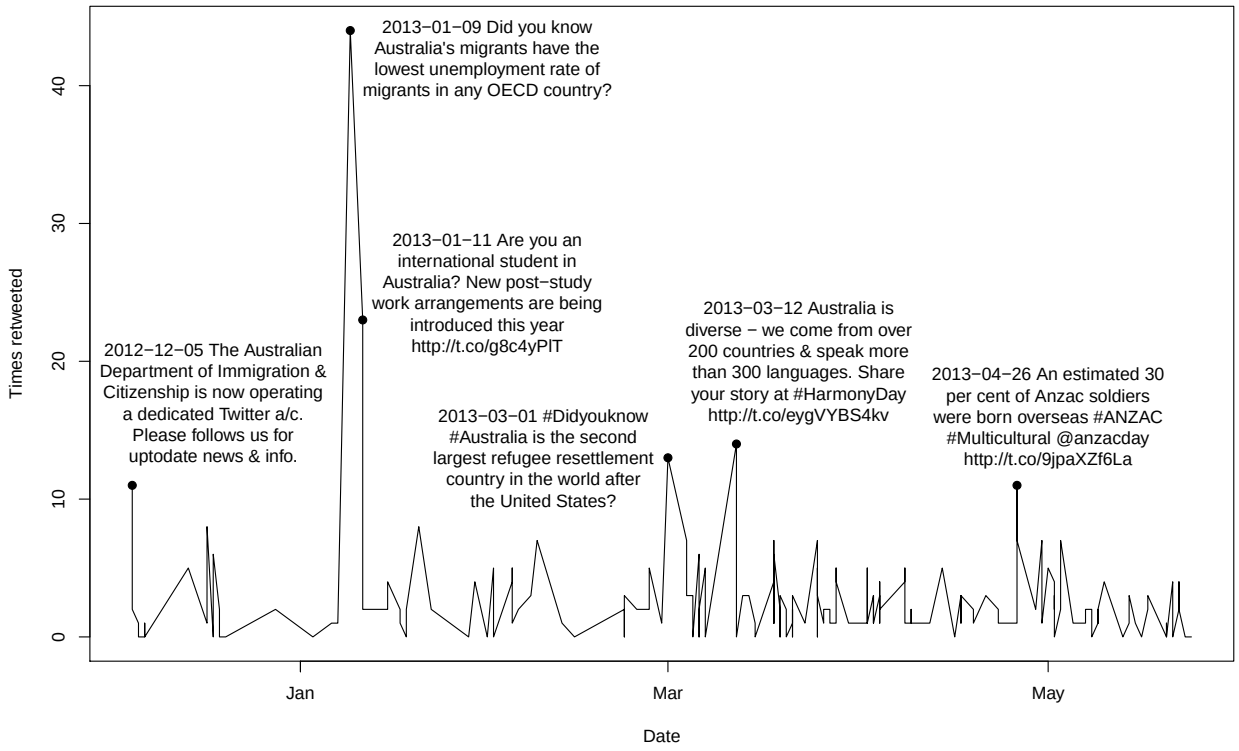


Figure 7: Tweets Most Retweeted

on specific topics based on Twitter hashtags, such as "#AustralianVisa" and "#refugee", and preform sentiment analysis for new legislations and policies.

Another possible future work is to study social network with data from multiple social media platforms, such as Twitter, Facebook[9] and Google+[10], and investigate interactions between government agencies, migration agencies and individuals.

Acknowledgements

I'd like to thank the Intent Management and Analytics Section, Department of Immigration and Citizenship for providing an opportunity to do this work in research weeks. I'd also like to thank Greg Hood from the Department of Agriculture, Fisheries and Forestry, and Fraser Tully and John D'arcy from the Department of Immigration and Citizenship for sharing their code for graph plotting.

References

Bae, Y. & Lee, H. (2012), 'Sentiment analysis of twitter audiences: Measuring the positive or negative influence of popular twitterers', *J. Am. Soc. Inf. Sci. Technol.* **63**(12), 2521–2535.
URL: *http://dx.doi.org/10.1002/asi.22768*

Bakshy, E., Hofman, J. M., Mason, W. A. & Watts, D. J. (2011), Everyone's an influencer: quantifying influence on twitter, *in* 'Proceedings of the fourth ACM international conference on Web search and data mining', WSDM '11, ACM, New York, NY, USA, pp. 65–74.
URL: *http://doi.acm.org/10.1145/1935826.1935845*

Blei, D. M., Ng, A. Y. & Jordan, M. I. (2003), 'Latent dirichlet allocation', *Journal of Machine Learning Research* **3**, 993–1022.
URL: *http://dl.acm.org/citation.cfm?id=944919.944937*

Chang, Y., Wang, X., Mei, Q. & Liu, Y. (2013), Towards twitter context summarization with user influence models, *in* 'Proceedings of the sixth ACM international conference on Web search and data mining', WSDM '13, ACM, New York, NY, USA, pp. 527–536.
URL: *http://doi.acm.org/10.1145/2433396.2433464*

Csardi, G. & Nepusz, T. (2006), 'The igraph software package for complex network research', *InterJournal* **Complex Systems**, 1695.
URL: *http://igraph.sf.net*

Feinerer, I. & Hornik, K. (2013), *tm: Text Mining Package*. R package version 0.5-8.3.
URL: *http://CRAN.R-project.org/package=tm*

Fellows, I. (2013), *wordcloud: Word Clouds*. R package version 2.4.
URL: *http://CRAN.R-project.org/package=wordcloud*

Gentry, J. (2013), *twitteR: R based Twitter client*. R package version 1.1.6.
URL: *http://CRAN.R-project.org/package=twitteR*

Gentry, J., Long, L., Gentleman, R., Falcon, S., Hahne, F., Sarkar, D. & Hansen, K. D. (2013), *Rgraphviz: Provides plotting capabilities for R graph objects*. R package version 2.4.1.

[9] http://www.facebook.com
[10] http://plus.google.com

Grün, B. & Hornik, K. (2011), 'topicmodels: An R package for fitting topic models', *Journal of Statistical Software* **40**(13), 1–30.
URL: *http://www.jstatsoft.org/v40/i13/*

Java, A., Song, X., Finin, T. & Tseng, B. (2007), Why we twitter: understanding microblogging usage and communities, *in* 'Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis', WebKDD/SNA-KDD '07, ACM, New York, NY, USA, pp. 56–65.
URL: *http://doi.acm.org/10.1145/1348549.1348556*

Kwak, H., Lee, C., Park, H. & Moon, S. (2010), What is Twitter, a social network or a news media?, *in* 'WWW '10: Proceedings of the 19th international conference on World wide web', ACM, New York, NY, USA, pp. 591–600.

Lang, D. T. (2013), *RCurl: General network (HTTP/FTP/...) client interface for R*. R package version 1.95-4.1.
URL: *http://CRAN.R-project.org/package=RCurl*

Lehmann, J., Gonçalves, B., Ramasco, J. J. & Cattuto, C. (2012), Dynamical classes of collective attention in twitter, *in* 'Proceedings of the 21st international conference on World Wide Web', WWW '12, ACM, New York, NY, USA, pp. 251–260.
URL: *http://doi.acm.org/10.1145/2187836.2187871*

Lu, R., Xu, Z., Zhang, Y. & Yang, Q. (2012), Life activity modeling of news event on twitter using energy function, *in* 'Proceedings of the 16th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining - Volume Part II', PAKDD'12, Springer-Verlag, Berlin, Heidelberg, pp. 73–84.
URL: *http://dx.doi.org/10.1007/978-3-642-30220-6*$_7$

Pennacchiotti, M., Silvestri, F., Vahabi, H. & Venturini, R. (2012), Making your interests follow you on twitter, *in* 'Proceedings of the 21st ACM international conference on Information and knowledge management', CIKM '12, ACM, New York, NY, USA, pp. 165–174.
URL: *http://doi.acm.org/10.1145/2396761.2396786*

Poblete, B., Garcia, R., Mendoza, M. & Jaimes, A. (2011), Do all birds tweet the same?: characterizing twitter around the world, *in* 'Proceedings of the 20th ACM international conference on Information and knowledge management', CIKM '11, ACM, New York, NY, USA, pp. 1025–1030.
URL: *http://doi.acm.org/10.1145/2063576.2063724*

R Core Team (2013), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.
URL: *http://www.R-project.org/*

Stringhini, G., Egele, M., Kruegel, C. & Vigna, G. (2012), Poultry markets: on the underground economy of twitter followers, *in* 'Proceedings of the 2012 ACM workshop on Workshop on online social networks', WOSN '12, ACM, New York, NY, USA, pp. 1–6.
URL: *http://doi.acm.org/10.1145/2342549.2342551*

Szomszor, M., Kostkova, P. & Louis, C. S. (2011), Twitter informatics: Tracking and understanding public reaction during the 2009 swine flu pandemic, *in* 'Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Volume 01', WI-IAT '11, IEEE Computer Society, Washington, DC, USA, pp. 320–323.
URL: *http://dx.doi.org/10.1109/WI-IAT.2011.311*

Tao, K., Abel, F., Gao, Q. & Houben, G.-J. (2012), Tums: twitter-based user modeling service, *in* 'Proceedings of the 8th international conference on The Semantic Web', ESWC'11, Springer-Verlag, Berlin, Heidelberg, pp. 269–283.
URL: *http://dx.doi.org/10.1007/978-3-642-25953-1*$_2$2

Twitter Blog (2013), 'Celebrating #twitter7'.
URL: *https://blog.twitter.com/2013/celebrating-twitter7*

Zhao, Y. (2013), 'Using text mining to find out what @RDataMining tweets are about'.
URL: *http://www.rdatamining.com/examples/text-mining*

Zubiaga, A., Spina, D., Fresno, V. & Martínez, R. (2011), Classifying trending topics: a typology of conversation triggers on twitter, *in* 'Proceedings of the 20th ACM international conference on Information and knowledge management', CIKM '11, ACM, New York, NY, USA, pp. 2461–2464.
URL: *http://doi.acm.org/10.1145/2063576.2063992*

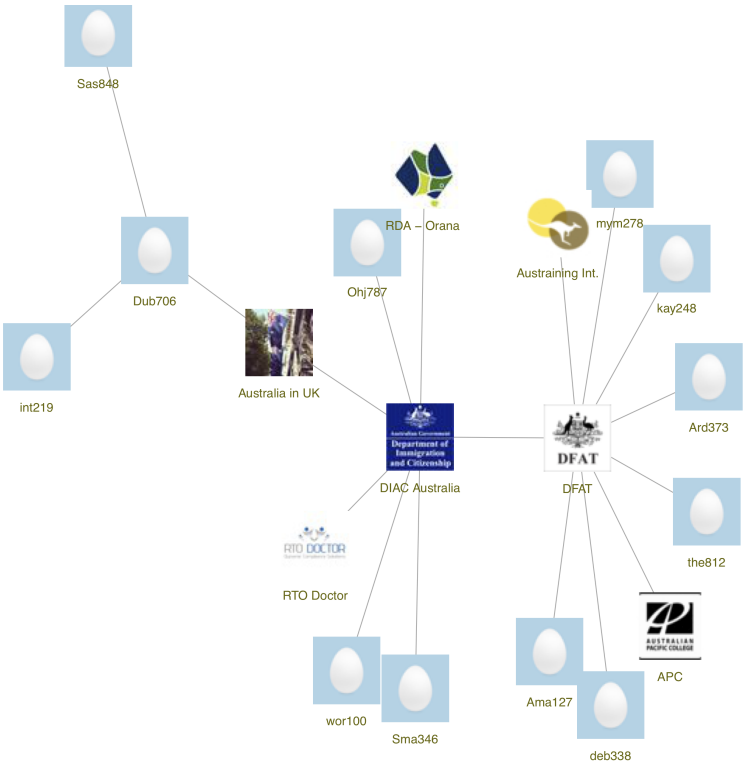


Figure 8: Retweet Graph - I

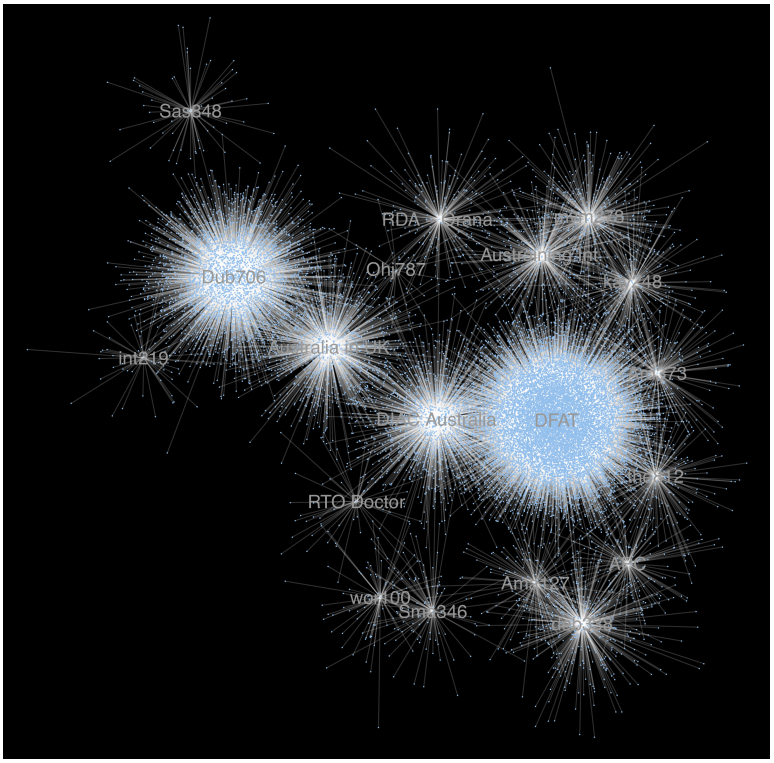


Figure 9: Retweet Graph - II

Cyberbullying Detection based on Text-Stream Classification

Vinita Nahar[1] **Xue Li**[1] **Chaoyi Pang**[2] **Yang Zhang**[3]

[1] School of Information Technology and Electrical Engineering,
The University of Queensland, Australia
Email: v.nahar@uq.edu.au, xueli@itee.uq.edu.au

[2] The Australian E-Health Research Center,
CSIRO, Australia
Email: Chaoyi.Pang@csiro.au

[3] College of Information Engineering,
Northwest A&F University, China
Email: zhangyang@nwsuaf.edu.cn

Abstract

Current studies on cyberbullying detection, under text classification, mainly assume that the streaming text can be fully labelled. However, the exponential growth of unlabelled data in online content makes this assumption impractical. In this paper, we propose a session-based framework for automatic detection of cyberbullying from the huge amount of unlabelled streaming text. Given that the streaming data from Social Networks arrives in large volume at the server system, we incorporate an ensemble of one-class classifiers in the session-based framework. The proposed framework addresses the real world scenario, where only a small set of positive instances are available for initial training. Our main contribution in this paper is to automatically detect cyberbullying in real world situations, where labelled data is not readily available. Our early results show that the proposed approach is reasonably effective for the automatic detection of cyberbullying on Social Networks. The experiments indicate that the ensemble learner outperforms the single window and fixed window approaches, while learning is from positive and unlabelled data.

Keywords: Cyberbullying, Text-Stream Classification, Social Networks.

1 Introduction

Social Networks, such as Twitter and MySpace, are currently the most popular means of information sharing, building relationships, friendships and gossiping. Socialising on the Internet is one of the most common activities among young people. However, there are major disadvantages associated with this interactive form of communication, including the frequent violation of social ethics through various forms of online abuse, plus the risk of being targeted in cybercrime attacks such as cyberbullying and online harassment. Nowadays, cyberbullying is becoming an increasingly obvious concern, especially among children and teenagers. Cyberbullying is defined as "an aggressive, intentional act carried out by a group or individual, using electronic forms of contact, repeatedly and over time against a victim, who cannot easily defend him or herself" (Smith et al. 2008). Research shows growing numbers of children are becoming involved in cyberbullying activities with the intention to harm others (Campbell 2005, Ybarraa & Mitchell 2004) and the consequences of cyberbullying can be more serious than traditional bullying (Campbell 2005). In comparison to verbal bullying, which a targeted victim may forget, the written words remain in print over a longer period of time, which can be read repeatedly by the bully, bystander and victim. Negative consequences of bullying for the victim can include depression, anxiety, low self-esteem (Kaltiala-Heino et al. 2000) and in some cases, attempted suicide (Hindujaa & Patchinb 2010).

Though cyberbullying has received a lot of attention from the Social Science research perspective, it has received much less attention on the research front from the automatic detection standpoint. Current research in this area considers the detection of cyberbullying as a two, or multi class classification problem under supervised learning. In such cases, adequate positive and negative training examples are required for effective classification performance. Although cyberbullying is a growing problem in online communities, the offensive material is not labelled or classified as such, in any form, which makes the investigation of cyberbullying very challenging. In addition, it is too expensive, in terms of time and effort, to manually identify and label bullying messages. However, the exponential growth of online text streams, where contents are either seldom labelled or not labelled at all, makes supervised approaches unfeasible for the automatic detection of cyberbullying instances on Social Networks. In contrast to existing cyberbullying detection methods which require both positive and negative training, we are focussing on building a classifier from a small set of positive training examples, while all other data remains unlabelled and no negative samples are available from training. For this work, we assume that it is possible to extract web history from the server computer to prepare a very small set of positive instances (cyberbullying posts) for initial training, using manual labelling.

In this paper, we propose to integrate a one-class classification scheme in a session-based framework of unlabelled text streams in order to build an automatic classifier in the absence of negative examples for training. To construct this proposed framework, the following challenges have been identified:

- ***Insufficient positive training examples:*** As an alternate to manually labelling of a huge amount of streaming data, only a small number of positively labelled instances are provided for the system.
- ***Insufficient negative training examples:*** Collection of a sufficient number of strong negative examples for training is a labour intensive and time consuming task. In the context of cyberbullying, labelling negative examples is subject to personal opinion because these can be any normal post.
- ***Rare occurrence of cyberbullying instances:*** In relation to the overall amount of data, occurrences of cyberbullying instances are rare, which makes the automatic training of the classifier very difficult.

The objective of a one-class classifier is to discriminate one class of objects (the target class) from all other possible objects, by learning from a training set containing only the objects of the targeted class. The description of the target class should be created in such a way that the likelihood of inclusion of any other objects can be minimised. We incorporate a list of swear-keyword[1] as an indicator for the possible presence of cyberbullying in a one-class classifier. We process the streaming text arriving from the Social Networks with a high speed at the system and in sessions. A session is defined as a chunk of memory block that is used to process incoming streaming text. Each session will be processed using the one-class classification scheme to predict whether a post in the current session is bullying or not. The newly predicted instances of cyberbullying from these sessions will be added to the initial training set examples. Our impetus to employ the one-class scheme is that in the absence of negative samples, the proposed approach will extract reliable negative and a greater number of strong positive instances from each successive session via an ensemble of one-class classifiers, by learning from a very small number of available positive instances.

In this work, we compare the proposed ensemble-based algorithm with baseline single window and fixed window memory management techniques. The experimental results indicate the effectiveness of the proposed session-based one-class ensemble method when combined with the swear-keywords list for cyberbullying detection in terms of test results. Our contributions can be summarized as follows:

- We proposed a novel cyberbullying detection approach under a text streaming scenario and using a one-class classification technique to detect cyberbullying instances, both effectively and automatically.
- We proposed a session-based framework to automatically detect rare cyberbullying instances by an ensemble of one-class classifiers, which extracts reliable negative and strong positive examples of cyberbullying for training. In addition to the one-class classifier scheme, a swear-keyword list is used during feature space design.
- The results of the experiments indicate that the proposed session-based one-class classification scheme tackles rare cyberbullying instances effectively, with the help of only a small number of positive training items.

[1] http://www.noswearing.com/

The rest of this paper is organised as follows. In section 2, a critical literature review is presented, while Section 3 explains the proposed methodology. Section 4 explains how the experiments were performed. The results are also discussed in this section. The conclusions and future work are presented in Section 5.

2 Related Work

2.1 Cyberbullying detection

In a recent study, Dadvar et al. used gender-specific features for cyberbullying detection from a MySpace dataset (Dadvar et al. 2012). They trained a SVM classifier on separate female and male authored posts. The dataset consisted of 34% female and 66% male users. Their gender ratio results indicated an improved performance over N-grams, TF-IDF, and foul words frequency, as feature sets. In the follow-up work they used content-based features, Cyberbullying features and user-based features under supervised learning, to detect cyberbullying (Dadvar et al. 2013). Dadvar et al. have shown improved performance using these features in the detection of cyberbullying. Yin et al. used content information only from online forums for the detection of harassment posts (Yin et al. 2009). Yin et al also proposed contextual features based on the similarity measure between posts. They hypothesised that the posts which are dramatically different from their neighbours are more likely to be harassment posts. Dinakar et al. deconstructed cyberbullying detection into sensitive-topic detection, which is likely to result in bullying discussions, including sexuality, race, intelligence, and profanity (Dinakar et al. 2011). They demonstrated improved performance of binary label-specific classifier over multi-class classifier.

However, these studies involved the text message format and did not consider additional user profile information. The work incorporated topic models such as PLSA (Hofmann 1999) and LDA (Blei et al. 2003), features that are generated under pre-defined topics, and then only features under bullying-like topics are selected (Nahar et al. 2012, 2013). Further, in order to identify predators and victims, users involved in cyberbullying messages are ranked as the most influential predators and the most offended victims through a graph model.

These cyberbullying detection techniques are modelled on a limited set of labelled data under the umbrella of supervised learning, while largely ignoring the fact that the Social Networks data is not labelled as such. In this paper, we propose semi-supervised learning by constructing a one-class ensemble learner.

2.2 Semi-supervised learning for text classification

Some approaches focussed on labelling keywords instead of labelling training samples (Ko & Seo 2009, Liu et al. 2004, Mccallum & Nigam 1999, Qiu et al. 2009, Yang et al. 2011). Initially McCallum & Nigam (1999) manually provided sufficient keywords for each class, which were then used to extract training samples (Mccallum & Nigam 1999). Then the NB-EM algorithm was employed to build final classifiers. Liu et al. (2004) extended this idea by forming clusters of unlabelled documents in order to label descriptive keywords for each category (Liu et al. 2004). Though it alleviates the user input for keywords at the initial level, it needs more representative words for effective

description of the cluster and in addition, the method used a static keyword list, which does not consider further expansion of keywords. For text classification, utilisation of the title term of each category and unlabelled documents was proposed (Ko & Seo 2009, Qiu et al. 2009, Yang et al. 2011). Text labelling was done through the bootstrapping algorithm and used the feature projection method to create an efficient classifier (Ko & Seo 2009). Qiu et al. (2009) used WordNet and expanded keywords to retrieve relevant samples. Then they applied the DocMine algorithm (Barbar et al. 2003) to extract the positive keywords, followed by the application of the PNLH algorithm (Fung et al. 2006) to extract additional positive examples (Qiu et al. 2009). The final classifier was built by the up-to-date PU learning algorithm. In contrast to using the single keyword-based method (Qiu et al. 2009), Yang et al. proposed using three keywords to solve the problem of polysemy (Yang et al. 2011).

However, in cyberbullying detection, only keywords-based (swear-keywords) training is not sufficient, because swear words can be used in any context not always relating to particular person e.g. the fu**ing car. In addition to swear-keywords, there are other features such as personal pronouns ('you', 'yours', etc.) that can be used to correctly detect bullying post as the appearance of a personal pronoun closer to a swear word in a post is more likely to indicate that it could be a bullying post, e.g. You are a big shit, go to hell'.

In recent years many advanced algorithms have been proposed to build text classifiers from positive and unlabelled data (Fung et al. 2006, Yu et al. 2004). Fung et al. (Fung et al. 2006) extracted reliable negative examples from unlabelled samples with their proposed function to automatically calculate the feature strength $H(f_j)$ of feature f_j in the absence of prior knowledge, by normalizing P and U documents that contains f_j as follows:

$$H(f_j) = \frac{n_P(f_j) - min_P}{max_P - min_P} - \frac{n_U(f_j) - min_U}{max_U - min_U} \quad (1)$$

In formula 1, $\max_P$ and $\min_P$ are the maximum values and minimum values of $n_P(f_j)$ for $f_j \in P$, similar formula belongs to $\max_U$ and $\min_U$. How a reliable P is selected, can be defined as:

$$\theta = \frac{1}{N} \sum_{f_j \in P} H(f_j) \quad (2)$$

Though this approach is robust when P_n is very small, its feature extraction process wrongly classifies many negatives as positives and some positives as negatives for cyberbullying detection. The reasons can be summarised as: i. PNLH method is applicable in domains, where user interest is less dynamic as compared to the cyberbullying domain ii. Experiments are conducted on the 20NewsGroup [2] data and this data is in a standard set, which is mostly complete, comprehensive and less noisy.

However given that the data is steaming from Social Networking sites like Twitter, MySpace etc., these approaches are not directly applicable. Characteristics of Social Networks data are: very noisy, incomplete sentences, short sentences, spelling mistakes, own choice of selection of words to convey a message and multiple topics within a one discussion thread. Therefore, given the context of cyberbullying detection from Social Networks, we propose to extend this model by incorporating swear-keywords in a one-class classification ensemble approach.

[2] http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.data.html

3 Cyberbullying detection through Session-based framework on streaming text

$$\begin{aligned} &S_{1,1}, S_{1,2}, ..., S_{1,m_1}; \\ &\qquad S_{2,1}, S_{2,2}, ..., S_{2,m_2}; \\ &\qquad\quad ...; \\ &\qquad\quad S_{n,1}, S_{n,2}, ..., S_{n,m_n} \end{aligned} \quad (3)$$

Given that text streams of variable length arrive on server system in sessions at a high speed as represented by equation 3, we propose a session-based framework, which incorporates one-class classification. Figure 1, shows the general framework in the session-based setting, S_i represents the i^{th} session. Each session contains posts and label information, which can be represented as $S_i = <B_{ij}, y_{ij}>$. $B_{ij} \in \{b_{i1}, b_{i2}, ..., b_{in}\}$ is a set of available posts in i^{th} session and $y_{ij} = \{+1, U_n\}$ are the associated labels to each post. $y_{ij} = +1$ represents a positive post while $y_{ij} = U_n$ indicates an unlabelled message. The one-class classifier will discriminate target class, which are cyberbullying samples from all other unlabelled data of the current session, by learning from a training set containing only the target class examples.

Algorithm 1 : Session-based ensemble classifier for cyberbullying instances classification via one-class classification

Input:
- P_n: Small set of positive instances available for initial training;
- U_n: Set of unlabelled examples of the incoming session;
- E_{n-1}: Classifier's ensemble of the previous session;
- K_p: Keywords list;

Output:
- E_n: Classifier's ensemble of the current session;

1. Data pre-processing and TF-IDF (Session);
2. $E_{n-1} \leftarrow P_n$; initialize E_{n-1} with P_n;
3. //Extraction step:
   T_n, Compute reliable negative/positive instances from U_n via one-class classifier using P_n;
4. //Enlargement step:
   $T_n \leftarrow T_n$ U P_n;
5. Train E (classifier's ensemble) with T_n;
6. $E_n \leftarrow E_{n-1}$ U E;
7. $U_n' \leftarrow U_n$ - T_n;
8. Classify U_n' by E_n;
9. $E_n \leftarrow$ update E_n by re-ranking classifiers of E_n;
10. **return** E_n;

Algorithm 1, shows the overall framework for the proposed session-based system. Step 1 is data pre-processing, where we use the well-known Bag-of-Word (BoW) model to extract features, and we chose the $TF-IDF$ weighting scheme to represent data as an input to the algorithm. $TF-IDF$ gives appropriate importance to terms at the local and global level.

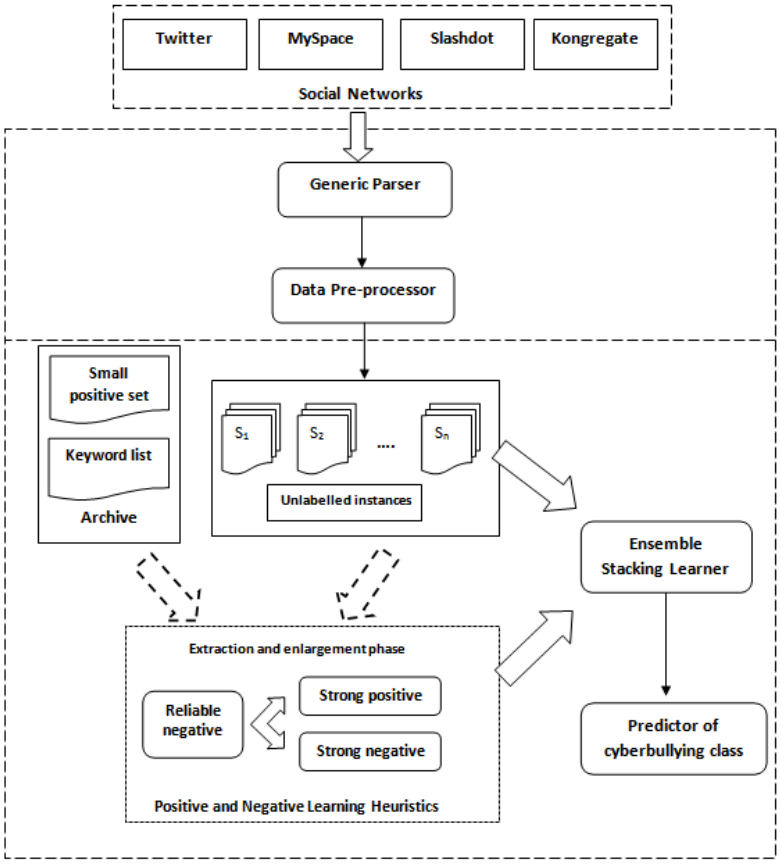


Figure 1: Session-based system design for cyberbullying detection from streaming text

This means data to be represented is based on the frequency of terms in positive and unlabelled data as:

$$TFIDF_{ij} = TF_{ij}.IDF_i \tag{4}$$

TF_{ij} is the importance of word i in post j:

$$TF_{ij} = \frac{n_{ij}}{\sum_l n_{lj}} \tag{5}$$

where, n_{ij} is the count of the term i in post j and denominator is the count of total terms in post j.

$$IDF_i = log\frac{B}{|b_j : f_i \in b_j|} \tag{6}$$

where,$|B|$ is the total number of posts arriving on the system and the denominator is the number of posts containing the term f_i. Thus, terms with good discrimination power between bullying and non-bullying posts will receive higher IDF scores. We assume that these terms will also include bullying-like words, which appear both in bullying and normal posts, while the common terms that appear in many posts, and which therefore do not contain useful discrimination information, receive a lower IDF score. In this work, we have updated the $TF - IDF$ value with each arriving session.

At Step 2, the ensemble is initialized with the very small set of positive bullying instances (Pn). Initial P_n can be achieved by extracting the web history of the survey computer and manual labelling. We simply used 155 manually labelled positive bullying messages to improve the effectiveness of the classifier. In step 3, we extracted reliable negative instances from the current session containing unlabelled streaming data, by incorporating the foul-keyword list and one-class classification scheme (Fung et al. 2006). This method subsequently extracts more reliable negative and strong positive instances from remaining unlabelled data of the current session, while step 4 is an enlargement step where extracted reliable training instances are added to the previous training T_n. Step 5 is a training stage, which is explained in algorithm 2 and step 6 is adding newly trained classifiers to the current ensemble. Step 7 represents remaining unlabelled data of the current session.

Step 8, the newly built model (E_n, ensemble of classifiers), is used to compute the class label of U'_n. Here we chose to use ensemble-base classifiers under concept drift (Street & Kim 2001, Wang et al. 2003, Zhu et al. 2004, Zhang et al. 2008). Concept drift is used to capture positive features in the current session. Our intuition to use concept drift for learning the importance of base classifier is that it will help to capture the changing or emerging bullying-like words. Zhang et al. (2008) summarised the following formula to predict the class label as:

$$E_n(s) = \sum_{i=1}^{|E_n|} \alpha_i c_i(s) \tag{7}$$

where s is a given unlabelled instance, c_i is the i_{th} base classifier in ensemble E_n and $\alpha_i \in R$, (1<i<E_n), is the significance of c_i (i^{th} base classifier).

In formula 7, according to Street and Kim (2001), $\alpha_i \in \{0, +1\}$ for majority voting (Street & Kim 2001) whereas α_i represents the accuracy of c_i, for accuracy

weighted voting (Wang et al. 2003). Zhu et al. (2004) defined $\sum_{i=1}^{|E_n|} \alpha_i = 1, \alpha_i \in \{0, +1\}$ for the dynamic classifier selection (Zhu et al. 2004). In this work, we chose to learn the importance of a base classifier for classification by the ensemble stacking learner proposed by Zhang et al. (2008) under concept drift (Zhang et al. 2008).

Finally (step 9), unlike concept drift, where weak classifiers are generally deleted, we re-rank the classifiers, as even the weak classifiers contain some information which can be helpful for detecting rare bullying instances. The weak classifiers are added at the end of ensemble E_n.

Algorithm 2 : Base classifier training

Input:
T_n: Training for current session;
E_{n-1}: Ensemble of the classifier of the previous session;
K_p: Keywords list;
Output:
E_n: Ensemble classifiers from the current session;
1. $T \leftarrow T_n$ U K_p;
2. Train C by T;
3. $T \leftarrow$ LatestMSession(T_n);
4. Train C' by T;
5. $E_n \leftarrow$ {C, C'};
6. **return** E_n;

Algorithm 3 : Base classifier selection in Ensemble

Input:
T_n: Training for current session;
E_{n-1}: Ensemble of the classifier current session;
$ENSEMBLE\_SIZE$, parameter;
Output:
E_n: Ensemble of classifiers;
1. //Selection M latest training set via one-class ensemble stacker model;
2. $T_s = LatestMSessions(T_n)$;
3. //Sorting in decreasing order
4. if ($S \neq \phi$) then
5. $S = Sort(c_i) by PPR(c_i)$
6. $E_n = E_n + S$ ($ENSEMBLE\_SIZE$);
7. **return** E_n;

Algorithm 2 explains base classifier training on the current session. The base classifiers of the current session will be trained in the absence of labelled data in streaming text. In this work, we use swear-keyword list K_p as a unique feature set. The presences of swear words such as 'f**k', 'bullshit', 'stupid' etc., in the contents increases the likelihood of a potential cyberbullying post. All the swear-keywords in the message are grouped together and are represented in a normalised form as a unique feature set. In the framework, we will train two base classifiers, C, C':

(i) C, trained on T_n and K_p, where T_n is the training extracted from the previous session and K_p is a swear-keyword list. Swear-keywords are used to improve the performance of the one-class classifier;

(ii) C', trained on the swear-keyword list, K_p only.

At step 1 and 2, the first base classifier, C will be trained with T_n U K_p. At step 3 and 4, the second base classifier will be trained, with the Latest M training set. Here we have to stress that the Latest M training set is based on the re-ranked base classifiers in E_n. At step 5, the final base classifier will be selected by ensemble stacking learner, which is explained in Algorithm 3.

Base classifier selection in an ensemble is important because the size of the ensemble increases with the addition of new base classifiers from the arriving sessions. The proposed algorithm for base classifier selection, in an ensemble of the classifiers, E_n consists of base classifier selection up to $ENSEMBLE\_SIZE$, and sorting of the base classifiers within the ensemble is performed based on re-ranking, where weak classifiers ranked lower in the ensemble E_n.

In algorithm 3, $ENSEMBLE\_SIZE$ is a user-defined parameter, which defines ensemble size to keep the list of classifiers below a certain threshold. Then, because of one-class classification, we sort the base classifiers based on the accuracy of positive instances. Step 2 shows the selection of M latest training sets. For our framework stacking training set (T_s) can be define as:

$$T_s = LatestMSessions(T) \tag{8}$$

where we select latest M training sets from T_n.

Step 4 shows sorting of c_i in S (base classifier of S) in decreasing order. In step 4, E_n is accumulated with base classifiers below threshold. To compute the Positive Prediction Rate (PPR) by c_i, we simply used the positive prediction function defined for one-class classification (Zhang et al. 2008) as below:

$$PPR(c_i) = \frac{\sum_{j=1}^{n} PositivePrediction(c_i, j)}{\sum_{j=1}^{n} |d_j|} \tag{9}$$

where $PositivePrediction(c_i, j)$ represent the number of correctly predicted positive instances by c_i in s_n.

4 Experiments and Results

Dataset: For the experiments, we used data provided by Fundacion Barcelona Media[3] for the workshop on Content Analysis for the Web 2.0. The given data were collected from the four different Social Networking sites; Twitter, MySpace, Kongregate, and Slashdot. The size of the total dataset is 1.57 million data instances. Characteristics of data from these four sites varies, which made our work even more challenging. For instance, post length of a tweet is very short in comparison to a MySpace post. MySpace posts are in the form of a discussion thread, where in some cases discussion deviates from the topic. On the other hand, Slashdot data is the user comments on news posts. Kongregate are real time chat logs of online gaming. Our task is to extract bullying instances from streaming text of any type.

The raw data is available in XML file format of different structures. A data parser is developed to extract the content and time information. Because of the nature of Social Network data, extensive pre-processing is conducted to improve the quality of data and the performance of the subsequent steps. In the pre-processing module, we ignored the most frequent and the least frequent words, which do not carry significant information. In addition, we removed,

[3] http://caw2.barcelonamedia.org/

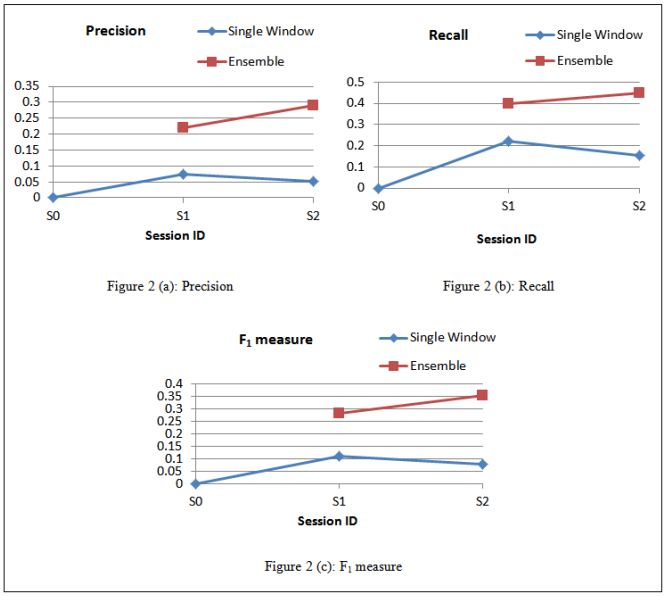


Figure 2 (a): Precision

Figure 2 (b): Recall

Figure 2 (c): F_1 measure

Figure 2: Experimental result on 0.005% positive training (10000 Unlabelled instances)

hash tag and re-tweet swear-keywords from Twitter datasets to avoid bias, and we also removed web addresses. All the message contents were converted into lower case letters. The stop words were removed and all words were converted into a seed word using WV-Tool. We employed the session scenario by sorting messages based on the time and generated streaming sessions of varying length. We used the TF-IDF scheme to represent data as an input to the learning algorithm, by providing appropriate weight to each unigram feature. The existing $TF - IDF$ values updated by the system on the arrival of a new session.

Evaluation matrix: The classification of cyberbullying messages is a very critical issue because of false positive and false negative cases. On one hand, to identify non-bullying instances as bullying itself is a sensitive issue (false positive) and, on the other hand, the system should not bypass the bullying post as a normal post (false negative). Therefore, false positive and false negative instances are both critical. Thus, precision, recall and the $F_1 measure$ were considered for the performance evaluation metric as defined below:

Precision: The total number of correctly identified true bullying posts out of retrieved bullying posts.

Recall: Number of correctly identified bullying cases from the total number of true bullying cases.

F_1 measure: The equally weighted harmonic mean of precision and recall.

Our system works on the streaming sessions, therefore we present the classifier performance on the average value. For n sessions, average values of precision, recall and the $F_1 measure$ can be defined by the formulas (7), (8) and (9):

$$Precision_{avg} = \frac{\sum_{i=1}^{n} Precision_n}{n} \quad (10)$$

$$Recall_{avg} = \frac{\sum_{i=1}^{n} Recall_n}{n} \quad (11)$$

$$F_{1_{avg}} = \frac{\sum_{i=1}^{n} F_{1_n}}{n} \quad (12)$$

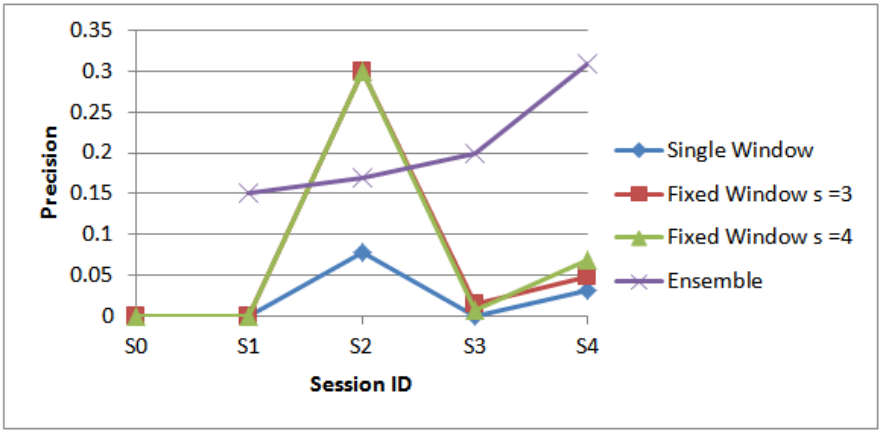
We compare the performance of the proposed ensemble approach with the single and fixed window methods. In the single window, the classifier is built on the current session only, while in the fixed window, the classifier is built on the instances of fixed window sizes s = 3 and s = 4.

Scenario 1: Experimental results with the one-class ensemble stacking classifier

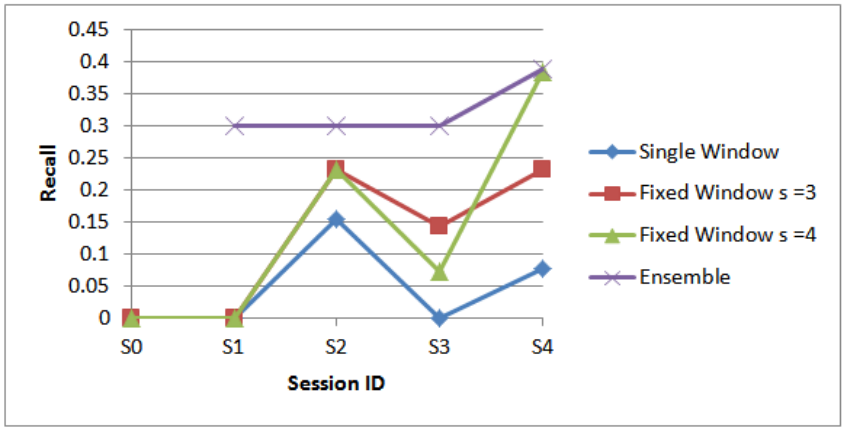
In scenario 1, we conducted an experiment, when only 0.005% of positive instances of cyberbullying were available for initial training, while incoming sessions were unlabelled. In this scenario, to ensure that there were enough bullying posts available in the streaming sessions, only two sessions were generated, each of 5,000 instances. We considered only two sessions in this scenario as we observed that if there are many more sessions i.e. if s = 5, then some of the sessions could contain no bullying posts and therefore negative instances would accumulate in the ensemble. As a consequence, in this scenario, the experiment was conducted only on the single window and on the ensemble classifier. We fed the system with 0.005% positive instances for initial training; one-class classifier extracted more reliable negative and positive samples for the prediction of unseen instances. However, overall positive instances in the total data were still too low and therefore single window was not able to perform well. Figure 1, shows the comparison of the single window and ensemble, where we can see that the proposed ensemble learner outperformed the single window.

Scenario 2: Experimental results with the one-class ensemble stacking classifier, filtered by swear-keywords

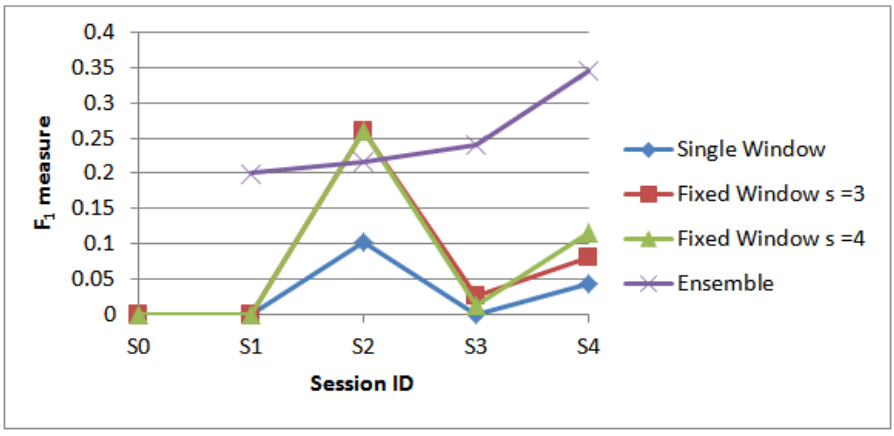
In scenario 2, we first filtered messages, based on the swear-keywords, as swear-keywords indicate presence of the abusive or bad posts. It is also important to note that the presence of swear-keywords, alone, is not necessarily a clear indicator that the post is a case of cyberbullying. In this scenario, the ensemble classifier performed better than all other methods, after extracting and enlarging a good training set. The performance of the fixed window classifier, with window sizes s = 3, and s =



(a) Precision



(b) Recall



(c) F_1 measure

Figure 3: Experimental result on 0.0025 % initial positive training, 20000 unlabelled instances (including 10000 containing keywords)

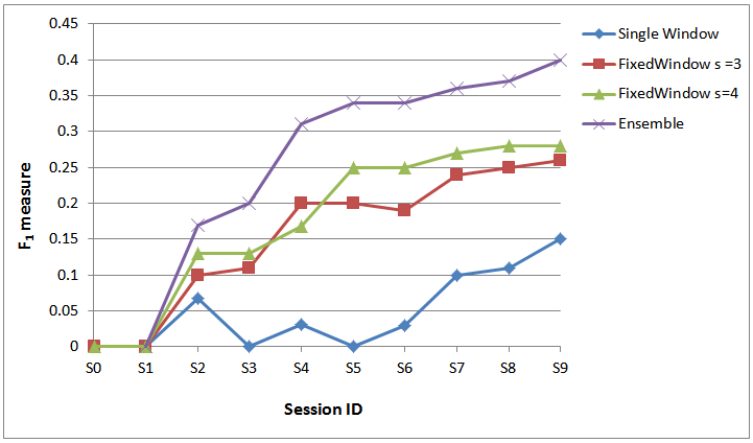


Figure 4: Experimental result on 0.0025 % initial positive training on 10 streaming sessions (each session containing 1,57,000 unlabelled instances)

4, was nearly the same, and with the third session ($S2$), both these fixed window classifier instances performed better than the ensemble one. After that the effectiveness of the fixed window method was declined, whereas the performance of the ensemble classifier was increased with the arrival of new sessions. Basically, single window classification is not able to function properly in the absence of the enough training sets. From figure 3, we can note that as the new sessions arrive at the system, successful performance of the fixed window classifier, with window size s = 3, decreases, while performance of the ensemble classifier outperforms the three other methods. This is because with window size s = 3, the training capacity is not enough to classify new instances. Only at the last session, recall of the fixed window classifier, with size s = 4, is similar to the ensemble classifier, when it is able to collect enough training sets. However, ensemble classifier continues to perform well in all sessions because of the extraction of good training sets.

Scenario 3: Experimental result with the one-class ensemble stacking classifier, on 10 sessions (no filter)

As shown in figure 4, scenario 3 constitutes only 0.0025% initial positive training on 10 streaming sessions, where each session contains 157,000 unlabelled instances. In total 1.57 million instances were processed. In the streaming of 10 sessions, only precision was compared [precision is defined as: $TP/(TP + FP)$]. For instance, we were interested in true positive and false positive instances identified by the system. Here we observed that in the text streams, bullying posts were very rare as most of the sessions did not contain bullying posts, although from our manual observation we found that some messages contain swear-keywords. For example a message may only contain single words, like 'hell', 'shit'. Most of the messages containing single swear-keywords were bypassed by our system as these were non-bullying messages. In the validated dataset, posts containing common words like 'shit' and 'hell', were not labelled as bullying. In scenario 3, the ensemble learner outperformed all other methods, which shows the strong ability for ensemble learner to identify cyberbullying traces from the huge amounts of unlabelled data. With the accumulation of positive and/or negative training samples, the improvement in precision in fixed window size s = 3, s= 4 and ensemble learner was observed throughout the sessions. However, single window did not performed well in this scenario. In this scenario, recall was not compared because of the high level of negative data. In future work, we will extend our system by incorporating a user feedback module, so as to ensure that false negative cases are correctly identified.

5 Conclusion and Future Work

In this paper on cyberbullying detection, we proposed a session-based one-class ensemble classifier on the streaming text to provide an alternative for unlabelled data classification. Previous works on cyberbullying detection were conducted on limited labelled datasets, while training was carried out using positive and negative instances. We investigated the one-class ensemble learning method for cyberbullying detection to tackle the real world situation, where streaming data is not labelled at all. We devised a way to train our system using a small set of positive trainings only, where the system automatically extracts reliable negative and more strong positive samples for training from the huge amount of unlabelled data. In addition, through our investigations, we found that cyberbullying detection is a complex phenomenon because of the difference between cyberbullying, cyberteasing and cyber-jokes. This differentiation leads to false positive and false negative cases, hence diminishes performance of the classifier, as cyberbullying is a subject to personal feeling and interpretation. Nevertheless, our early results in this paper indicate that the proposed approach is a feasible means to learn cyberbullying instances, effectively and automatically, from unlabelled text streams from SNs.

In the future, we will extend our work by incorporating a sophisticated user feedback system to filter out cyberbullying-like instances such as cyberteasing or cyber-jokes cases from the real cyberbullying. This will improve the learning (discriminating) capability of the classifiers. More advanced bullying-specific features including users cyberbullying patterns will be incorporated along with the baseline swear-keywords method in the session-based one-class ensemble learner scheme. It will also be interesting to look at the user groups, or circles, to which users belong. For instance, Facebook enables its users to create and add friends to various groups such as close-friends, family, acquaintances etc, while Google allows

people to create and add to circles. These groups indicate the relationship between the users to some extent, which will be helpful to detect cyberbullying more accurately, as cyberbullying, also known as peer victimization, where cyberbullies include friends, siblings, fellow students and people unknown to the victim.

References

Barbar, D., Domeniconi, C. & Kang, N. (2003), Mining relevant text from unlabelled documents, *in* 'Proceedings of the Third IEEE International Conference on Data Mining(ICDM)'.

Blei, D. M., Ng, A. Y. & Jordan, M. I. (2003), 'Latent dirichlet allocation', *Journal of Machine Learning Research* **3**, 993–1022.

Campbell, M. A. (2005), 'Cyber bullying: An old problem in a new guise?', *Australian Journal of Guidance and Counselling* **15**, 68–76.

Dadvar, M., de Jong, F., Ordelman, R. & Trieschnigg, D. (2012), Improved cyberbullying detection using gender information, *in* 'Proceedings of the Twelfth Dutch-Belgian Information Retrieval Workshop', pp. 23–25.

Dadvar, M., Trieschnigg, D., Ordelman, R. & de Jong, F. (2013), Improving cyberbullying detection with user context, *in* 'European Conference on Information Retrieval', pp. 693–696.

Dinakar, K., Reichart, R. & Lieberman, H. (2011), Modeling the detection of textual cyberbullying, *in* 'International Conference on Weblog and Social Media - Social Mobile Web Workshop'.

Fung, G. P. C., Yu, J. X., Lu, H. & Yu, P. S. (2006), 'Text classification without negative examples revisit', *IEEE Transactions on Knowledge and Data Engineering* **18**(1), 6–20.

Hindujaa, S. & Patchinb, J. W. (2010), 'Bullying, cyberbullying, and suicide', *Archives of Suicide Research* **14**, 206–221.

Hofmann, T. (1999), Probabilistic latent semantic analysis, *in* 'In Proc. of Uncertainty in Artificial Intelligence, UAI99', pp. 289–296.

Kaltiala-Heino, R., Rimpel, M., Rantanen, P. & Rimpel, A. (2000), 'Bullying at school-an indicator of adolescents at risk for mental disorders', *Journal of Adolescence* **23**, 661–674.

Ko, Y. & Seo, J. (2009), 'Text classification from unlabeled documents with bootstrapping and feature projection techniques', *Information Processing and Management* **45**(1), 70–83.

Liu, B., Li, X., Lee, W. S. & Yu, P. S. (2004), Text classification by labeling words, *in* 'American Association for Artificial Intelligence (AAAI)', pp. 425–430.

Mccallum, A. & Nigam, K. (1999), Text classification by bootstrapping with keywords, em and shrinkage, *in* 'Proceedings of the Workshop in Unsupervised Learning in Natural Language Processing at ACL', pp. 52–58.

Nahar, V., Li, X. & Pang, C. (2013), 'An effective approach for cyberbullying detection', *Journal of Communications in Information Science and Management Engineering (CISME)* **3**, 238–247.

Nahar, V., Unankard, S., Li, X. & Pang, C. (2012), Sentiment analysis for effective detection of cyber bullying, *in* 'The 14th Asia-Pacific Web Conference (APWeb)', pp. 767–774.

Qiu, Q., Zhang, Y. & Zhu, J. (2009), Building a text classifier by a keyword and unlabeled documents, *in* 'Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD)', pp. 564–571.

Smith, P. K., Mahdavi, J., Carvalho, M., Fisher, S., Russell, S. & Tippett, N. (2008), 'Cyberbullying: its nature and impact in secondary school pupils', *Journal of Child Psychology and Psychiatry* **49**, 376–385.

Street, W. N. & Kim, Y. (2001), A streaming ensemble algorithm (sea) for large-scale classification, *in* 'Proceedings of the seventh international conference on Knowledge discovery and data mining (KDD)', pp. 377–382.

Wang, H., Fan, W., Yu, P. S. & Han, J. (2003), Mining concept-drifting data streams using ensemble classifiers, *in* 'Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)', pp. 226–235.

Yang, B., Zhang, Y. & Li, X. (2011), 'Classifying text streams by keywords using classifier ensemble', *Data and Knowledge Engineering* **70**(9), 775–793.

Ybarraa, M. L. & Mitchell, K. J. (2004), 'Youth engaging in online harassment: associations with caregiver-child relationships, internet use, and personal characteristics', *Journal of Adolescence* **27**, 319–336.

Yin, D., Xue, Z., Hong, L., Davisoni, B. D., Kontostathis, A. & Edwards, L. (2009), Detection of harassment on web 2.0, *in* 'Content Analysis in the WEB 2.0 (CAW2.0) Workshop at WWW2009'.

Yu, H., Han, J. & chuan Chang, K. C. (2004), 'Pebl: Web page classification without negative examples', *IEEE Transactions on Knowledge and Data Engineering* **16**, 70–81.

Zhang, Y., Li, X. & Orlowska, M. (2008), 'One-class classification of text streams with concept drift', *IEEE International Conference on Data Mining Workshops(ICDMW)* pp. 116–125.

Zhu, X., Wu, X. & Yang, Y. (2004), Dynamic classifier selection for effective mining from noisy data streams, *in* 'Proceedings of 4th IEEE international conference on Data Mining (ICDM)', pp. 305–312.

Using Social Media Data for Comparing Brand Awareness, Levels of Consumer Engagement, Public Opinion and Sentiment for Big Four Australian Banks.

Inna Kolyshkina[1], **Boris Levin** [2] **and Grant Goldsworthy**[3]

[1] School of Information Technology and Mathematical Sciences
Division of Information Technology Engineering & Environment
University of South Australia
GPO Box 2471
Adelaide SA 5001
Email: Inna.Kolyshkina@unisa.edu.au

[2] TBIS Holdings Pty Ltd
PO Box 899
North Sydney NSW 2059

[3]All Financial Services (NSW) Pty Ltd
PO Box H161
Australia Square, NSW 1215

Abstract

The growing availability and popularity of opinion-rich resources on the web led to an eruption of activity in the area of analysis of data coming from these resources. Opportunities exist to understand the extent of public engagement and sentiment toward a brand, a product or an event. In this paper, we present a case study for opinion extraction applied to the banking domain that illustrates how social media can be used to gain insight into the public opinion, sentiment and spread of social conversation related to this domain including changes that are triggered by a domain-relevant event. We applied advanced machine learning and data science techniques to the relevant social media and news data from the web to analyse the nature of public opinion in Australia toward the four major Australian banks in the context of the banks reaction to the Reserve Bank of Australia lowering the official interest rate. The resulting insights into public sentiment, reach, the topics discussed by the public and how these compared between the banks can be used proactively to inform organisational decision making.

Keywords: social media data, machine learning, random forests, generalised boosted models, text mining, R, sentiment analysis, topic modelling.

1 Introduction

Opinion mining of social media data triggered by the Web 2.0 success has been quickly developing as an important area of interest for both business and research. This paper presents a case study for public opinion extraction applied to the banking domain. The context of the study is as follows. In October 2012 the Reserve Bank of Australia lowered the official interest rate. The four major Australian banks took some time before acting then passed on rate cuts for borrowers of less than the full cut by the Reserve Bank.

The aim of the study was to discover public reactions and gain insights into the following questions:

- The volume and extent of reactions. Did people talk more about the banks as a result of the rate cut? How many people talked? How many people listened (i.e. were reached by the messages)?
- The nature of public sentiment. Did rate cuts affect consumer and media sentiment toward banks? How did consumer and media sentiment compare? Were media more or less critical of the banks than the public?
- Differentiation between the banks. How did the banks compare in terms of the number of people interested them and what people said about them?
- Public opinion groupings. What were the main topics discussed and opinions expressed about the banks? What population groups expressed them?
- Bank public relations initiatives. What levers (campaigns, community initiatives, sponsorship etc.) the banks were using to improve popularity and public sentiment? Did they work?

While the analysis was done using a number of social media sources, this paper concentrates on a methodology for data analysis using Twitter data.

This project illustrates how social media data can be harvested to gain insight that can be used to proactively manage organisational public image, brand awareness and customer satisfaction.

2 Data Extraction and Storage

2.1 Software used for data retrieval and storage

We used Java technology stack for data retrieval. The library twitter4j written in Java establishes the bridge

between the client program and the data available through the Twitter public API. The Java code that is responsible for various parts of data gathering and processing is written in a manner that allows for components reuse and deployment in various standalone and hosted environments, in particular, in a web application or as part of an enterprise solution. For the storage, PostgreSQL 9.0 was chosen.

2.2 Data extraction and storage process

We collected 12 weeks' worth of social media data starting from October 3 mentioning the "big four" Australian banks (WBC, CBA, NAB and ANZ banks) and originated in Australia. The Twitter API allowed us to extract the data geographic location of the poster, user name and textual self-description. The extraction was made in bulk (as opposed to one per call), to account for Twitter's throttling restrictions, in other words we accumulated user IDs available on tweets, then issued a single call to get users data for the list of IDs.

Figure 1 gives an overall view of what happens to the social data within our solution. The steps completed in order to prepare data for analysis are outlined below.

1. Data retrieval using the API published for developers. New data was collected on a daily basis and appended to the existing data set.
2. Data collation and pre-processing. Once the data had been retrieved, we prepared it for storage. All handling of the data in Java used tweets in the form of a tweet bean created from the Twitter data seed. This bean encapsulates, on top of the standard tweet data, some elements obtained as the result of the pre-processing, such as user location and gender, followers count, etc.
3. Validation. We used data analysis in the ways described below to develop specific rules to identify and filter out entries that were considered noise and had to be removed from further processing
4. Storage. The table structure that was used to store tweets was reasonably straightforward closely resembling that of the actual tweet. The Tweeter feed provided a unique tweet ID that could also be used as the database ID. As the tweet beans already contained data enriched at the pre-processing stage, persistence logic was very simple. An object-relational mapping product Hibernate facilitated seamless integration of persistence logic into Java code.

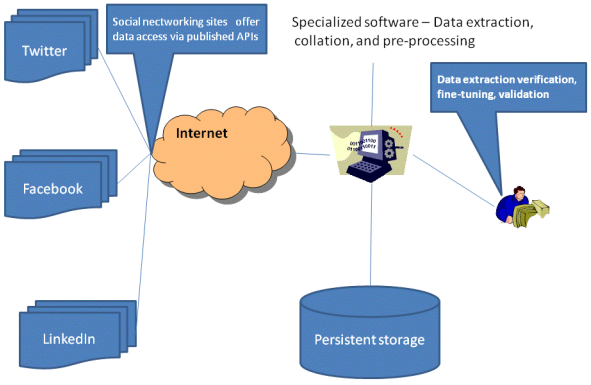


Figure 1: Extracting the social media data.

3 Data Analysis

3.1 Software used for data analysis

Data analysis was done using open-source software R (`www.cran.org`). We used text processing and predictive modelling techniques implemented in R packages including `tm`, `tau`, `lda`, `randomForest`, `gbm`, `earth`, `openNLP`, `wordNet`, `twitteR`, `stringr`, `plyr`, `lda` and `topicmodels`.

3.2 Text data pre-processing: cleaning and enrichment

Prior to the sentiment analysis and topic modelling we did extensive data pre-processing and enrichment of the text field.

As a first step, we filtered out the "noise data" including duplicated tweets; spam and advertising tweets (e.g. tweets sent by the banks themselves and tweets from third party organisations advertising products of a particular bank). We also excluded irrelevant tweets for example those containing abbreviations that are similar to the bank names, for example "`I cba getting up today`" — abbreviation "`cba`" is similar to "`CBA`" which stands for Commonwealth Bank of Australia.

We then performed fairly standard pre-processing of the text field (see for example Grun and Hornik (2011) and Davi et al. (2005): put all letters in a lower case, removed punctuation, stopwords, spaces, performed stemming etc. and transformed the text data field to a document-term matrix that became a basis of the input data for further analysis.

To make sure that any information describing potentially important features of a tweet had not been lost in the text pre-processing, we enriched the data with variables that described such features. For example, we added variables showing the number of characters in the tweet, count of words in the tweet, number of repeated letters (e.g. "`grrrrrrreat`"), number of capital letters (e.g. "`commbank is THE BEST`"), ratio of capital letters to the number of characters in the tweet, ratio of positive to negative words, ratio of positive and negative words to the total word count, ratio of stopwords to the total word count, count of each common emoticon (e.g. smile, frown) in the tweet, number and position in a phrase of exclamation and question marks, count of each punctuation mark in a tweet (e.g. "`I don't know ...`", "`good!!!!!!!!!!`"), number and position in a phrase of emotion-expressing words (e.g. `wow`, `yay`, `lol`, `haha`, `grrrr`), swear words, negation words (e.g. `not`, `don't`, `isn't` etc), stopwords etc. as well as flags for common 3 or more-words collocations being present in the tweet (e.g. "`reserve bank of Australia dropped interest rate`").

Data sparsity was addressed using the approach similar to that described in Phan et al. (2008) and Feinerer et al. (2008). A useful step in data preparation which helped in sparsity reduction was combining and recoding a variety of spelling variations of frequently used words and word combinations e.g. "`thank you`" ="`thankee`" ="`thankyou`" ="`thanks`" ="`thanx`"="`ty`" etc.

Additionally we took extra steps to further minimise the potential effect of sparsity on the validity

and robustness of the findings at the later stages of the data analysis, in the process of modelling. For example in performing sentiment analysis, we applied to the data three modelling methods (generalised boosted regression, random forests and multivariate adaptive regression splines) and checked their outputs for consistency.

3.3 Sentiment scoring

The purpose of our sentiment analysis was to establish whether a tweet carried a positive and negative opinion or emotion toward the bank it mentioned. For example the tweet "`lovely weather, great day, walking past Westpac building`" might express positive emotion but that emotion is not directed toward the bank and so it cannot be classified as a tweet with positive sentiment for the Westpac Banking Corporation.

3.4 Preparation for sentiment analysis

As is typical for sentiment analysis (for example, see Wilson et al. (2005)), prior to the analysis we prepared lexicons, word lists and synonym lists to be used in the analysis. We created our lexicons by modifying and combining well-known lexicons available in public domain, for example, Hu and Liu's opinion lexicon (http://www.cs.uic.edu/ liub/FBS/opinion-lexicon-English.rar (see Liu (2010), Hu and Liu (2004)) and further enriched them to reflect the specifics of the context of our domain as well as the Australian contextual specifics.

The lexicons, word lists and synonym lists we prepared included: list of words with positive or negative polarity generally and within our context (contextual polarity may be different from the word's prior polarity e.g. "`lowered rate`" is positive in the banking context but "`lower`" may be perceived as a negative word in general context), list of words expressing negation (e.g. "`not`", "`no`", "`isn't`" etc); list of stopwords , list of swear words (e.g. "`hell`'); emoticons and words expressing them (e.g. ☺ "`frown`"); list of words expressing emotion e.g. (e.g. `wow`, `yay`, `lol`, `haha`, `grrrr`); synonym lists — general and relative to the domain (e.g. "`Commbank`", "`CBA`" and "`Commonwealth bank`") and list of important words/concepts relative to the domain (e.g. "`interest rate`", "`Reserve Bank`" etc.)

3.5 Building a sentiment scoring model

To achieve maximum accuracy in an efficient and statistically valid way, we decided to approach assigning sentiment to a tweet as a classification problem and created a classification model for sentiment scoring.

We started from creating a rating data set similar to our data in terms of geography, domain and collection time which comprised 500 tweets. The tweets were then manually marked by two annotators as 1 standing for "positive towards the bank" (e.g. "`Westpac is the best`"), -1 standing for "negative towards the bank" (e.g. "`Westpac is the worst`"), or 0 standing for "other" (e.g. "`I am now in Westpac, see u soon`"). To maximise the reliability of the annotation, we marked a tweet as 1 or (-1) if the annotators agreed on the sentiment and 0 if annotators disagreed (the annotators agreed in 91% of cases).

Then, we used the annotated data to build a sentiment scoring model. To achieve this with maximal accuracy, three predictive models were applied to the annotated data: random forests, generalised boosted regression and multivariate adaptive regression splines, the target variable being the annotated sentiment score and the predictors being the elements of the document-term matrix and the added enrichment variables. The models were implemented in R using packages `randomForest`, `gbm` and `earth`. The models were built with cross validation on a 70% randomly selected training subset of the data and their accuracy was then tested on the remaining 30% test subset. The sentiment for the original tweet data was then calculated as the combined scores of the three models.

Apart from the derived score, another important output from the models was finding the words and phrase or sentence features that were the most important in predicting the sentiment. For example, the ratio of count of exclamation marks to the count of characters in the tweet was one of important drivers of a negative sentiment, while the number of smiles in the tweet was one of important drivers of a positive sentiment.

3.6 Establishing main common topics of public interest and the related sentiment

It was important to establish the main topics of interest to the public related to the major Australian banks and how they varied from one bank to another as well as the dynamics of public interest in the topics over time.

To establish the number and nature of the key topics of interest we applied topic modelling, the machine learning and natural language processing technique implemented using the latent Dirichlet allocation (LDA) as described in Blei and Lafferty (2009). LDA approach assumes that K latent topics are associated with a document collection, and that each document exhibits these topics with different proportions and the posterior distribution of the topics given the observed documents determines a hidden topical decomposition of the collection. The optimal number of topics K is established as the number that provides the best model fit, typically measured by a log-likelihood-based criterion (see, for example Grun and Hornik (2011)). In our case the best model fit was achieved when the number of topics was seven.

To establish the meaning of each of these topics, we reviewed the most frequent terms associated with the topic, and the results suggested that the topics were as follows: bank reaction to the interest rate drop, bank community initiatives, bank employee wages, offshoring skills, feedback on the customer service across banks, economic reports published by banks (e.g. Westpac Consumer Sentiment report) and banks' online interfaces.

We then scored messages by whether they were related to each of the seven key topics, and based on that established sentiment per topic and compared the sentiments across the banks.

3.7 Establishing the socioeconomic characteristics of the comment-makers

To understand whether topics, reach and sentiment varied by different population groups, we needed to establish the characteristics of the posters where this was possible.

The main data field, apart from the geographical information on the poster, which was applied for derivation of such information, was the user description field. To derive user's socioeconomic information (e.g. gender, marital status, occupation),

we performed textual analysis of this field including data cleaning and pre-processing similarly to the description provided above; creating relevant dictionaries and synonym lists for this context (for example "journalist" ="journo", "mother" = "mom" = "mum" ="mummy" = "mommy"); collocation analysis to find the most common words and word combinations for this field e.g. "wife and mother", "CEO", "freelance journalist", "part-time". We then used the outputs of this pre-processing to add flags that described the features of the user description field, for example:

- Occupation information, which also may serve as a proxy for age and income, for example,. "journalist", "student", "CEO", "manager", "developer", "part-time", "retired"
- Gender (e.g. "mother", "lady")
- Family status: marital and parental status children ("husband", "single")
- Organisation or private poster

Gender information for private posters was then further refined by comparing the names provided in the user name field to with the list of names by gender provided by US social security website http://www.ssa.gov/OACT/babynames/.

The sparsity in user description field was managed similar to how it was done for the text field analysis described above. To further reduce sparsity, we combined occupation or family status word groups that had less than 35 posters into a broader category. For example "journalist", "writer", "editor" and "columnist" was combined into the category called "writer_journalist''.

In some cases user description data did not provide clear information on the occupation, age or family status of the poster. For example a poster may have had a user name like "doglover123" and user description like "Don't walk behind me, I may not lead. Don't walk in front of me, I may not follow. Just walk beside me and be my friend". Preliminary textual analysis of the tweets posted by such users indicated that they were closer to private users then to organisations or news sources in terms of the use of swear words, emoticons, exclamation marks, dots and capital letters per tweet. While detailed investigation of data on this group was outside of our scope, in future it may provide more insight into its likely socioeconomic characteristics. In our analysis such posters were analysed as a separate group called "private_poster".

3.8 Measuring Public Engagement

Key questions of interest in terms of public engagement were as follows: how many people expressed a specific opinion or sentiment and how many people were exposed to these messages.

The former was measured as the number of messages. To gain insight into the latter we used a measure called reach. In the application of statistics to advertising and media analysis, reach refers to the total number of different people (or sometimes the percentage of the target audience) who had an opportunity to see the ad. The reach of a tweet, for the purposes of this project, was defined as the maximal theoretical number of the people who were "reached" by the tweet i.e. to whom the tweet was sent in the original or retweeted form and who therefore could have read the tweet. Twitter API allows us to obtain the number of the retweets for each tweet and the number of the followers of the poster. Reach was calculated as sum of number of retweets (or slightly reworded tweets) multiplied by the number of followers of the users who retweeted. This was achieved by iterating through the list of the tweets collected since the beginning of observations, finding the number of the retweets for each tweet and the number of the followers for each of the users who post a retweet. This data was accumulated for each tweet.

4 Findings — selected examples and their statistical significance

Findings of the study illustrate the commercial insight that can be extracted from social data in terms of public opinion and sentiment dynamics and spread in reaction to events. While the complete set of the delivered findings is out of scope of this article, we present some selected findings as an example.

4.1 Statistical significance of the reported differences between the banks in terms of the established sentiment, topics and reach

Comparison of the banks in terms of the reach, sentiment and coverage for each of the identified seven key topics of interest involved formal testing of the statistical significance of the differences between the banks with the null hypothesis of no difference existing in each case. These null hypotheses were tested by the analysis of variance approach.

The results indicated the presence of significant differences between the four banks in terms of the expressed sentiment ($p>0.05$) and the reach ($p<0.01$). Some of the topics such as offshoring skills and low bank employee salaries were significantly more expressed in relation to certain banks ($p<0.01$). For other topics, e.g. the topic related to economic reports produced by banks, there was no significant differences across the banks ($p>0.2$).

All the differences mentioned below are statistically significant unless stated otherwise.

4.2 Event-driven dynamics of public opinion. Influence of RBA's rate drop on public interest and public opinion toward banks

The RBA rate drop drew public attention to the big four banks. The total reach of bank-related messages significantly increased by 40% in the weeks following interest rate drop ($p<0.01$).

The big four banks' not matching fully the RBA interest rate drop caused a significant ($p<0.05$) fall in overall sentiment (5%) towards the banks in the two weeks after the event with the sentiment starting to improve in the third week

4.3 Public engagement findings summary. Comparison of banks by number of messages and reach

Bank 1 seemed to be the most popular bank to be discussed on Twitter with 31.0% all banking-related messages mentioning the bank. It was closely followed by Bank 2 (29.0%). However reach was higher for Bank 2 (33.2%) than for Bank 1 (26.3%) which suggests that Bank 2's social media strategy was more effective than Bank 1's. Bank 4 was third in public engagement with 23.8% messages mentioning the

bank; however reach was low (9.0%) suggesting opportunities to improve the bank's social media strategy. Bank 3 seemed to have had the lowest level of social media engagement among the four banks with only 5.6% media messages relating to the bank and 8.6% of bank-related media reach.

4.4 Sentiment analysis findings summary

Sponsorship and community initiatives-related messages improved sentiment toward banks while the most negative sentiment was related to banks' interest rate drop not matching the RBA level, Bank 1 not treating employees fairly, Bank 2 off-shoring skills and Bank 3's poor online interface as well as consumer complaints across the banks.

We summarised public sentiment by posters' gender, occupation, marital status etc derived from the data as described above. The categories with the significantly different from the average sentiment were as follows. Among the private posters, the consumer group that had markedly higher sentiment than the rest were executives (managers, CEOs, CTOs etc.). This group expressed 9.4% higher sentiment than the average, positively commented on rates and bank stocks and did not make customer service complaints. The consumer groups that had markedly lower sentiment than the rest were firstly IT professionals who commented on banks' online interfaces inadequacy and were disappointed by Bank 2's IT skills offshoring and secondly those describing themselves as married or having children who expressed disappointment by the banks failing to meet the RBA level of rate cuts, were sympathetic to the low-paid banking employees and made a number of customer service complaints. Overall, private posters expressed significantly ($p<0.05$) lower sentiment than organisations and media sources.

4.5 Main seven topics of interest across banks and associated sentiment

The established main topics of interest are listed here starting from the topic with the highest reach and ending with that of the lowest reach. The topics included banks' reaction to the interest rate drop by RBA (mostly negative sentiment), banks community initiatives (neutral/positive sentiment), bank employee wages (negative sentiment), banks offshoring skills (negative sentiment), feedback on customer service across banks (mostly negative sentiment), economic reports published by banks (neutral sentiment) and banks' online interface (mostly negative sentiment).

4.6 Customer service feedback and complaints summary

Insights from the analysis of the spontaneous consumer feedback can be directly used by banks in improving customer experience and maintaining their market share.

The level of customer service provided by the banks was among the key topics of public interest. Consumer comments were providing feedback on banks' customer service online, via ATM and in the branches. Of this, up to 80% was negative. The positive feedback focused on helpfulness of staff in the branches (particularly for Bank 1). The complaints focused on banks' online interface deficiencies (particularly that of Bank 4), customer service in branches (most negative feeling being created by Bank 2), email spamming (particularly Bank 1), ATM issues and customer fees (no significant difference across banks).

5 Conclusion

In this article we present an example of a process for extracting social media data about Australian banks, analysing it and arriving at insights about the topics being discussed, the spread of discussion, the sentiment expressed and the dynamics of the above. Our approach capitalises on modern-day technology and machine learning advances on the one hand, and on the power of the open source software movement, on the other. Having applied the methodology to a real situation that followed the rates drop, we were able to discover mainstream societal responses, as well as to capture finer nuances of how public reacted to that specific event and to certain aspects of the banking industry as a whole. The methodology described can be easily adapted and applied in other studies. The techniques can be extended to deal with larger volumes of information and with more complex analysis requirements. Overall, the business community would benefit from reliable feedback on their marketing effort, industry initiatives, promotion campaigns, etc. that can be obtained from social media using these techniques.

References

Blei, D. and Lafferty, J. (2009), *Text Mining: Classification, Clustering, and Applications*, Chapman and Hall/CRC Press, chapter Topic Models.

Davi, A., Haughton, D., Nasr, N., Shah, G., Skaletsky, M. and Spack, R. (2005), 'A review of two textmining packages: SAS TextMining and WordStat', *The American Statistician* **59**(1), 89–103.

Feinerer, I., Hornik, K. and Meyer, D. (2008), 'Text mining infrastructure in R.', *Journal of Statistical Software* **25**(5), 1–54.

Grun, B. and Hornik, K. (2011), 'Topicmodels: An R Package for Fitting Topic Models', *Journal of Statistical Software* **40**(13), 1–30.

Hu, M. and Liu, B. (2004), 'Mining and summarizing customer reviews.', *KDD-2004* .

Liu, B. (2010), *Sentiment analysis and subjectivity. Handbook of Natural Language Processing*, second edn.

Phan, X., Nguyen, L. and Horiguchi, S. (2008), Learning to classify short and sparse text & web with hidden topics from large-scale data collections, *in* 'Proceedings of the 17th International World Wide Web Conference (WWW 2008)', Beijing, China, pp. 91–100.

Wilson, T., Wiebe, J. and Hoffmann, P. (2005), Recognizing contextual polarity in phrase-level sentiment analysis, *in* 'Proceedings of HLT-EMNLP-2005'.

Predicting usefulness of online reviews using stochastic gradient boosting and randomized trees

Madhav Kumar[1] **Shreyes Upadhyay**[2]

[1] Fractal Analytics,
New Jersey, USA
Email: madhavkumar2005@gmail.com

[2] Diamond Management and Technology Consultants
Mumbai, India
Email: shreyes.upadhyay@gmail.com

Abstract

This paper presents our analysis of online user reviews from different business categories posted on the internet rating and review services website Yelp. We use business, reviewer, and review level data to generate predictive features for estimating the number of useful votes an online review is expected to receive. Unstructured text data are mined using natural language processing techniques and combined with structured features to train two different machine learning algorithms - Stochastic Gradient Boosted Regression Trees and Extremely Randomized Trees. The results from both of these algorithms are ensembled to generate better performing predictions. The approach described in this paper mirrors the one used by one of the authors in a Kaggle competition hosted by Yelp. Out of 352 participants, the author stood 3^{rd} on the final leaderboard.

Keywords: gbm, helpfulness, online user review, opinion mining, randomized trees, text mining

1 Introduction

Virtual social infrastructures have created a market place for opinions where insights are exchanged to facilitate informed consumer decision making. From an economic perspective, these opinion markets tend to be both efficient and effective with potential safeguards against information asymmetry for the consumers and a low cost marketing channel for the producers. Between the consumers and the producers are e-commerce websites which are either directly (e.g., Amazon[1], ebay[2], App store[3]) or indirectly (e.g., Yelp[4], FourSquare[5], IMDB[6]) involved with the sales of the product. Understanding this tripartite structure and the dynamics behind it is crucial for both producers, since these opinions tend to have a direct impact on product sales (Duan et al. 2008, Ghose and Ipeirotis 2011), and for e-commerce retailers, given that the cost of consumer migration from one retailer to the other is negligible.

The principal constituent of these opinion markets are online reviews posted by consumers based on their experience and/or knowledge of the product. Online reviews provide a wealth of information about the characteristics of the product and its quality. These pieces of information help potential consumers in making an informed choice before purchasing the product. Hence, it is important that this information be available to the consumer in a easily digestible and succinct manner. However, on popular websites like Amazon and IMDB, this information can be spread out in multiple pages with thousands of free text reviews, not all of which are equally relevant. Digging through this text overload to get to relevant pieces of information can be inefficient and needless to say, extremely difficult.

How does one then decide which information is relevant and which is not? To put this into perspective of online buyers and e-commerce retailers, how does one decide which reviews are useful and which are not? Amazon answered this question and hit a $2.7B jackpot (Spool 2009) when it provided users with the option of voting for a review that they found helpful. Most websites that host review services include this feature now. This partially solves the problem of filtering out useful reviews from the others. However, among the ones that did not get helpful/useful votes, which ones were unhelpful and which were simply not read? This information is not easily available and by not accounting for it, the system as a whole becomes inefficient by losing out on potentially useful data. Given the competitive nature of such websites, it is imperative that the user experience be as smooth as possible with the most relevant information be available with minimum cognitive effort.

In this paper we have attempted to solve the generic problem faced by e-commerce retailers and review service websites – of minimizing text overload by prioritizing information based on usefulness. Specifically, we try to identify how many viewers will find a particular review to be useful. For empirical analysis, we used online reviews posted on Yelp. We then mined the text using natural language processing techniques and trained machine learning algorithms to estimate the number of useful votes a review is expected to receive. The approach described in this paper is largely what one of the authors followed while participating on a online data mining competition

[1] www.amazon.com
[2] www.ebay.com
[3] https://itunes.apple.com/us/genre/ios/id36?mt=8
[4] www.yelp.com
[5] www.foursquare.com
[6] www.imdb.com

conducted by Yelp on Kaggle[7].

Kaggle is an online portal that hosts data mining competitions for large corporations, start-ups, governments, and universities. Work on Kaggle is largely competitive where the data are twisted and turned to squeeze out tiny pieces of information for marginal improvements. Hence, not all methodologies applied there are applicable in a research driven setting. Keeping this in mind we present a pruned version of the approach that will provide a business and statistical view of the problem along with a thorough explanation of our solution.

This paper is divided into 8 sections. Section 2 describes the data mining problem followed by the literature review in section 3. Sections 4 and 5 present a detailed description of the data and our pre-processing work respectively. Section 6 describes the modeling process along with the algorithms used. Section 7 presents the results and section 8 concludes the paper.

2 Problem definition

The data on websites like Yelp are built on three verticals - 1) the viewers, 2) the reviewers, and 3) the businesses. Since this is a dynamic portal, the role of viewers and reviewers is interchangeable. At any given point of time, viewers are essentially the current consumers of Yelp and the potential consumers of a business. They view what different businesses have to offer and make their consumption decision based on the reviews posted by reviewers. To facilitate this decision making process, Yelp provides viewers with three pieces of information – 1) a quantitative field which denotes the average star rating of the business till date, 2) a second quantitative field (multiple instances) for the star rating given to the business by a particular reviewer, and 3) actual text written by reviewers describing their opinion about the business. Many reviews provide a detailed description of the reviewer's personal experience and a justification of the star rating. However, not all of these opinions are useful to viewers. While some are very well written providing enough details about the business and the experience, some are curt statements with a few words of praise or critique, and some are just spam.

Reviews for popular businesses can go up in hundreds with a certain proportion falling in each bucket as described above. Such volume of data leads to information overload for the viewer. On top of it, the presence of unhelpful matter and spam makes the problem worse by wasting the viewers' time as they dig through useless text. To overcome this problem Yelp provides viewers the option of voting for reviews that they find helpful. The votes can be assigned to one or more of the following three attributes:

1. Was this review useful?
2. Was this review funny?
3. Was this review cool?

Aggregation of votes over time allows easy distillation of reviews based on their usefulness to viewers. When a new viewer then visits the page for the business, the most useful reviews can be shown on top for fast and effective decision making.

Using the votes as a indicator for usefulness is a fast and easily scalable method. However, judging usefulness based on this sole criterion would not be statistically robust or economically efficient even though it might be directionally correct. This methodology raises at least three concerns that we present below:

Age of review (in days)	Mean of useful votes
[0, 30]	0.741
(30, 180]	0.814
(180, 540]	1.037
(540, 1000]	1.199
(1000, 2875]	2.228

Table 1: Mean useful votes with increasing age bracket

Inflated useful votes – reviews that get more useful votes tend to get more attention from other viewers which leads to further increase in the number of useful votes. Such a cycle inflates the true usefulness of the review as compared to the other reviews only because they receive more visibility which eventually leads to over-shadowing of the usefulness of less read reviews.

Age bias – reviews tend to accumulate votes over time. Any review that has been recently posted might not get noticed, especially if the business already has a good number of reviews. Such situations reinforce the point above for reviews that do not get highlighted due to over-crowding of older reviews that have been voted useful by many viewers. Table 1 presents this scenario with the mean of the number of useful votes received split by age brackets. The age bracket shows range of the difference in days between the review draft date and the data snap shot date.

Not useful vs. not read – Yelp does provide viewers the opportunity to vote for reviews that they find useful. However, there is no way to highlight whether the other reviews were not useful or were simply not read. Unlike Amazon or IMDB, Yelp does not have a ballot to vote for reviews that were not useful. For example, a viewer might read the first 2–3 reviews and vote them to be useful. This does not imply that the reviews below were not useful; they might not have received the attention of the viewer. This again follows in line with the first point that reviews that initially receive a few useful votes tend to develop on those votes to receive more and the cycle continues.

We believe that the search for solutions to these problems motivated Yelp to host an open data mining competition. The goal of the competition was to identify reviews that viewers would find most useful. Statistically, the objective was to predict the number of useful votes a review is expected to receive based on the business, reviewer, and review level information.

3 Literature review

There has been considerable research under the umbrella of opinion mining each with a slightly different

[7] www.kaggle.com

flavor of definition, data, or algorithm but widely different presentation of results. Many previous studies have taken data from Amazon (Forman et al. 2008, Kim et al. 2006, Ghose and Ipeirotis 2011, Liu et al. 2007, Mudambi and Schuff 2010) as the base for their analysis. In addition, data from CNETD (Cao et al. 2010), IMDB (Liu et al. 2008), and Epinions (Turney 2002) have also been used for analyzing opinions. This is understandable considering that these websites are the forerunners in their respective categories and have a large user base.

While there has been consensus on the sources of data, the definition of the problem has witnessed remarkable variation. Kim et al. and Ghose et al. estimate the helpfulness of a review by defining it as the ratio total number of helpful votes divided by the total number of votes (Kim et al. 2006, Ghose and Ipeirotis 2011). Defining the problem in terms of the helpfulness ratio creates an imbalance since only reviews that have received any votes can be considered. For example, Liu et al. only consider reviews that have received at least 10 votes (Liu et al. 2008). A different approach is taken by Ghose & Iperiotis who convert helpfulness to a binary variable by mapping the helpfulness ratio to 0–1 based on a threshold value (Ghose and Ipeirotis 2011). Cao et al. convert the problem to one of ordinal logits and model on the number of helpful votes with an upper bound at 7 (Cao et al. 2010). While this approach circumvents the helpfulness ratio imbalance, it places an arbitrary artificial cap on the number helpful votes a review is expected to receive. Liu et al. go a step further by manually categorizing reviews according to their helpfulness as either "best", "good", "fair", or "bad" (Liu et al. 2007). They define the helpfulness criteria based on the specification quality of the review. Though they have invested considerable effort in understanding the helpfulness of reviews, their working sample is small (4,909 reviews) and limited in scope to digital cameras only.

In terms of analyzed features, most variables from previous papers can be categorized as per the of classification Kim et al. into – a) structural (e.g., punctuation, review length, number of words), b) lexical (e.g., n-grams), c) syntactic (e.g., part of speech tagging), d) semantic (e.g., information about the product), and e) meta-data (e.g., user and product level information) (Kim et al. 2006). For example, Cao et al. describe their features as basic, stylistic, and semantic (Cao et al. 2010). Their basic features include the age of the review in days, the extremeness level of the review, and whether the reviewer wrote in different sections of the review. Their stylistic features are similar to the structural ones mentioned above, and the syntactic features are a mix of the semantic and lexical from Kim et al. Ghose et al. use the reviewer data as well in their analysis. Specifically, they create features from the reviewer's profile and reputation (Ghose and Ipeirotis 2011). In addition, they also include information on readability and subjectivity of the review.

Much of the relevant literature is dominated by the use Support Vector Machines (SVM) (Burges 1998). Using the radial basis function (RBF) to predict helpfulness of reviews seems to have provided promising results (Kim et al. 2006, Liu et al. 2007, 2008). Ghose et al. deviate from this by first doing an exploratory study using 2SLS with instrumental variables (Wooldridge 2002) to predict the logarithm of helpfulness (Ghose and Ipeirotis 2011) and then building a binary classification model using Random Forests (Breiman 2001). Similar exploratory approach is used by Forman et al. except that they model directly on helpfulness without transforming it (Forman et al. 2008). Cao et al. add more variety to the choice of algorithms by using ordinal logits for their predictive model (Cao et al. 2010).

To summarize, considerable effort has gone into developing the intellectual capital surrounding opinion mining with each new piece of research building upon the previous one and increasing the community knowledge base in this area. In our paper, we extend the scope of these previous research efforts by – 1) considering reviews that are not limited to a single category and thereby eliminating any industry or product specific bias; 2) removing the helpfulness ratio imbalance by analyzing all reviews irrespective of whether they have or have not received any useful/helpful votes; 3) applying two different machine learning algorithms and ensembling their outputs in an effort to improve the accuracy of the predictions and reduce variance of the system.

4 Data

Data were provided by Yelp for 252,863 reviews out of which useful votes for 222,907 reviews were available for training. These reviews were subset from the universe on Yelp in two ways – 1) spatially by considering businesses only in the state of Arizona and 2) temporally by considering all reviews up till Jan 19, 2013 for training and from Jan 20, 2013 till Mar 12, 2013 for testing.

The data were stored in 4 file sets - review, user, business, and checkin. Each file set had two files, one for training and the other for testing. A description of these file sets is provided below.

4.1 Review

The review data were unique at a review id level. They contained the actual review in the form of free text along with the following structured attributes – review draft date, star rating given by the reviewer to the business, user id of the reviewer, and the business id of the business. The training portion also included the number of useful, funny, and cool votes received by the review. The values in the votes variables were the cumulative number of votes received for each attribute by the review till the data snapshot date – Jan 19, 2013. From the three votes variables, the number of useful votes was the target and much of the attention in this paper is focused on this variable.

Table 2 provides the frequency of the number of useful votes received by reviews in the training data. About 95,000 (41%) reviews had not received any useful vote and about 65,000 (28%) reviews had received only 1 useful vote.

4.2 Reviewers

There were 51,296 reviewers present in the data set. For most reviewers, Yelp stores information on the total number of funny, useful, and cool votes received along with the name, average rating given by them to different businesses, and total number of reviews posted. However, some users can choose to keep their profiles private, in which case, none of the fields mentioned above are populated. In addition, to prevent an easy mapping of the number of useful votes received by a particular review and the total number of useful votes received by all the reviews written by a particular reviewer, the votes information was not provided for reviewers solely present in the testing

Useful votes	Frequency
0	95,370
1	65,301
2	31,466
3	15,351
4	7,997
5	4,560
6	2,773
7	1,814
8	1,308
9	896
>=10	3,071

Table 2: Frequency table of number of useful votes

set. To summarize, we had three types of reviewers along with their availability of information

1. 43,873 *training reviewers* with complete data on historical statistics of votes, rating, and review count publicly available. These reviewers were present only in the training data.
2. 5,105 *testing reviewers* with data on historical statistics of rating and review count but no information on votes. These reviewers were present only in the testing data. There are two points to note about these reviewers:
   (a) Their historical data on votes is available publicly but was not provided during the competition
   (b) They only formed a subset of testing data. Reviews written by training reviewers could be part of the testing data if they were drafted after Jan 19, 2013
3. 2,318 *private reviewers* with no data for votes, rating, or review count. These reviewers chose to keep their profiles private and hence no information for them is publicly available. They were present both in the training and testing data.

4.3 Businesses

The reviews were for 12,742 businesses belonging to 523 categories. However, neither the businesses nor the categories were mutually exclusive. For example, food joints with multiple branches like KFC were counted as separated businesses in overlapping categories of fast food and restaurants. The categories for different businesses ranged from restaurants to stores to services shops to pet groomers. Figure 1 shows the number of reviews by popular business categories. About 76% of the reviews in the data set were for food, restaurants and related businesses.

For each business, information on its name, address, city, state, latitude, longitude, whether still open or closed, categories, total number of reviews, and average star rating was provided.

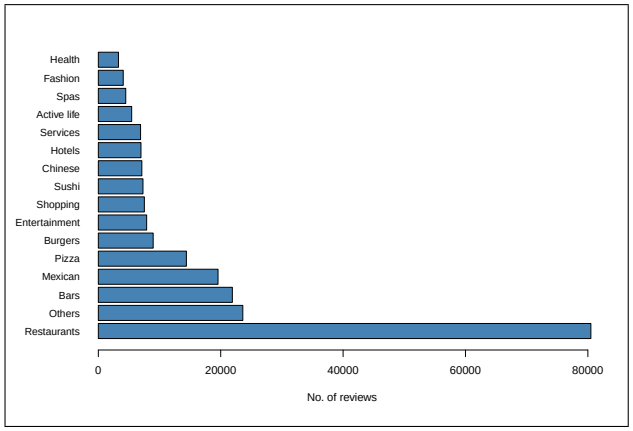


Figure 1: No. of reviews by business categories

4.4 Checkin

Checkin data were at a business id level. They were available for 9,016 businesses and included the cumulative number of checkins during each hourly interval of the day for each day of the week till the data snapshot date. Most of the data were sparse with a high number of missing values for each checkin point.

5 Pre-processing

Data were cleaned and processed to convert raw variables and unstructured text data into meaningful and algorithm readable features. It was done in two stages – feature creation and feature selection. Details for both have been provided below.

5.1 Feature creation

Four feature sets were created from review, reviewer, business, and checkin data for building models to predict the number of useful votes a review is expected to get. The feature sets included structural, lexical, meta, and interaction features.

5.1.1 Structural features

Structural features were created at a review level and contained information on the writing framework behind each review. These included the length of the review, number of sentences, number of lines, number of words, number of punctuation marks, number of numbers, number of capitalized words, and presence of a url. Parts of speech were tagged to calculate the number of adjectives, nouns, and verbs. The star rating given by the reviewer to the business was taken directly from the original data. Lastly, age of the review was calculated as the difference in days between the review draft date and the data snapshot date.

5.1.2 Lexical features

Lexical features included document-term matrices of n-grams $\{n : 1, 2, 3\}$. A document-term matrix is a two dimensional array created from a set of text documents. Each document forms the rows of the array and each unique word from all the documents shapes the columns (Feinerer et al. 2008, Feinerer and Hornik 2013, Manning et al. 2008). The cells in the array are filled with frequency count of the word occurring (defined by the column) in each corresponding document (defined by the row). Using the frequency count of the

words to fill the cells is known as the term frequency (tf) weighing scheme. Another weighing scheme popularly used is term frequency-inverse document frequency (tf-idf). Inverse document frequency of a term is given by

$$idf_t = \log(\frac{N}{df_t}) \quad (1)$$

where t is the term, N is the total number of documents, and df_t is the number of documents that contain the term t. The value in each cell is then equal to $tf \times idf$.

We created document-term matrices with both weighing schemes - term frequency and term frequency-inverse document frequency (tf-idf). N-grams were created after removing common English stop words and stemming using the Porter Stemmer (Manning et al. 2008). A sparsity threshold of 99.9%–99.99%[8] was chosen to constrain the data set to a manageable size. Finally, a total of 3,840 1-grams, 6,116 2-grams, and 4,281 3-grams were created with both weighing schemes respectively.

5.1.3 Meta features

Meta features were created at reviewer and business levels. For reviewers, summary statistics for length of reviews, age of reviews, distinct number of categories reviewed, and distinct number of businesses reviewed were calculated. The total number of reviews and the total number of useful votes were taken directly from the reviewer files. For testing and private reviewers, the total number of useful votes were imputed using the mean value from the training reviewers.

Similar summary statistics were extracted for businesses as well except for the number of useful votes. In addition, binary variables from categories were created to indicate if the business belonged to a particular category or not. Checkin data were also mined, however, due to their high level of sparsity only three meaningful aggregate variables were extracted - mean, max, and sum of all checkins till the data snapshot date.

5.1.4 Interaction features

Interaction features included products/ratios between structural and meta features, and unsupervised clusters using reviewer meta data. For example, the product of length of review and age of review, the product of length of review and the number of distinct business categories reviewed, and a set of ten clusters based on the number of reviews by a reviewer and the average number of useful votes received by the reviewer. The unsupervised clustering was performed using k-means clustering. Most of these features had little intuition behind them and were mainly derived for competitive gain on leaderboard.

5.2 Feature selection

Feature selection largely revolved around the lexical feature set. The idea was to select the best subset of features that – 1) would not demand exceptionally high computation time and power, and 2) give reasonably comparable accuracy as the larger set. We used two methods for feature selection – regularized linear regression and principal component analysis.

[8] Sparsity threshold of 99.9% implies that the terms should be present in atleast 0.1% of all reviews

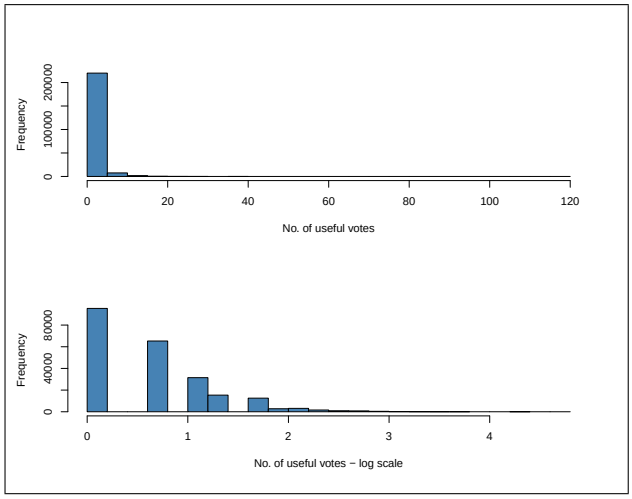


Figure 2: Distribution of the number of useful votes

5.2.1 Regularized linear regression

Regularized linear regression, using glmnet (Friedman et al. 2010), was run for each document-term matrix separately and the best features were selected through a two-level 25 (5x5) fold cross validation procedure. The first level involved cross-validating through the training data and the second level involved cross-validating within each fold from the first level to select the best parameter value (lambda) for glmnet. For the entire analysis the alpha parameter was fixed at 1 as required for a LASSO penalty (Friedman et al. 2010, Tibshirani 1996).

5.2.2 Principal component analysis

Randomized PCA using singular value decomposition (Pedregosa et al. 2011) was performed on each document-term matrix with the number of components fixed at 50.

6 Modeling

The objective of the competition required to build a model to predict the number of useful votes a review is expected to receive. The model's accuracy were to be judged by comparing the predictions to the ground truth values from the testing data using the root mean square log error (rmsle) as given in equation 2

$$rmsle = \sqrt{\frac{\sum_{i=1}^{n}(log(y_i+1) - log(\hat{y}_i+1))^2}{n}} \quad (2)$$

where n is the total number of reviews, y_i is the actual number of useful votes a review received, and $\hat{y}_i$ is the predicted number of useful votes a review is expected to receive. Most regression algorithms operate by minimizing the root mean square error instead of the root mean square log error. Keeping this in mind, we modified our target variable by taking its natural log, i.e., $\tilde{y} = \ln(y+1)$, where y is the number of useful votes and the 1 is added to check for reviews with zero votes. Figure 2 shows the distribution of the original dependent variable and the log transformed dependent variable.

Many previous winning methods on Kaggle[9] used tree based methods like Random Forests and Boosted

[9] www.kaggle.com

Regression Trees. We used both these methods and combined their results to generate the final predictions. The models were trained using 85% of the original training sample, keeping the remaining 15% as a hold-out for parameter selection and model calibration. A brief description of these methods and our selected hyper-parameters is provided below.

6.1 Stochastic Gradient Boosted Regression Trees (GBM)

Gradient Boosting is an algorithm that builds stage-wise additive models in a greedy fashion (Friedman 2001). Stochastic Gradient Boosting allows each additive model to be built on a random sub-space of training observations by randomly sampling with replacement before each iteration (Friedman 2002). The training model within the algorithm is called a base learner. For our purpose, we used the GBM implementation in R (Ridgeway 2013) with regression trees as base learners. We built multiple GBM models using different combinations of structural, meta, and interaction features along with different parameter settings and validated each of them independently on the hold-out sample. The two main hyper-parameters, shrinkage and number of trees, were chosen keeping the accuracy–computation time trade off in mind. The parameter settings for the best model were as follows – shrinkage = 0.04, number of trees = 4000, interaction depth = 7, minimum observations in a node = 70, and percentage of training data to be randomly sampled for building each tree = 50%.

6.2 Extremely Randomized Trees (ERT)

Extremely randomized trees is an algorithm similar to Breiman's Random Forests (Breiman 2001). Random Forests involves creating multiple bagged (bootstrap aggregated) trees each over a random subspace of features. The predictions are then averaged over the entire forest. In Random Forests, each parent node is split into further nodes by choosing the best split among the random subset of features chosen for the particular tree (Breiman 2001, Liaw and Weiner 2002). ERT is different from this in two ways – 1) it builds each tree on the same sub-sample of training observations, 2) it first generates contender split thresholds randomly and then chooses the best split from these thresholds (Geurts et al. 2006). By adding randomness in splits, ERT reduces the variance of the system more than Random Forests do but at the expense of an increased bias (Pedregosa et al. 2011).

We used the scikit-learn implementation in python (Pedregosa et al. 2011) to build Extremely Randomized Trees. We built 150 trees by allowing them to grow fully using the combination of *max_depth = None* and *min_samples_split = 1* as parameters. All four feature sets were used for building the model. Only one document-term matrix was used at a time from the lexical feature sets. A total of six models were built, each time with a different document-term matrix. Out of the six contender feature matrices, bi-grams of term frequencies had the lowest error on the holdout sample.

6.3 Ensemble

Ensembling is a technique that involves combining multiple models to improve the accuracy of the overall system. It is the same concept that is used internally while building a Random Forest (Breiman 2001) or training a Gradient Boosting Machine (Friedman 2001) and can be generalized to combining models trained using different algorithms. Ensembling different models was popularized during the Netflix Prize [10] when the milestone and final winners published their methodology (Koren 2009, Piotte 2009, Toscher and Jahrer 2009). We combined two models–a GBM and an ERT using a simple average with equal weights to create an ensemble of their predictions.

7 Results

7.1 Important factors for useful reviews

Prior research has shown varied results as to which variables are most important for predicting usefulness of reviews. Cao et al. found that semantic characteristics (substance of the review) provide most important pieces of information towards usefulness (Cao et al. 2010) . Using Random Forests and converting helpfulness to a classification problem Ghose et al. concluded that readability, subjectivity, and reviewer characteristics to be of equal predictive power (Ghose and Ipeirotis 2011). Using non-linear SVMs, Liu et al. found reviewer expertise, writing style, and timeliness to be important predictors for review helpfulness (Liu et al. 2008).

Our results show that reviewer characteristics are most influential in predicting the number of useful votes. The single most important variable in our model was the reviewer's cumulative number of useful votes for till date. This was followed by a set of structural features like the length of the review, the age of the review, and the number of lines in a review. We found lexical features to be least predictive among all the four feature sets. Table 3 presents a list of top 12 variables in order of their importance from GBM.

Rank	Variable
1	No. of useful votes of the reviewer
2	Age of review
3	Length of review × Age of review
4	Length of review × No. of distinct categories reviewed
5	No. of lines in review
6	Reviewer cluster (categorical)
7	No. of useful votes of the reviewer × Age of review
8	Average age of all reviews by the reviewer
9	Star rating by reviewer
10	No. of reviews by reviewer
11	Length of review
12	Average age of all reviews by the reviewer

Table 3: Important variables from GBM

[10] http://www.netflixprize.com/

Model	Data sample		
	Holdout	Public LB	Private LB
GBM	0.4505	0.4506	0.4535
ERT	0.4577	0.4508	0.4514
Ensemble	**0.4469**	**0.4446**	**0.4462**
Rank 1 model	-	0.4390	**0.4404**
Rank 2 model	-	**0.4377**	0.4408

Table 4: RMSLE of models on different data samples

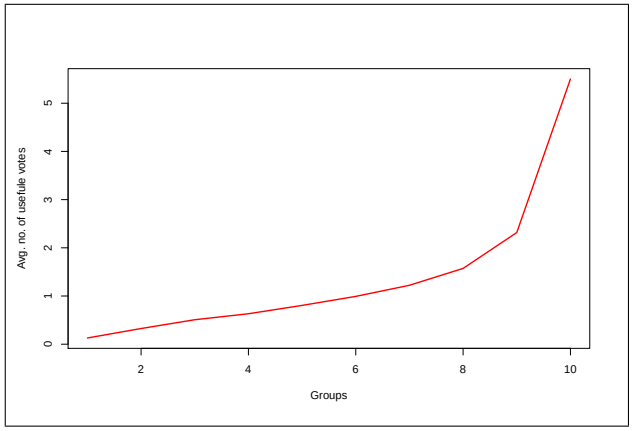


Figure 3: Lift separation based on predicted values

7.2 Model performance

Our ensemble model was the third best entry on the private leaderboard as judged by Equation 2. Table 4 presents a comparison of the relative scores of our models on the hold out sample, public leaderboard (LB), and the private leaderboard. The scores of the best two models on the leaderboards are also shown.

Figure 3 shows the lift separation provided the predictions from the model. The graph is generated by arranging the ground truth values based on the predictions, bucketing them in to 10 groups using the prediction order, and taking the mean of the ground truth values in each group. Based on the graph, the model gives a Lift Ratio [11] of 3.91 in the highest group and a normalized gini of 0.7687 on the holdout sample.

8 Conclusion and future work

Our paper builds on the previous work by adding value in three different ways – 1) we consider reviews for different and diverse business categories ranging from restaurants to pet shops which makes our results easily generalizable across different product environments; 2) we analyze all reviews irrespective of the whether they have received any useful vote till date or not; 3) we build an ensemble of ensembles by training two different machine learning algorithms and combining their results for superior accuracy.

In addition to the above three points, our results are based on a sample of more than 250,000 reviews by 50,000 reviewers across 12,000 businesses. According to our knowledge, this is considerably larger than any other sample analyzed in previous research efforts.

We base our results on two robust and powerful machine learning algorithms trained on a large and varied sample using structural, lexical, reviewer, and business characteristics. Our results suggest that reviewer information is most important in predicting usefulness of reviews. These results are similar to those of Forman et al. (Forman et al. 2008). We also find that lexical features derived from the review text are least influential in predicting the number of useful votes a review will receive. These results are contradictory to that obtained by Cao et al. (Cao et al. 2010). However, Cao et al. do not consider reviewer information in their analysis making it difficult to directly compare our results with theirs. Lastly, our model performs well on unseen new data. Competing against 351 other models, our model ranked 3^{rd} on the final leaderboard standings.

Though our work augments other research undertakings in different ways, it is constrained by many limitations. First, we do not explore the structure of the review fully. For example, Ghose et al. use multiple readability indexes to estimate the effort required by the viewer in reading the review (Ghose and Ipeirotis 2011). They also include subjectivity analysis in their model and their results suggest that reviewer characteristics, readability, and subjectivity are equally important in predicting helpfulness of reviews. Including these additional dimensions in our work will help provide a better understanding of the reviewer-usefulness complexity. Second, the hyperparameters of our training algorithms were based on previous experience, reference manuals, and results of past Kaggle competitions hinting that they might be sub-optimal for this particular problem. A better approach would be to do a grid search across the length of the training data using cross-validation to determine the optimal values. Third, the objective of our study was more predictive rather than exploratory in nature. To this goal, we used algorithms that are proven to be more accurate but partially black-box, eclipsing the true underlying relationship between the target and the predictor variables.

Our paper provides valuable extensions across multiple dimensions of opinion mining. However, given the limitations mentioned above, this paper can be extended in depth and breadth in many different ways. For instance, our results hold good for reviews that have reviewer characteristics available. But there are numerous reviews written by anonymous reviewers. Directly implementing our model on these might not yield the best results. A more suited methodology would be to deal with these reviews as a cold-start problem and analyze them separately. Another challenging problem to solve would be to build a system for judging review usefulness that develops in a self-correcting manner through feedback loops by factoring in new reviews at regular intervals (Liu et al. 2008). Such a system would prove useful for review websites like Yelp which are devoted to service oriented businesses.

[11] Lift Ratio = $\frac{\mathbb{E}(Target|Segment)}{\mathbb{E}(Target|Population)}$

References

Koren, Y., (2009), The BellKor Solution to the Netflix Grand Prize, Available from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.162.2118

Breiman, L. (2001), 'Random forests', *Machine Learning*, Vol. 45.1, pp. 5–32.

Burges C.J.C.,(1998), 'A tutorial on support vector machines for pattern recognition', *Data Mining and Knowledge Discovery*, Vol. 2, Issue 2, pp. 121–167.

Cao, Q., Duan, W., & Gan, Q., (2011), 'Exploring Determinants of Voting for the "Helpfulness" of Online User Reviews: A Text Mining Approach', *Decision Support Systems*, Vol. 50, Issue 2, pp. 511–521.

Duan, W., Gu, B., & Whinston, A.B., (2008), 'The dynamics of online word-of-mouth and product sales an empirical investigation of the movie industry', *Journal of Retailing*, Vol. 84, Issue 2, pp. 233–242.

Feinerer, I., Hornik, K., & Meyer, D., (2008), 'Text mining infrastructure in R', *Journal of Statistical Software*, Vol. 25, Issue. 5, pp. 1–54

Feinerer, I., & Hornik, K., (2013), 'tm: Text mining package. R package version 0.5-8.3.', Available from: http://CRAN.R-project.org/package=tm

Forman C., Ghose, A., & Wiesenfield, B., (2008), 'Examining the relationship between reviews and sales: the role of reviewer identity disclosure in electronic markets', *Information Systems Research*, Vol. 19, Issue 3, pp. 291–313.

Friedman, J. (2001), 'Stochastic gradient boosting', *Computational Statistics & Data Analysis*, Vol. 38, pp. 367–378.

Friedman, J. (2002), 'Greedy function approximation: A gradient boosting machine', *The Annals of Statistics*, Vol. 29, pp. 1189–1232.

Friedman, J., Trevor, H., & Tibshirani, R., (2010), 'Regularization paths for generalized linear models via coordinate descent', *Journal of statistical software*, Vol. 33, Issue 1, pp. 1–22.

Geurts, P., Ernst, D., & Wehenkel, L., (2006), 'Extremely randomized trees', *Machine Learning*, Vol. 63, Issue 1, pp. 3–42.

Ghose, A., Ipeirotis, P.G. (1993), 'Estimating the helpfulness and economic impact of product reviews: Mining text and reviewer characteristics', *'IEEE Transactions on Knowledge and Data Engineering'*, Vol. 23, Issue 10, pp. 1498–1512.

Liaw, A. & Weiner, M. (2002), 'Classification and regression by randomForest', *R News*, Vol. 2/3, pp. 18–22.

Liu, J., Cao, Y., Lin, C.Y., Huang, Y., & Zhou, M., (2007), 'Low quality product review detection in opinion summarization', *Joint conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 334–342.

Liu, Y., Huang, X., An, A., & Yu, X., (2008), 'HelpMeter: A nonlinear model for predicting the helpfulness of online reviews', *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Vol. 1, pp. 793–796.

Kim, S.M., Pantel, P., Chklovski, T., & Pennacchiotti, M., (2006), 'Automatically assessing review helpfulness', *Conference on Empirical Methods in Natural Language Processing*, pp. 423–430.

Manning, C., Raghavan, P., & Schutze, H., (2008), 'Introduction to Information Retrieval', *Cambridge University Press*

Mudambi, S., & Schuff, D., (2010), 'What makes a helpful online review? A study of customer reviews on amazon.com', *MIS Quarterly*, Vol 34, Issue 1, pp. 185–200

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E., (2011), 'Scikit-learn: Machine learning in python', *Journal of Machine Learning Research*, Vol. 12, pp. 2825–2830.

Piotte, M., & Chabbert, M., (2009), The Pragmatic Theory Solution to the Netflix Grand Prize, Available from: http://www.netflixprize.com/assets/GrandPrize2009_BPC_PragmaticTheory.pdf

Ridgeway, G. (2013), 'gbm: Generalized Boosted Regression Models', Url. http://CRAN.R-project.org/package=gbm

Spool, J., (2009), The magic behind Amazon's 2.7 billion dollar question, Available from: http://www.uie.com/articles/magicbehindamazon/

Tibshirani, R., (1996), 'Regression shrinkage and selection via the Lasso', *Journal of the Royal Statistical Society*, Vol. 58, Issue. 1, pp. 267–288.

Toscher, A., & Jahrer, M., (2009), The BigChaos Solution to the Netflix Grand Prize,

Turney, P., (2002), 'Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews', *Proceedings of the 40^{th} Annual Meeting of the Association for Computational Linguistics*, pp. 417–424

Wooldridge, (2002), 'Econometric analysis of cross-sectional and panel data', *MIT Press*, Cambridge, MA.

Predictive Modelling Using Random Forest and Its Hybrid Methods with Geostatistical Techniques in Marine Environmental Geosciences

Jin Li
Coastal, Marine and Climate Change Group
Environmental Geoscience Division, Geoscience Australia
GPO Box 378, Canberra 2601, ACT

jin.li@ga.gov.au

Abstract

The accuracy of spatially continuous information of seabed sediments, usually generated from point samples using spatial interpolation methods (SIMs), is crucial for evidence-based decision making in marine environmental management and conservation. Improving the accuracy by identifying the most accurate methods is essential, but also challenging since the accuracy is often data specific and affected by multiple factors. Because of its high predictive accuracy, random forests (RF) method was integrated into spatial statistics by combining it with existing SIMs, which resulted in new hybrid methods with improved accuracy. However, these methods may also be data specific. In this study, on the basis of seabed mud content data in the southwest Australian Exclusive Economic Zone, we experimentally tested: 1) the effects of input secondary variables on the performance of RF, SIMs and their hybrid methods; 2) the effects of cross-validation on their performance; and 3) whether the performance of these methods is data specific. For RF and the hybrid methods, up to 21 variables were used as predictors. The predictive accuracy was assessed in terms of relative mean absolute error and relative root mean squared error based on the average of 100 iterations of 10-fold cross-validation. The findings were compared with previous studies and discussed. The most accurate method to predict the spatial distribution of seabed mud content in the study area was identified. This study provides suggestions and recommendations for the application of these hybrid methods to spatial predictive modelling, not only in environmental sciences, but also in other relevant disciplines.

Keywords: machine learning; data mining; model selection; cross-validation; ordinary kriging; spatial prediction.

1 Introduction

Spatially continuous data of seabed sediments is often required for seascape mapping, prediction of marine biodiversity, and marine environmental planning and conservation (Pitcher et al., 2008, McArthur et al., 2010). However, spatially continuous data are not readily available and seabed sediment information is usually collected by point sampling. Spatially continuous data must then be predicted from the point samples.

Statistical and mathematical techniques for spatial prediction are essential tools for generating spatially continuous data from point data. These methods are, however, often data or variable specific and their performance depends on many factors (Li and Heap, 2011). The accuracy of the predicted spatially continuous information using spatial interpolation methods (SIMs) is crucial for evidence-based decision making for marine environmental management and conservation. Due to its high predictive accuracy in data mining and other disciplines (Cutler et al., 2007, Diaz-Uriarte and de Andres, 2006, Shan et al., 2006), random forests (RF) method was introduced to spatial statistics by combining it with commonly used SIMs to predict the spatial distribution of seabed sediments (Li, 2011, Li et al., 2010). This development opened an alternative source of methods for spatial prediction. These hybrid methods, RFOK and RFIDW (i.e. the hybrids of RF with inverse distance weighting (IDW) or ordinary kriging (OK)), have been shown to have high predictive capacity (Li et al., 2011b, Li et al., 2011c, Li et al., 2012b). However, these methods may also be data specific like other spatial prediction methods.

Model selection is often required for selecting an optimal model from a number of candidate models. RF is often argued to be insensitive to non-important variables, as it selects the most important variable at each node split (Okun and Priisalu, 2007). RF can also deliver good predictive performance even when most predictive variables are noisy (Diaz-Uriarte and de Andres, 2006). Furthermore, the performance of RF is argued to depend only on the number of strong features and not on the number of noisy variables if sample size is large (500 to 1000) (Biau, 2012). Thus, model selection becomes less important for RF and consequently for the hybrid methods as well. This assumption was tested for spatial predictions (Li et al., 2011b, Li et al., 2012a, Li et al., 2011a, Li et al., 2012b), suggesting that model selection is important for RF. A model selection procedure for RF was developed previously by Li et al. (2013) based on randomForest package in R 2.13.0 (R Development Core Team, 2011); but the idea of selecting the most important variables using RF for model development dates back to early 2007 as evidenced in Arthur et al. (2010). Since it has not been applied to any other studies, further testing is warranted.

In this study, we aim to identify the most accurate model to predict the spatial distribution of seabed mud

content in the southwest region of the Australian continental margin by testing 1) the effects of input secondary variables on the performance of RF, RFOK and RFIDW; 2) the effects of cross-validation on the performance of RF, RFOK and RFIDW; and 3) whether the performance of RF and their hybrid methods is data specific.

2 Data and Data Quality Control

2.1 Mud content data and data quality control

The mud content data used in this study was extracted from the Marine Samples (MARS) database. The accuracy and precision of attributes assigned to each sample in the MARS database varies, which can result in data noise (Li et al., 2010). Hence data quality control needs to be employed to reduce relevant data noise.

2.1.1 MARS database

The MARS database was created in 2003 with the vision of collating all existing seabed sediment sample data for the Australian Marine Jurisdiction (AMJ) into a single database (http://www.ga.gov.au/oracle/mars). The content and structure of the database, its data sources and definitions of sediment data types have been detailed in Li et al. (2010). Preliminary data quality control was performed according to Geoscience Australian Data Standards, Validation and Release Handbook (Li et al., 2010). This resulted in a total of 14,360 samples in the MARS database as recorded on 19 December 2012.

2.1.2 2.1.2. Data quality control

To reduce possible data noise, a data quality control approach similar to that detailed in Li et al. (2010) was adopted. Seven criteria were used to clean the data. The data points must:

1) be within the continental Australian Exclusive Economic Zone (AEEZ);
2) be non-dredged sample;
3) lie below sea level (i.e. non-positive bathymetry);
4) have at least 3 digits after decimal points in latitude/longitude;
5) be with a sampling base depth <=5 cm;
6) be unique (i.e. no duplicates); and
7) have sediment data (i.e. no missing value).

Criterion 4 selects samples with accurate location information and criterion 6 removes duplicated samples. Otherwise, the remaining criteria are identical to those described in Li et al. (2010). Since geomorphological features of the Australian margin and adjacent seafloor created by Heap and Harris (2008) are categorically expressed bathymetry (Li et al., 2010), a geomorphology related criterion for data quality control was not considered in this study.

The sample size was gradually reduced from 14,360 to 6,966 after applying the data quality control criteria (Figure 1). This final dataset containing 6,966 samples within AEEZ was ready to be used.

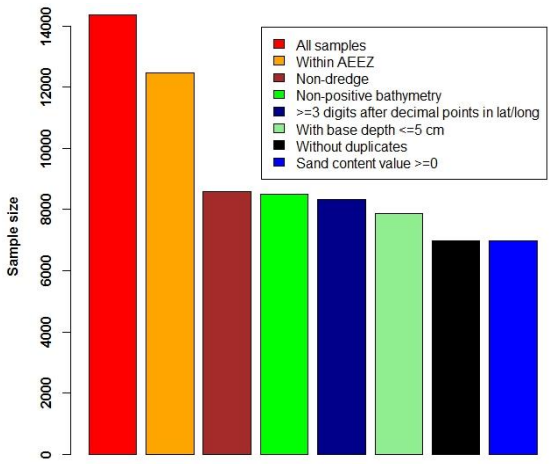


Figure 1. Changes of sediment sample size with data quality control criteria.

2.2 Study region

The study area is located in the southwest region of AEEZ (Figure 2). This region covers an area of about 513,000 km^2 and comprises four geomorphic provinces (Heap and Harris, 2008), mostly on the shelf and slope, in water depths ranging from 0 to 5917 m and adjacent to a coastline of north-south orientation.

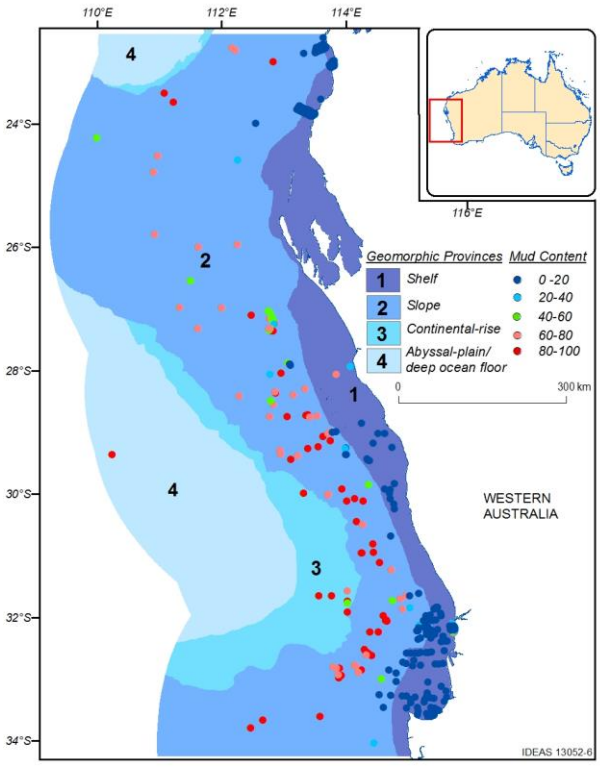


Figure 2. Spatial distribution of mud content samples and their occurrence in the geomorphic provinces

In total, 522 samples of seabed sediments were located in the study area and considered in this study following data quality control. Sample density was very low, with 1.02 samples per 1000 km^2 on average. The sediments are divided into three groups: mud, sand and gravel (Li et al., 2010). In this study, we only focused on seabed mud

content. The mean of mud content in the study area is 22.9%, with a coefficient of variation of 145.1%.

2.3 Predictors

A wide range of predictors could potentially be used as secondary information to improve the spatial prediction of marine environmental data. However, only six predictors that were justified and used in previous studies (Li et al., 2012b, Li et al., 2011b) were employed in this study. These predictors are: bathymetry (bathy), distance-to-coast (dist.coast), seabed slope (slope), seabed relief (relief), latitude (lat) and longitude (long). Of these predictors, bathymetry data was based on Whiteway (2009), and slope and relief were derived from the bathymetry data. All datasets of these variables were generated in ArcGIS at a 250 m resolution using the methods detailed by Li et al. (2010, 2012b). The coordinates were based on WGS84 in this study as explained in previous studies (Li et al., 2011b, Li et al., 2011c).

In total, 15 variables (i.e. $bathy^2$, $bathy^3$, $dist.coast^2$, $dist.coast^3$, $slope^2$, $slope^3$, $relief^2$, $relief^3$, lat^2, $long^2$, lat*long, $lat*long^2$, $long*lat^2$, lat^3 and $long^3$) were derived. These variables were used to compensate the small number of predictors as in previous studies (Li et al., 2011, Li et al., 2011d).

3 Predictive Methods, Model Development and Selection, and Model Evaluation

3.1 Methods for spatial prediction

RF, RFOK, and RFIDW were tested in this study, as were two most commonly compared SIMs (Li and Heap, 2011), IDW and OK. For RFOK and RFID, up to 21 variables were used as predictors in the RF component. The residuals of RF were then interpolated using IDW with a distance power of 2 and a searching window size of 12, and using OK with a Spherical model and a searching window size of 7.

For RF, the predictors used are identical to those used in the RF component in RFOK and RFIDW. For IDW, a searching window size of 20 was used. For OK, a square root transformation was applied, and a Spherical model and a searching window size of 20 were used.

All these parameters were chosen based on our previous findings for predicting the seabed mud content in AEEZ (Li et al., 2011, Li et al., 2011d).

3.2 Model development and selection

The model development and model selection were based on a formal model simplification procedure for RF developed by Li et al. (2013). As to the role of input variables, contradictory findings were observed in previous studies in environmental sciences. On one hand, it was observed that including some 'redundant', 'irrelevant' or correlated variables (e.g. $bathy^2$, which is correlated with bathy) could improve the predictive accuracy (Li et al., 2012a, Li et al., 2012b), suggesting that correlated variables may be able to compensate for the small number of predictors in environmental sciences. On the other hand, models without derived variables were observed to be more accurate than models with 'redundant variables', suggesting that excluding the correlated variables may improve the predictive accuracy (Li et al., 2011a, Li et al., 2011b). Therefore, we considered these findings in this study.

The models for RF, RFOK and RFIDW were initially developed by using all six predictors and their 15 derived variables. We then reduced the full model by gradually removing the least important variable(s) based on the variable importance measure by RF. After reaching the model with minimum number of predictors (i.e. only one predictor remained), we then used the six predictors only and repeated the above procedure to remove the least important variable. This was to examine any possible effects of the correlated variables on the selection of the six predictors during above model development process. In total, 24 models were developed for RF, RFOK and RFIDW (Table1).

3.3 Model evaluation

To evaluate the performance of these methods, a 10-fold cross-validation was used. To reduce the influence of randomness associated with the 10-fold cross-validation that each method may receive different samples for prediction and validation, the 10-fold cross-validation was repeated 100 times. Relative mean absolute error (RMAE) and relative root mean square error (RRMSE) (Li and Heap, 2011) were used to assess the performance of the methods tested and to compare with findings in previous studies.

The modelling was implemented in R 2.15.1 (R Development Core Team, 2012), using packages raster for extracting data from different data layers, gstat for geostatistical modelling and randomForest for random forest modelling. Predictions were corrected by resetting the faulty estimates to the nearest bound of the data range (i.e. 0 or 100%) if applicable (Goovaerts, 1997).

4 Effects of input secondary variables on the performance of RF, RFOK and RFIDW

The predictive errors of all three methods fluctuated with the input secondary variables in terms of RRMSE (Figure 3). For RF, RRMSE gradually decreased from model 1 to model 9, then slightly increased from model 10 onwards with an abrupt increase from model 14, and reached the highest value for model 19 containing only one predictor. For RFOK and RFIDW, RRMSE gradually decreased from model one to model 13, then increased from model 14 onwards and reached the maximum value for model 19 containing only one predictor. Overall, RF with model nine, and RFOK and RFIDW with model 13 were relatively more accurate than with other models. This model contains eight predictors, including six derived variables. Similar patterns were observed in terms of RMAE, so they were not presented.

5 Variation of Predictive Errors of RF, SIMs and Their Hybrid Methods with Individual Cross-validations

The predictive errors varied with iterations for IDW, OK, and the best model of RF, RFOK and RFIDW in terms of RMAE and RRMSE (Figure 4). The variation of predictive errors in the 100 times 10-fold cross-validation

Modelling.process	Predictors	No.predictors
Model 1: All 21 predictors	All 21 variables	21
Model 2: - srelief and crelief from model 1	lon, lat, bathy, dist, relief, slope, sbathy, cbathy, sdist.coast, cdist.coast, sslope, cslope, slat, clat, slon, clon, latlon, latslon, slatlon	19
Model 3: - sslope from model 2	lon, lat, bathy, dist, relief, slope, sbathy, cbathy, sdist.coast, cdist.coast, cslope, slat, clat, slon, clon, latlon, latslon, slatlon	18
Model 4: - cslope from model 3	lon, lat, bathy, dist, relief, slope, sbathy, cbathy, sdist.coast, cdist.coast, slat, clat, slon, clon, latlon, latslon, slatlon	17
Model 5: - slope from model 4	lon, lat, bathy, dist, relief, sbathy, cbathy, sdist.coast, cdist.coast, slat, clat, slon, clon, latlon, latslon, slatlon	16
Model 6: - clat from model 5	lon, lat, bathy, dist, relief, sbathy, cbathy, sdist.coast, cdist.coast, slat, slon, clon, latlon, latslon, slatlon	15
Model 7: - slat from model 6	lon, lat, bathy, dist, relief, sbathy, cbathy, sdist.coast, cdist.coast, slon, clon, latlon, latslon, slatlon	14
Model 8: - sdist.coast from model 7	lon, lat, bathy, dist, relief, sbathy, cbathy, cdist.coast, slon, clon, latlon, latslon, slatlon	13
Model 9: - relief from model 8	lon, lat, bathy, dist, sbathy, cbathy, cdist.coast, slon, clon, latlon, latslon, slatlon	12
Model 10: - latslon from model 9	lon, lat, bathy, dist, sbathy, cbathy, cdist.coast, slon, clon, latlon, slatlon	11
Model 11: - lat from model 10	lon, bathy, dist, sbathy, cbathy, cdist.coast, slon, clon, latlon, slatlon	10
Model 12: - cdist.coast from model 11	lon, bathy, dist, sbathy, cbathy, slon, clon, latlon, slatlon	9
Model 13: - dist from model 12	lon, bathy, sbathy, cbathy, slon, clon, latlon, slatlon	8
Model 14: - latlon and slatlon from model 13	lon, bathy, sbathy, cbathy, slon, clon	6
Model 15: - sbathy from model 14	lon, bathy, cbathy, slon, clon	5
Model 16: - slon from model 15	lon, bathy, cbathy, clon	4
Model 17: - clon from model 16	lon, bathy, cbathy	3
Model 18: - cbathy from model 17	lon, bathy	2
Model 19: - lon from model 18	bathy	1
Model 20: lon, lat, bathy, dist, relief, slope	lon, lat, bathy, dist, relief, slope	6
Model 21: lon, lat, bathy, dist, relief	lon, lat, bathy, dist, relief	5
Model 22: lon, bathy, dist, relief	lon, bathy, dist, relief	4
Model 23: lon, bathy, dist	lon, bathy, dist	3
Model 24: All 18 variables (the control)	All 18 variables (excluding relief and its second and third orders)	18

Table 1. Models developed after model selection for the seabed mud content in the southwest region, AEEZ

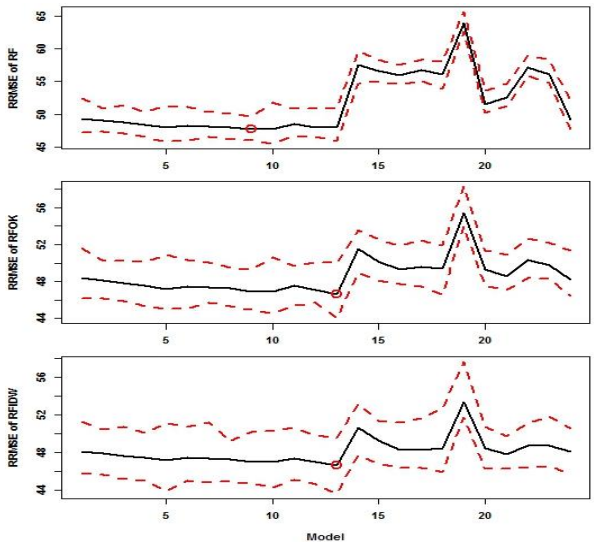


Figure 3. RRMSE (%) (mean: black line; minimum and maximum: dash lines) of 24 models of RF, RFOK and RFIDW for seabed mud content; and the model with the minimum RRMSE (red circle).

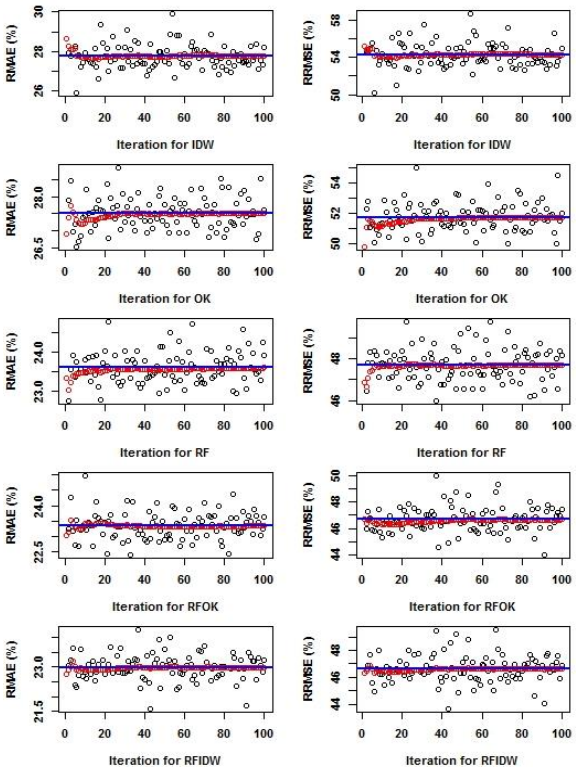


Figure 4. RMAE (%) and RRMSE (%) of IDW, OK, and the best model of RF, RFOK and RFIDW for each of 100 times of 10-fold cross-validation (black circle), their accumulative average (red circle) and overall average (blue line) for seabed mud content.

was considerable for all methods. RMAE varied between 25.9% and 29.9% for IDW, between 26.5% and 28.9% for OK, between 22.8% and 24.8% for RF, between 22.4% and 25.0% for RFOK, and between 21.6% and 24.3% for RFIDW. RRMSE fluctuated between 50.2% and 58.7% for IDW, between 49.8% and 55.0% for OK, between 46.0% and 49.8% for RF, between 44.0% and 50.0% for RFOK and between 43.7% and 49.5% for RFIDW.

The variation of predictive errors between iterations changed with methods (Figure 4). The variation of RMAE between iterations was about 4.0% for IDW. It was between 2.0–2.7% for the remaining methods, and RF was with the least variation. The variation of RRMSE between iterations was 8.5% for IDW, while it varied between 3.8–5.8% for the rest methods. Again RF was with the least variation.

The accumulative averages of RMAE and RRMSE gradually converged to the overall means for all methods as the number of iteration increased (Figure 4). It is apparent that the accumulative averages largely converged to the overall means after 60–80 iterations for OK, RF and RFOK. However, variations in the averages

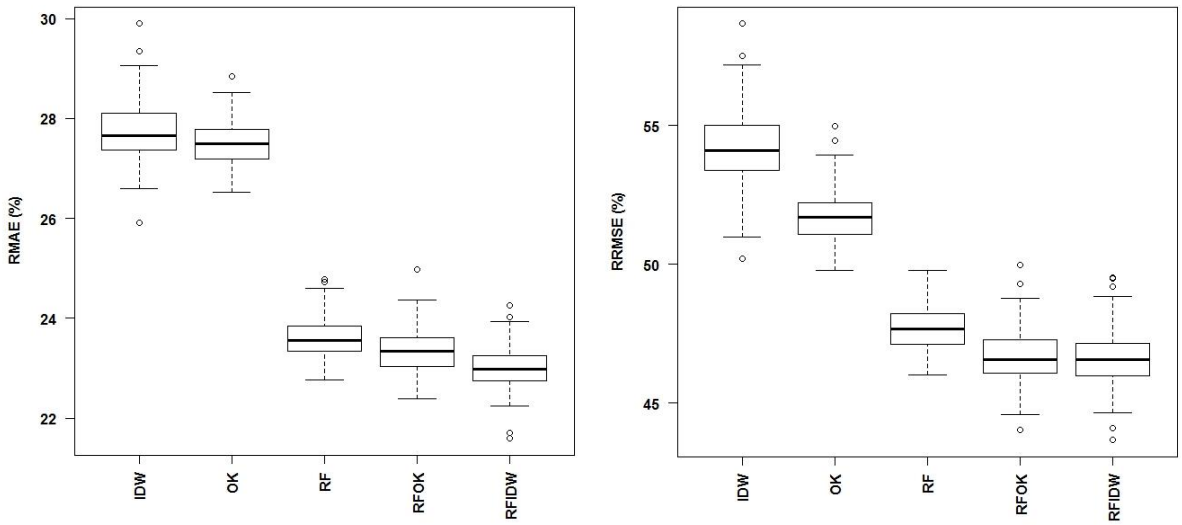


Figure 5. RMAE (%) and RRMSE (%) of IDW, OK, RF, RFOK and RFIDW for mud content: summary statistics based on the 100 times of 10-fold cross-validation.

were still noticeable after 90 iterations for IDW and RFIDW, although such variations were marginal.

6 Performance of RF, SIMs and Their Hybrid Methods

RF, RFOK and RFIDW were the most accurate three methods (Figure 5). They were significantly more accurate than the most commonly compared SIMs (i.e. IDW and OK) based on Mann-Whitney test for IDW in terms of both RMAE and RRMSE, and based on t-test for OK in terms of both RMAE and RRMSE (all with a p value < 0.0001). Of these three methods, RFIDW and RFOK were significantly more accurate than RF in terms of both RMAE and RRMSE based on paired t-test (with a p value < 0.0001). RFOK was significantly less accurate than RFIDW in terms of RMAE based on paired t-test (with a p value < 0.0001), and they were not significantly different in terms of RRMSE based on paired t-test (with a p value = 0.3843). Overall, RFIDW is preferred over RF and RFOK.

7 Discussion and Conclusions

7.1 Model selection for RF and its hybrid methods

Selecting an optimal model from a number of candidate models is crucial in model development. For regression models, this is usually achieved via model simplification procedures such as backward or forward stepwise regression, which usually identifies the most parsimonious model. This resultant model is, however, not necessarily the most accurate model in terms of predictive accuracy. For RF and its hybrid methods, the model selection has been demonstrated to be essential in environmental sciences (Li et al., 2011b, Li et al., 2012a, Li et al., 2011a, Li et al., 2012b). In this study, it was observed that the predictive errors fluctuate with the input secondary variables for RF, RFOK and RFIDW. This finding further demonstrates that model selection is essential for RF and its hybrid methods, although it was argued that the performance of RF depends only on the number of strong features and not on the number of noisy variables given sample size is large (500 to 1000) (Biau, 2012). This could be attributed to a few factors. The first is because our sample size is relatively small, typical in environmental sciences. The second is that the causal variables are known and included in the model in the simulation by Biau (2012), but the causal variables are either unknown or unavailable in this study. The last is that, in environmental sciences, it could be almost certain that not only the causal variables are often unavailable, but complex interactions are also expected. Therefore, seemingly redundant correlated variables could make some kind of compensation to above factors and contribute to the improvement in predictive accuracy as observed above.

In this study, we adopted a model simplification procedure for RF (Li et al., 2013) that differs fundamentally from the procedures for regression models. Firstly, it simplifies the models based on the variable importance instead of on a significance test. And secondly, the optimal model is chosen based on its predictive accuracy instead of its parsimoniousness. The second feature is particularly important for identifying a predictive model as the predictive accuracy is the ultimate measure for selecting the predictive model. And finally, it is based on the average of multiple iterations, so the results are more stable and thus more reliable.

Inclusion of correlated variables contributed to the increase in predictive accuracy, which is consistent with findings in some previous studies (Li et al., 2012a, Li et al., 2012b). However, this is not supported by other previous studies (Li et al., 2011a, Li et al., 2011b, Li and Sanabria, 2013). These contradictory findings further demonstrate that model selection is crucial and correlated variables should be considered in the model development as they may be able to improve predictive accuracy.

7.2 Cross-validation and predictive errors

Two important findings were observed regarding cross-validation and the performance of predictive methods in this study. The first is that the predictive errors vary over different iterations of the 10-fold cross-validation for each of the methods tested in terms of both RMAE and

RRMSE. This phenomenon has been observed for RF previously (Li et al., 2013). This study confirms that IDW and OK behave similarly as well; and so do the hybrid methods as it may be inherited from either RF, or IDW/OK, or both. This means that predictive accuracy of a method may differ considerably between two 10-fold cross-validations, suggesting that findings derived from two different 10-fold cross-validations may not be comparable even for the same method. This may explain the observations by Genuer et al. (2010), where such differences were attributed to small sample size; it is possible, however, that two different cross-validations were used, which could be the main cause of the differences.

It is apparent that randomness associated with 10-fold cross-validation results in considerably variation in predictive accuracy between iterations. Consequently, the predictive accuracies of various methods based on single cross-validation, and the predictive accuracies of one method based on different cross-validations or on different datasets, are not directly comparable. Therefore, care should be taken when comparing the performance of method(s) derived from different datasets or variables; and the randomness associated with cross-validation needs to be considered in experimental design and in assessing the results of relevant simulation experiments. If comparisons of the predictive accuracies of various methods need to be done based on only one n-fold cross-validation, the dataset should be divided into n sub-datasets as implemented by Li et al. (2010, 2011b). This would make sure that each method is applied to the same datasets, thus ensuring comparable results. If comparisons of the predictive accuracies of various methods need to be performed based on different cross-validations, at least 100 cross-validation iterations are recommended.

The second important finding is that the variation magnitude of predictive errors between iterations changes with methods, indicating that different methods display different responses to individual cross-validations, with IDW showing the biggest variation. This explains why the overall mean of predictive errors of IDW converges more slowly than other methods as more iterations are required to stabilise the mean for IDW. The same applies to RFIDW, because IDW is one of its components. This further suggests that the results of different methods based on single cross-validation with each method being applied to the same datasets are comparable, but they may not be repeatable. This is because such heterogeneity in the variation could lead to different comparative results from different cross-validation runs.

7.3 RF and the hybrid methods

RF, RFOK and RFIDW prove to be the most accurate methods in terms of both RMAE and RRMSE in this study, which support the findings for seabed sediment predictions in previous studies (Li et al., 2011b, Li et al., 2011c, Li et al., 2012b). Overall, RFIDW is preferred. These findings prove that RF and the hybrid methods are not data specific. However, their models are data specific because their predictive errors change with the input secondary variables and an optimal set of predictors need to be selected for the methods for individual primary variables. Therefore, the best model needs to be selected according to individual situations. 'No free lunch theorems' for optimisation (Wolpert and Macready, 1997) are still applicable to RF and the hybrid methods in environmental modelling. The high predictive performance of RF and the hybrid methods should be attributed to their features as discussed in previous studies (Li et al., 2011b, Li et al., 2011c, Li et al., 2011a). They are recommended for spatial prediction in environmental sciences and other relevant disciplines in the future.

7.4 Conclusions

In conclusion, model selection is crucial for RF and the hybrid methods. These methods are not data specific, but their models are. Hence best model needs to be identified for individual studies. Comparison of the predictive accuracies of different methods based on single 10-fold cross-validation, and comparison of the predictive accuracies of the same method based on different cross-validations or on different datasets should be avoided. At least 100 times iterations of cross-validation are recommended for assessing the performance of predictive methods. RFIDW is the most accurate method in this study. Given the high predictive accuracy of the hybrid methods, they are recommended for spatial prediction not only in environmental sciences, but also in other relevant disciplines in the future.

8 Acknowledgements

Peter Tan, Johnathan Kool and Riko Hashimoto provided valuable comments on an earlier draft of this manuscript. Xiaojing Li extracted sediment samples from MARS database. Zhi Huang provided bathymetry, distance to coast, slope and relief data. Chris Lawson produced a map. This paper is published with permission of the Chief Executive Officer, Geoscience Australia.

9 References

ARTHUR, A. D., LI, J., HENRY, S. & CUNNINGHAM, S. A. 2010. Influence of woody vegetation on pollinator densities in oilseed *Brassica* fields in an Australian temperate landscape. *Basic and Applied Ecology,* 11**,** 406-414.

BIAU, G. 2012. Analysis of a random forest method. *Journal of Machine Learning Research,* 13**,** 1063-1095.

CUTLER, D. R., EDWARDS, T. C. J., BEARD, K. H., CUTLER, A., HESS, K. T., GIBSON, J. & LAWLER, J. J. 2007. Random forests for classification in ecology. *Ecography,* 88**,** 2783-2792.

DIAZ-URIARTE, R. & DE ANDRES, S. A. 2006. Gene selection and classification of microarray data using random forest. *BMC Bioinformatics,* 7**,** 1-13.

GENUER, R., POGGI, J. M. & TULEAU-MALOT, C. 2010. Variable selection using random forest. *Pattern Recognition Letters,* 31**,** 2225-2236.

GOOVAERTS, P. 1997. *Geostatistics for Natural Resources Evaluation,* New York, Oxford University Press.

HEAP, A. D. & HARRIS, P. T. 2008. Geomorphology of the Australian margin and adjacent seafloor. *Australian Journal of Earth Sciences,* 55**,** 555-585.

LI, J. 2011. Novel spatial interpolation methods for environmental properties: using point samples of mud content as an example. *The Survey Statistician: The Newsletter of the International Association of Survey Statisticians* No. 63**,** 15-16.

LI, J. & HEAP, A. 2011. A review of comparative studies of spatial interpolation methods in environmental sciences: performance and impact factors. *Ecological Informatics,* 6**,** 228-241.

LI, J., HEAP, A., POTTER, A. & DANIELL, J. J. 2011a. Predicting Seabed Mud Content across the Australian Margin II: Performance of Machine Learning Methods and Their Combination with Ordinary Kriging and Inverse Distance Squared. Geoscience Australia, Record 2011/07, 69pp.

LI, J., HEAP, A. D., POTTER, A. & DANIELL, J. 2011b. Application of machine learning methods to spatial interpolation of environmental variables. *Environmental Modelling & Software,* 26**,** 1647-1659.

LI, J., HEAP, A. D., POTTER, A., HUANG, Z. & DANIELL, J. 2011c. Can we improve the spatial predictions of seabed sediments? A case study of spatial interpolation of mud content across the southwest Australian margin. *Continental Shelf Research,* 31**,** 1365-1376.

LI, J., HEAP, A. D., POTTER, A., HUANG, Z. & DANIELL, J. 2011d. Seabed mud content across the Australian continental EEZ 2011. Geoscience Australia.

LI, J., POTTER, A. & HEAP, A. 2012a. Irrelevant Inputs and Parameter Choices: Do They Matter to Random Forest for Predicting Marine Environmental Variables? *Australian Statistical Conference 2012.* Adelaide.

LI, J., POTTER, A., HUANG, Z., DANIELL, J. J. & HEAP, A. 2010. Predicting Seabed Mud Content across the Australian Margin: Comparison of Statistical and Mathematical Techniques Using a Simulation Experiment. Geoscience Australia, 2010/11, 146pp.

LI, J., POTTER, A., HUANG, Z. & HEAP, A. 2012b. Predicting Seabed Sand Content across the Australian Margin Using Machine Learning and Geostatistical Methods. Geoscience Australia, Record 2012/48, 115pp.

LI, J., SIWABESSY, J., TRAN, M., HUANG, Z. & HEAP, A. 2013. Predicting Seabed Hardness Using Random Forest in R. *In:* ZHAO, Y. & CEN, Y. (eds.) *Data Mining Applications with R.* Elsevier (in press).

MCARTHUR, M. A., BROOKE, B. P., PRZESLAWSKI, R., RYAN, D. A., LUCIEER, V. L., NICHOL, S., MCCALLUM, A. W., MELLIN, C., CRESSWELL, I. D. & RADKE, L. C. 2010. On the use of abiotic surrogates to describe marine benthic biodiversity. *Estuarine, Coastal and Shelf Science,* 88**,** 21-32.

OKUN, O. & PRIISALU, H. Random forest for gene expression based cancer classification: overlooked issues. *In:* MARTÍ, J., BENEDÍ, J. M., MENDONÇA, A. M. & SERRAT, J., eds. Pattern Recognition and Image Analysis: Third Iberian Conference, IbPRIA 2007 June 6-8, 2007 2007 Girona, Spain. Lecture Notes in Computer Science, 4478: 483-490.

PITCHER, C. R., DOHERTY, P. J. & ANDERSON, T. J. 2008. Seabed environments, habitats and biological assemblages. *In:* HUTCHINGS, P., KINGSFORD, M. & HOEGH-GULDBERG, O. (eds.) *The Great Barrier Reef: biology, environment and management.* Collingwood: CSIRO Publishing.

R Development Core Team. 2011. R: A Language and Environment for Statistical Computing. Vienna: R Foundation for Statistical Computing.

R Development Core Team. 2012. R: A Language and Environment for Statistical Computing. Vienna: R Foundation for Statistical Computing.

SHAN, Y., PAULL, D. & MCKAY, R. I. 2006. Machine learning of poorly predictable ecological data. *Ecological Modelling,* 195**,** 129-138.

WHITEWAY, T. 2009. Australian Bathymetry and Topography Grid, June 2009. Geoscience Australia.

WOLPERT, D. & MACREADY, W. 1997. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation,* 1**,** 67-82.

A New Modification of Kohonen Neural Network for VQ and Clustering Problems

Ehsan Mohebi[1] **Adil M. Bagirov**[2]

[1,2]School of Science, Information Technology and Engineering
University of Ballarat, Victoria, 3353, Australia,
Email: [1]e.mohebi@ballarat.edu.au, [2]a.bagirov@ballarat.edu.au

Abstract

Vector Quantization (VQ) and Clustering are significantly important in the field of data mining and pattern recognition. The Self Organizing Maps has been widely used for clustering and topology visualization. The topology of the SOM and its initial neurons play an important role in the convergence of the Kohonen neural network. In this paper, in order to improve the convergence of the SOM we introduce an algorithm based on the split and merging of clusters to initialize neurons. We also introduce a topology based on this initialization to optimize the vector quantization error. Such an approach allows one to find global or near global solution to the vector quantization and consequently clustering problem. The numerical results on 4 small to large real-world data sets are reported to demonstrate the performance of the proposed algorithm.

Keywords: Self Organizing Maps, Clustering, Vector Quantization, Split and Merge Algorithm.

1 Introduction

Clustering is the process of learning concept of raw data by dividing the data into groups of similar objects (Berkhin 2006, Arous 2010). Many clustering algorithms have been proposed based on statistics, machine learning, neural networks and optimization techniques (Jain et al. 1999, Berkhin 2006).

The self organizing map (Kohonen 2001) (SOM) is the well known data mining tool where the aim is to visualize a high dimensional data into usually a 2-Dim grid. The SOM contains a set of neurons that gradually adapts to input data by competitive learning and creates ordered prototypes. The ordered prototypes preserve the topology of the mapped data and make the SOM to be very suitable for cluster analysis (Yang et al. 2012). This adaption is based on a similarity measure, which is usually Euclidean distance, and repositioning of neurons in a 2-Dim space using a learning algorithm. The performance of the SOM strongly depends on a learning algorithm (Haese 1998, Fiannaca et al. 2011, 2007, Goncalves et al. 1998).

Different versions of the SOM have been introduced in (Alahakoon et al. 2000, Appiah et al. 2012, Arous 2010, Ayadi et al. 2012, Brugger et al. 2008, Cheng et al. 2009, Chi & Yang 2006, 2008, Cottrell et al. 2009, Ghaseminezhad & Karami 2011, Gorgonio & Costa 2008, Lapidot et al. 2002, Shah-Hosseini & Safabakhsh 2003, Tasdemir et al. 2011, Vesanto & Alhoniemi 2000, Wong et al. 2006, Xu et al. 2005, Yang et al. 2012, Yen & Wu 2008, Zheng & Greenleaf 1996). The paper (Wong et al. 2006) presents an automated detection algorithm based on the SOM assuming that the training data is adequate representation of the sample distribution. Therefore, the SOM is trained using a small proportion of the sample data set and the algorithm defines a region around prototypes by employing a parameter $r_j,\ j = 1, \ldots, q$ (where q is the number of neurons) that represents the distance of the farthest projected sample into the neuron j. The upcoming samples are distributed into the network and novelties are those samples which cannot fit into these regions. A combinatorial two-stage clustering algorithm based on the SOM is introduced in (Chi & Yang 2008). The numerical results of the enhanced SOM using the Ant Colony Optimization technique and the k-means demonstrates the superiority of the proposed algorithm in comparison with the SOM and k-means. Similarly in (Brugger et al. 2008) an enhanced version of the Clusot algorithm (Bogdan & Rosenstiel 2001) is applied in the SOM for automatic cluster detection. In (Tasdemir et al. 2011) the SOM's prototypes are clustered hierarchically based on the density instead of the distance dissimilarity. Recently, a new two-stage algorithm is proposed in (Yang et al. 2012) that applies the graph cut algorithm (Shi & Malik 2000) to the SOM output. Results presented demonstrate that this algorithm outperforms direct clustering methods using less computational time.

A dynamic SOM is a version of the SOM where its structure is not fixed during the learning phase. In (Alahakoon et al. 2000), a growing self organizing map (GSOM) is presented which defines a spread factor to measure and control the growth of the network. Similarly in (Ayadi et al. 2012), a multi level interior growing SOM is introduced. Unlike the GSOM, which allows the growth only from border sides, this algorithm allows neurons to grow even from an interior node of the map.

Another extension of the SOM algorithm is presented in (Haese 1998). This extension automatically calculates the learning parameters during the training. The algorithm is based on the Kalman filter estimation technique and the idea of the topographic product. The Fast Learning SOM (FLSOM) algorithm is presented in (Fiannaca et al. 2011), which is based on the application of the simulated annealing (SA) metaheuristics to the SOM learning. The SA is used to modify the learning rate factor in an adaptive way. The FLSOM shows a good convergence, better

than the original SOM algorithm.

In all above modifications of the SOM there are no any specific procedures to initialize neurons. Therefore the most of these algorithms are still sensitive to the initialization of neurons. In this paper, to improve the performance of the SOM we propose an initialization algorithm based on the split and merge procedure. The high dense areas in input data space are detected by the proposed split and merge procedure. Then neurons are generated in those detected areas. A new topology is presented for the SOM to restrict the adaption of the neurons to those neighborhood ones which are located in the same high density areas. Such an approach leads to better local minimum of the quantization error than that of by the SOM. The proposed algorithm is tested using eight real-world data sets.

The rest of the paper is organized as follows. The basic self organizing maps and its learning algorithm are presented in Section 3. In Section 2, the split and merge procedure is introduced. The SOM initialization algorithm is presented in Section 3.1. In Section 3.2 a new topology for the SOM is proposed. The modified SOM algorithm and its implementation are discussed in Section 4. Numerical results are presented in Section 5 and Section 6 concludes the paper.

2 Merging and Splitting Algorithms

In this section we discuss splitting and merging procedures in cluster analysis. More specifically, first we present one algorithm for splitting and one algorithm for merging. Finally, we present an algorithm based on the combination of these two algorithms.

Assume that we have a set of k cluster centers $\Lambda = \{c_1, \cdots, c_k\},\ c_i \in \mathbb{R}^n$. These centers are solutions to the following problem:

$$\text{minimize } f = \sum_{i=1}^{k} \sum_{j=1}^{m} \|x_j - c_i\|^2 \text{ where } x_j \in C_i, \quad (1)$$

here c_i is the center point of the set C_i.

In some cases data points from the set C_i are not dense in some neighborhood of its center c_i. Given a radius $\varepsilon > 0$ we consider the following two sets for the cluster C_i:

$$\Phi_c^i(\varepsilon) = \{x_j \in C_i |\ d(x_j, c_i) \leq \varepsilon\}, \quad (2)$$

and

$$\Phi_s^i(\varepsilon) = \{x_j \in C_i |\ \varepsilon < d(x_j, c_i) \leq r_i\},$$

where

$$r_i = \max_j \{d(x_j, c_i) |\ x_j \in C_i\},\ i = 1, \ldots, k.$$

Two clusters C_i and C_l are said to be well separated if $d(c_i, c_l) \geq (r_i + r_l)$.

It is clear that for any cluster $C_i, i = 1, \ldots, k$ there exist $\varepsilon_i \in (0, r_i]$ such that $|\Phi_c^i(\varepsilon)| = \max(|\Phi_c^i(\varepsilon)|, |\Phi_s^i(\varepsilon)|)$ for all $\varepsilon \in (\varepsilon_i, r_i]$. Consider the following equation:

$$\varepsilon_i = \beta r_i.$$

where $\beta \in (0, 1)$. If the ε_i is sufficiently small then data points from the cluster C_i are dense around its center c_i.

The ε_i will be used to design a splitting algorithm for clusters whereas the definition of well separated clusters will be used to design a merging algorithm.

2.1 Splitting

In this subsection we describe the splitting procedure for clusters. This will be done using the parameter β and also special scheme to identify parts of a cluster where most of point reside.

Assume that a set of k clusters, $\Omega = \{C_1, ..., C_k\}$ and a number $\beta \in (0, 1)$ are given. The number of points within the radius $\varepsilon_i = \beta r_i$ from the center of the cluster C_i is:

$$w_c^i = |\Phi_c^i(\varepsilon_i)|.$$

We introduce the angle $\theta_{i,j}$ between the cluster center c_j and the data point $x_i \in C_j$ as follows assuming both $c_j \neq 0$ and $x_i \neq 0$:

$$\theta_{i,j} = \arccos \frac{\langle x_i, c_j \rangle}{\|x_i\| \|c_j\|}. \quad (3)$$

Remark 1 *In order to make (3) well-defined we transform the cluster C_j so that the point $v = (\delta, \ldots, \delta) \in \mathbb{R}^n$ becomes its center. Here $\delta > 0$ is a sufficiently small number, say $\delta \in (0, 0.1]$. It is clear that points x_i from this cluster will be transformed as follows:*

$$\bar{x}_i^t = x_i^t - c_j^t + \delta,\ t = 1, \ldots, n.$$

Moreover we consider only those $\bar{x}_i$ which satisfy the following condition:

$$\varepsilon_j < d(\bar{x}_i, v) \leq r_j.$$

Then the angle $\theta_{i,j}$ is defined between v and $\bar{x}_i$.

Now we introduce the following two sets:

$$\Phi_u^j(\varepsilon_j) = \{x_i \in C_j |\ \varepsilon_j < d(x_i, c_j) \leq r_j,\ 0 \leq \theta_{i,j} \leq \frac{\pi}{2}\}, \quad (4)$$

and

$$\Phi_d^j(\varepsilon_j) = \{x_i \in C_j |\ \varepsilon_j < d(x_i, c_j) \leq r_j,\ \frac{\pi}{2} \leq \theta_{i,j} \leq \pi\}. \quad (5)$$

The cardinalities of these sets are $w_u^j = |\Phi_u^j(\varepsilon_j)|$ and $w_d^j = |\Phi_d^j(\varepsilon_j)|$, respectively.

The sets $\Phi_c^i(\varepsilon_i), \Phi_u^j(\varepsilon_j)$ and $\Phi_d^j(\varepsilon_j)$ satisfy the following conditions:

1. $w_u^j + w_d^j + w_c^j = |C_j|$;
2. $\Phi_c^j(\varepsilon_i) \cup \Phi_u^j(\varepsilon_j) \cup \Phi_d^j(\varepsilon_j) = C_j$;
3. $\Phi_c^j(\varepsilon_i) \cap \Phi_u^j(\varepsilon_j) = \emptyset$, $\Phi_c^j(\varepsilon_i) \cap \Phi_d^j(\varepsilon_j) = \emptyset$, $\Phi_u^j(\varepsilon_j) \cap \Phi_d^j(\varepsilon_j) = \emptyset$.

Application of the splitting procedure to the cluster C_j depends on the values of w_u^j, w_d^j and w_c^j. If

$$w_c^j \geq \max\{w_d^j, w_c^j\} \quad (6)$$

then data points are dense around the cluster center and we do not split such a cluster. If

$$w_c^j < \max\{w_d^j, w_c^j\} \quad (7)$$

then we split this cluster into two new ones. In order to do so we define the following two subsets of $\Phi_c^i(\varepsilon_i)$:

$$\Phi_{cu}^j(\varepsilon_j) = \left\{x_i \in \Phi_c^i(\varepsilon_i) | d(x_i, c_j) \leq \varepsilon_j,\ 0 \leq \theta_{i,j} \leq \frac{\pi}{2}\right\}, \quad (8)$$

and

$$\Phi^j_{cd}(\varepsilon_j) = \left\{ x_i \in \Phi^i_c(\varepsilon_i) | d(x_i, c_j) \leq \varepsilon_j,\ \frac{\pi}{2} \leq \theta_{i,j} \leq \pi \right\}. \tag{9}$$

Then the cluster C_j is split into two new clusters as follows:

$$C^*_j = \{\Phi^j_u(\varepsilon_j) \cup \Phi^j_{cu}(\varepsilon_j)\}, \tag{10}$$

with the center

$$c^*_j = \frac{1}{|C^*_j|} \sum_{x_i \in C^*_j} x_i, \tag{11}$$

and

$$C^*_{j'} = \{\Phi^j_d(\varepsilon_j) \cup \Phi^j_{cd}(\varepsilon_j)\}. \tag{12}$$

with the center

$$c^*_{j'} = \frac{1}{|C^*_{j'}|} \sum_{x_i \in C^*_{j'}} x_i. \tag{13}$$

Thus, the splitting algorithm can be summarized as follows:

Algorithm 1 Splitting algorithm
Step 0. Input: A collection of k clusters $\Omega = \{C_1, ..., C_k\}$, and the ratio $\beta \in (0, 1)$.
Step 1. Select cluster $C_j \in \Omega$ and calculate its center c_j.
Step 2. Calculate $d(x_i, c_j)$ and also $\theta_{i,j}$ using (3) for all data point $x_i \in C_j$.
Step 3. For each cluster C_j calculate sets $\Phi^j_c(\varepsilon_i), \Phi^j_u(\varepsilon_j), \Phi^j_d(\varepsilon_j)$ using (2), (4) and (5), respectively.
Step 4. If (6) is satisfied then go to Step 6, otherwise go to Step 5.
Step 5. Split the cluster C_j into two new clusters C^*_j and $C^*_{j'}$ using (10) and (12), respectively. Update Ω and set $k := k + 1$.
Step 6. If all clusters $C_j,\ j = 1, \ldots, k$ are visited terminate, otherwise go to Step 2.

2.2 Merging

Assume that the collection of k clusters, $\Omega = \{C_1, ..., C_k\}$, is given. It may happen that (also after applying the splitting algorithm) some clusters are not well separated. In this subsection we design an algorithm to merge clusters which are well separated from each other.

According to the definition of well separated clusters two clusters $C_j, C_p \in \Omega$ should be merged if

$$d(c_j, c_p) - (r_j + r_p) < 0. \tag{14}$$

These two clusters are merged into one cluster as follows:

$$C^*_j = C_j \cup C_p, \tag{15}$$

with the center

$$c^*_j = \frac{1}{|C^*_j|} \sum_{x_i \in C^*_j} x_i. \tag{16}$$

For the cluster C^*_j we use only the index j meaning that the cluster C_p joins the cluster C_j. Then the merging algorithm can be summarized as follows:

Algorithm 2 Merging algorithm
Step 0. Input: A collection of k clusters $\Omega = \{C_1, ..., C_k\}$.
Step 1. Select cluster $C_j \in \Omega$ and calculate its center c_j.
Step 2. Select cluster $C_p \in \Omega$ and calculate its center c_p, where $j \neq p$.
Step 3. If the condition (14) is satisfied then go to Step 4, otherwise go to Step 6.
Step 4. Merge clusters C_j and C_p using (15). Update the set Ω and set $k := k - 1$.
Step 5. If all cluster $C_p,\ p = 1, \cdots, k$ and $j \neq p$ are visited go to Step 6, otherwise go to Step 2.
Step 6. If all clusters $C_j \in \Omega$ are visited terminate, otherwise go to Step 1.

One can see that Algorithm 1 and 2 are complementary. In other words, to have stable cluster centers these algorithms should be applied iteratively until the cluster centers become stable. The stability can be checked by monitoring the value of (1) until satisfying the strictly decreasing value or the maximum number of iteration can be predefined in advance. In this paper we use the second criterion and the split and merge algorithm is presented as follows.

Algorithm 3 Split and Merge algorithm
Step 0. Input: A collection of k clusters $\Omega = \{C_1, ..., C_k\}$, the maximum number of iterations $\gamma_{max} > 0$ and the ratio $\beta \in (0, 1)$. Set $i := 0$.
Step 1. Set $i := i + 1$.
Step 2. Apply Algorithm 1 to the collection of clusters Ω. This algorithm will generate a new collection of clusters Ω.
Step 3. Apply Algorithm 2 to the collection of clusters Ω.
Step 4. If $i > \gamma_{max}$ terminate, otherwise go to Step 1.

3 Self Organizing Maps

The SOM is an unsupervised neural network (Kohonen 2001) that usually contains a 2-Dim array of neurons $\Psi = \{w_1, \cdots, w_q\}$. Assume that we are given the set of m input data vectors $A = \{x_1, \cdots, x_m\}$ where $x_i \in \mathbb{R}^n,\ i = 1, \cdots, m$. In the SOM a weight $w_j \in \mathbb{R}^n$ is associated with the neuron $j,\ j = 1, \cdots, q$. For given $j \in \{1, \ldots, q\}$ define the following set:

$$S_j = \{x_k : d(x_k, w_j) < d(x_k, w_l),\ l \neq j,\ l = 1, \ldots, q\} \tag{17}$$

where

$$d(x, y) = \|x - y\| = \left(\sum_{t=1}^{n} (x^t - y^t)^2 \right)^{1/2},\ x, y \in \mathbb{R}^n$$

is the Euclidean distance.

One data point $x_i,\ i \in \{1, \ldots, m\}$ at a time is presented to the network and is compared with all weight vectors. The nearest $w_j,\ j = 1, \cdots, q$ is selected as the *best matching unit* (BMU) for the i-th data point. This data point is mapped to the best matching neuron. Therefore,

$$S_j = S_j \cup x_i.$$

The set of neighborhood weights $N_c = \{w_l : p(c, l) \leq r,\ l \neq c\}$ around the BMU are updated where $p(c, l)$ is the distance between the BMU and the neighborhood neuron l in 2-Dim coordinates of

the network topology and r is the predefined radius. Furthermore, $p(c,l) \in \mathbb{N}$ and $0 < p(c,l) \leq r$. The aim in this paper is to solve the following problem:

$$\text{minimize} \quad E = \frac{1}{m}\sum_{i=1}^{m} \|x_i - w_c\|, \tag{18}$$

where w_c is the weight of the BMU of x_i, $i = 1, \ldots, m$. A general description of the SOM algorithm is as follows.

Algorithm 4 SOM algorithm
Step 1. Initialize the dimension of the network, the maximum number of iterations (T), a radius (r) of the network and weight vectors $w_j, j = 1, \ldots, q$. Set iteration counter $\tau := 0$.
Step 2. Select data $x_i, i = 1, \ldots, m$ and find its closest neuron c, that is

$$c := \underset{j=1,\ldots,q}{\operatorname{argmin}} \|x_i - w_j\|. \tag{19}$$

Step 3. Update the set of neighborhood neurons $w_j \in N_c$ using the following equation:

$$w_j := w_j + \alpha(\tau)h(\tau)(x_i - w_j). \tag{20}$$

(Here h is a neighborhood function and $\alpha(\tau)$ is a learning rate at the iteration τ.)
Step 4. If all input data are presented to the network go to Step 5, otherwise go to Step 2.
Step 5. Calculate E_τ using (18). If $\tau > T$ terminate, otherwise set $\tau := \tau + 1$ and go to Step 2.

The neighborhood function h in Step 3 of Algorithm 4 plays an important role in the SOM. Usually h is a decreasing exponential function of τ. The learning rate α is a decreasing linear function of τ and σ reduces the width of the neighborhood function h as $\tau \to T$.

3.1 SOM Initialization Algorithm

Usually the set of SOM neurons $\Psi = \{w_1, \cdots, w_q\}$ are initialized randomly (Kohonen 2001). This leads the network to converge only to local solutions of Problem (18). Furthermore the SOM suffers from slow convergence. In other words, the number of iterations to learn the input data become large and the neurons may not learn some data points correctly. In this section we present a new algorithm based on Algorithm 1 and 2 to initialize the neurons of the SOM and then define a modified topology of neurons at the initial points.

Algorithm 5 SOM initialization algorithm
Step 0 (Initialization). A set of m input data vectors $A = \{x_1, \cdots, x_m\}$. Set $\Psi := \emptyset$.
Step 1. Calculate the center c^* of the set A, set $w_1 := c^*$ and

$$\Psi := \Psi \cup w_1.$$

Step 2. Apply Algorithm 3 on Ψ. This algorithm will generate a new set Ψ of neurons.
Step 3. Set the final Ψ as initial neurons of the SOM.

Algorithm 5 ensures that the initial neurons are located in distinct high density area of the input data space which is found by Algorithm 1. Algorithm 2 guarantees that initial neurons are not close to each other.

3.2 SOM with Modified Topology

We have the set of initial neurons $\Psi = \{w_1, \cdots, w_{\hat{q}}\}$ applying the SOM initialization algorithm. We generate a set of e number of neurons $u_z \in \mathbb{R}^n$, $z = 1, \ldots, e$ using each individual neuron w_i, $i = 1, \cdots, \hat{q}$ as follows:

$$g_i = \{u_z | w_i^t - \lambda\varepsilon_i \leq u_z^t \leq w_i^t + \lambda\varepsilon_i\}, \tag{21}$$

where $t = 1, \ldots, n$, $z = 1, \ldots, e$ and $\lambda \in \mathbb{R}$, $\lambda > 1$. One can see that all the neurons in the set g_i are close to w_i to cover up the dense area which is centered by neuron w_i. The use of such neurons allows to decrease the quantization error (18). In Problem (18) the local solution is obtained while none of the activated neurons are far from its mapped data points. Therefore, the set of neurons defined by (21) guarantees that the SOM learning process escapes from local solutions of Problem (18) and converges to the global ones.

Usually in the SOM topology, all neighborhood neurons are connected to each other in order to spread the adaption to adjacent neurons. For each pair $w_i, w_j, i, j = 1, \ldots, \hat{q}$, $i \neq j$ we define the following integer number:

$$\hat{r}_{ij} = \left\lceil \frac{d(w_i, w_j)}{\varepsilon_i + \varepsilon_j} \right\rceil. \tag{22}$$

Here $\lceil x \rceil$ is a smallest integer greater than or equal to x, called its ceiling. Note that the parameter $\hat{r}_{ij}$ for neurons $u_i, u_j \in g_k$, $i, j = 1, \ldots, e$, $i \neq j$, $k = 1, \cdots, \hat{q}$ is set to 1. In order to determine the connectivity of neurons we define the threshold $r_0 \geq 1$ and say that two neurons w_i, w_j are connected if $\hat{r}_{ij} \leq r_0$. The threshold r_0 is defined for the whole data set. The neurons in the sets g_i, $i = 1, \cdots, \bar{q}$ are connected to their parent neuron w_i and to each other as well. Then we have the following connectivity matrix for the new topology:

1. $con(i,j) \in \{0,1\}$, $w_i, w_j \in \Psi$.
2. $con(i,j) \in \{0\}$, $u_i \in g_k$, $u_j \in g_p$, $k \neq p$.
3. $con(i,j) \in \{1\}$, $u_i \in g_k$, $u_j \in g_p$, $k = p$.
4. $con(i,j) \in \{1\}$, $u_i \in g_k$, $w_j \in \Psi$, $k = j$.
5. $con(i,j) \in \{0\}$, $u_i \in g_k$, $w_j \in \Psi$, $k \neq j$.

This new topology guarantees that neurons from one dense area are not connected with those from another dense area and therefore according to the equation (20) such neurons do not change each others weight.

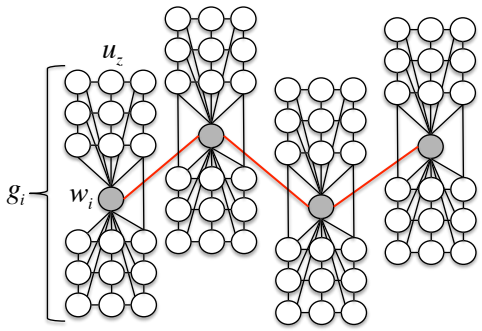


Figure 1: Topology of modified SOM.

In Figure 1 the initial neurons are in gray and the generated neurons around each initial neuron are in white color. There is no any connection between two separate set of generated neurons.

Algorithm 6 SOM topology generation
Step 0. Given: A set of $\Psi = \{w_1, \cdots, w_{\bar{q}}\}$ of initial neurons and a number $\lambda > 1$.
Step 1. Select a w_i and generate g_i using equation (21).
Step 2. Connect w_i all $u_z \in g_i$.
Step 3. If all $w_i \in \Psi$ are visited go to Step 4, otherwise go to Step 1.
Step 4. Select a w_i and connect it to all $w_j \in \Psi$ with $\hat{r}_{ij} < \bar{r}$.
Step 5. If all $w_i \in \Psi$ are visited terminate, otherwise go to Step 4.

4 The Modified SOM Algorithm and Its Implementation

In this section, we modify Algorithm 4 applying new initialization algorithm for neurons and modified SOM topology. The new algorithm can be summarized as follows.

Algorithm 7 Modified SOM algorithm
Step 0. (Initialization) Initialize the maximum number of iterations T of the network. Set the maximum number of iterations in the Splitting and Merging algorithm as γ_{max} and the value of the ratio $\beta > 0$. Set the initial value of iteration and ratio to γ_0. Set the step length β_s and set the iteration counter $\tau := 0$. A set of m input data vectors $A = \{x_1, \cdots, x_m\}$.
Step 1. (Split and Merge). Apply Algorithm 5 to A for $\gamma_0 \to \gamma_{max}$ and $\beta_0 \to \beta_{max}$ to generate the set of $\Psi = \{w_1, \cdots, w_{\hat{q}}\}$ which minimizes the function f in (1). This set is initial weights of neurons.
Step 2. (SOM topology). Apply Algorithm 6 to the set Ψ.
Step 3. Select data $x_i, i = 1, \ldots, m$ and find its closest neuron $w_c \in \{\Psi \bigcup_{i=1}^{\hat{q}} g_i\}$, that is

$$c := \underset{j=1,\ldots,(\hat{q}\cdot e)}{\operatorname{argmin}} \|x_i - w_j\|. \tag{23}$$

Step 4. Update the set of neighborhood neurons $w_j \in N_c$ using the following equation:

$$w_j := w_j + \alpha(\tau)h(\tau)(x_i - w_j), \tag{24}$$

where

$$N_c = \begin{cases} g_i \cup w_i & \text{if } w_c = u_z \in g_i, \\ g_i \cup w_i \cup \Xi & \text{if } w_c = w_i,\ w_i, w_j \in \Psi, \end{cases}$$

subject to

$$\Xi = \{w_j | d(w_i, w_j) < r',\ i \neq j\}.$$

Step 5. If all input data are presented to the network go to Step 6, otherwise go to Step 3.
Step 6. Calculate E_τ using (18). If $\tau > T$ terminate, otherwise set $\tau := \tau + 1$ and go to Step 3.

Note that the neighborhood function in equation (24) of Algorithm 7 is as follows.

$$h(\tau) = \exp\left(-\frac{\bar{r}^2}{2\sigma(\tau)^2}\right), \tag{25}$$

subject to

$$\sigma(\tau) = \eta\frac{T - \tau}{\tau},\ \eta \in \mathbb{R}, \tag{26}$$

and usually $\eta \geq 1$.

One can see that the Step 3 to 6 in Algorithm 7 is similar to the basic SOM. The only exception is in Step 4 where the set N_c is defined in order to improve the approximation of the global solution to the vector quantization problem.

4.1 Implementation of Algorithm 7

In Algorithm 4, weight vectors $w_j, j = 1, \cdots, q$ are initialized randomly. The maximum number of iterations T is set between 20 and 40 for small to large data set, respectively. Although, for large data sets more time is required to obtain stable network over input data. The topology of SOM network is rectangular (Kohonen 2001) with same number of neurons in each column and row (i.e. $n \times n$). Each interior neuron is connected with 8 neighborhood neurons, however this number is less than 5 for border neurons. Furthermore, the radius of map r is set to 2 for small and 4 for large number of neurons (see Table 1).

Table 1: Initialization of SOM parameters in Algorithm 4.

Data sets	Input Size	SOM Dim.	r	T
Small	$(\|A\| < 10^3)$	10×10	2	20
Medium	$(10^3 < \|A\| < 10^4)$	15×15	3	30
Large	$(10^4 < \|A\| < 0.5 \cdot 10^5)$	20×20	4	40
	$(\|A\| > 0.5 \cdot 10^5)$	25×25	3	20

As it is presented in Table 1, the number of neurons, maximum iteration number T and r are chosen incrementally in order to be applicable to larger input data sets. The exception is for the data set with size $|A| > 0.5 \cdot 10^5$, where r and T are smaller comparing to other large data sets to decrease the computational complexity.

In Step 1 of Algorithm 7, we set values of T same as in Table 1, $2 \leq \gamma \leq 6$ and $0.05 \leq \beta \leq 0.6$ with step length β_s to 0.05 for Algorithm 3. In Step 3, the parameter λ in Algorithm 6 using equation (21) is set to 1.5 for small and medium size data sets whereas this value is set to 2.5 for large ones. In Step 4, the Algorithm 6 generates 9 neurons ($|g_i|$), i.e. e is set to 9, around all neurons $w_i \in \Psi$. Therefore the total number of neurons is $|\Psi| \times |g_i| + |\Psi|$. It should be noted that for all datasets the parameter $\bar{r}$ in (22) is set to 3. Finally, the parameter η in (26) is set to 1 for all data sets.

5 Numerical Results

To demonstrate the effectiveness of the proposed algorithm, numerical experiments were carried out using a number of real-world data sets. Algorithm 7 was coded in NetBeans IDE under Java platform and tested on a MAC OSX with 2.7GHz core i7 CPU and 10GB of RAM. 4 data sets, one small (Iris), one medium size (Image Segmentation), one large (Gamma Telescope) and one very large (NE) were used in experiments. A brief description of data sets is presented in Table 2, more details can be found in (Bache & Lichman 2013, Theodoridis 1996, Reinelt 1991).

Table 2: Brief description of the data sets

Data sets	Number of instances	Number of attributes
Fisher's Iris Plant	150	4
Image Segmentation	2310	19
Gamma Telescope	19020	10
NE	50000	2

The results obtained by the Split and Merge algorithm, which is in Step 2 of Algorithm 7 is presented in Table 3. To define the improvement E_{im} obtained

by the proposed algorithm, we use the following formula:

$$E_{im} = \frac{E_{SOM} - E}{E_{SOM}} \cdot 100\%. \quad (27)$$

where E is the value obtained by the Modified SOM.

Table 3: The results of the Split and Merge algorithm

Data sets	β^*	γ^*	$\lvert\Psi\rvert$	f_{min}	t
Fisher's Iris Plant	0.05	2	23	4.95×10^1	0.02
Image Segmentation	0.10	2	62	3.17×10^7	0.14
Gamma Telescope	0.05	2	87	2.01×10^8	6.61
NE	0.20	2	61	3.66×10^2	4.57

The values of quantization error using equation (18) for different iterations and different data sets are presented in Tables 4. From these results one can see that the Modified SOM outperforms SOM in all data sets. The Modified SOM reduced the value of problem (18) significantly in Image Segmentation and Iris data sets up to 38.28% and 24.82%, respectively. On other data sets the improvement E_{im} is between 6.85% and 14.11%. One can see that the Modified SOM starts with a small value of E comparing to the SOM. This is due to optimized initialization of the Modified SOM algorithm. The computational effort used by the Modified SOM is much less than that of the SOM in all data sets. The Split and Merge algorithm that initializes the Modified SOM is very efficient and it is not time consuming. The new initialization algorithm which is based on the Split and Merge algorithm speeds up the convergence of the Modified SOM and makes it less time consuming than the SOM. The maximum time reduction by the Modified SOM was achieved in Gamma Telescope data sets. On the other hand the minimum computational time reduction is on very large data set: NE data set.

In Figure 2 the values of E obtained by the Modified SOM is compared with those obtained by the SOM on Iris data set. One can see that on Iris data set the Modified SOM starts with a value of E close to global one and converge to the optimal value within the given number of iterations. Since the SOM is initialized randomly, it takes more time to converge. In Figure 3 the CPU time required by the SOM and Modified SOM on Gamma Telescope is presented. One can see that the Modified SOM requires more CPU time at the early iterations due to running the Split and Merge algorithm for initialization. Once the Modified SOM initialized, the convergence is much faster than the SOM which is initialized randomly.

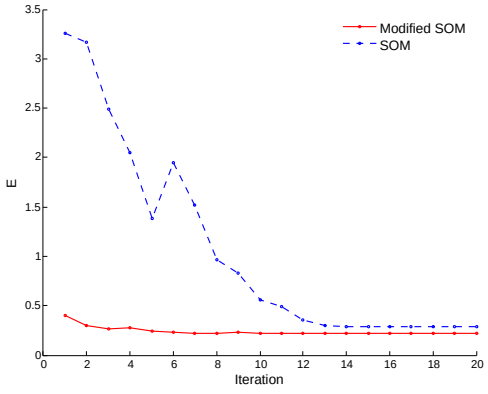


Figure 2: SOM vs Modified SOM using E values (Iris dataset).

Table 4: Results for all data sets

iter	E	t	E	t
	Iris			
	SOM		Modified SOM	
2	3.17E+00	0.06	2.96E-01	0.02
4	2.05E+00	0.08	2.76E-01	0.03
6	1.95E+00	0.09	2.29E-01	0.03
8	9.70E-01	0.09	2.23E-01	0.05
10	5.56E-01	0.11	2.22E-01	0.05
12	3.51E-01	0.12	2.22E-01	0.05
14	2.88E-01	0.12	2.22E-01	0.06
16	2.86E-01	0.14	2.22E-01	0.06
20	2.86E-01	0.16	2.15E-01	0.08
	Image Segmentation			
	SOM		Modified SOM	
2	1.82E+07	0.73	1.01E+02	0.40
4	2.41E+03	1.28	8.01E+01	0.62
6	1.84E+02	1.75	2.90E+01	0.84
10	1.42E+02	2.74	1.89E+01	1.26
14	1.02E+02	3.65	1.75E+01	1.68
18	4.55E+01	4.54	1.74E+01	2.09
22	2.69E+01	5.40	1.74E+01	2.51
25	2.69E+01	6.04	1.75E+01	2.84
30	2.69E+01	7.00	1.66E+01	3.35
	Gamma Telescope			
	SOM		Modified SOM	
2	3.93E+20	10.97	8.28E+01	8.39
4	5.11E+12	21.32	4.08E+01	10.00
6	2.22E+03	31.73	3.46E+01	11.62
10	2.21E+02	53.17	3.15E+01	14.84
18	1.30E+02	95.26	3.02E+01	21.23
22	7.56E+01	116.35	3.00E+01	24.43
26	4.53E+01	137.56	2.99E+01	27.63
30	3.34E+01	158.70	2.99E+01	30.81
40	3.33E+01	210.52	2.86E+01	38.44
	NE			
	SOM		Modified SOM	
2	2.02E+07	5.18	3.28E+07	6.47
4	3.32E-01	9.86	8.56E+32	8.18
6	3.01E-01	14.56	3.88E-02	9.89
8	2.70E-01	19.25	2.15E-02	11.54
10	2.56E-01	23.93	1.23E-02	13.20
12	1.94E-01	28.52	1.12E-02	14.88
14	5.41E-02	32.99	1.12E-02	16.55
16	1.16E-02	37.46	1.12E-02	18.24
20	1.12E-02	46.30	1.03E-02	21.51

Note that the error E shows the quantization quality of the network. However, there is a distortion measurement which can be used to calculate the overall quality of the map. Unlike the quantization error, the distortion measure ξ considers both vector quantization and topology preservation of the SOM. The distortion measure is defined as follows (Arous 2010, Ayadi et al. 2012):

$$\xi = \sum_{x_i \in A} \sum_{w_j \in \Psi} h_{cj} \|x_i - w_j\|^2, \quad j \neq c, \quad (28)$$

where c is the BMU of x_i and h_{cj} is the neighborhood function of neurons c and j defined by Equation (25).

Table 5 presents the distortion measure (28) and number of active neurons n_{act} for all data sets. One can see that the distortion error ξ obtained by Modified SOM is less than that obtained by the SOM in all data sets. This is due to the topology of the Modified SOM where the neurons from different dense areas are not connected. This prevents deterioration of the network from its optimal value of ξ and E simultaneously.

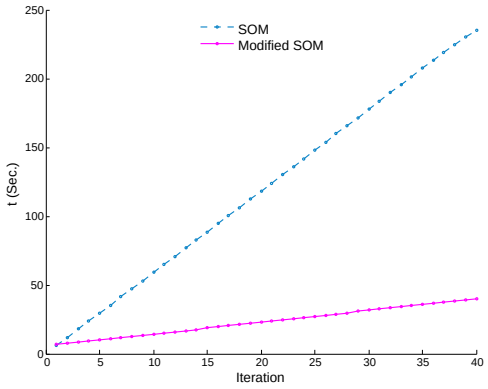


Figure 3: SOM vs Modified SOM using CPU time (Gamma Telescope dataset).

Table 5: Results of distortion measure on all data sets

	SOM		Modified SOM	
Dataset	ξ	n_{act}	ξ	n_{act}
Iris	1.25×10^{-6}	69	3.62×10^{-7}	92
Image Seg.	3.26×10^{-4}	210	8.73×10^{-5}	490
Gamma Telescope	2.35×10^{-5}	400	1.39×10^{-5}	759
NE	1.66×10^{-1}	375	3.11×10^{-2}	610

6 Conclusion

The aim in this paper was to propose an initialization algorithm and a new topology for the Modified Self Organizing Maps which restrict the neighborhood adaptations to only those neurons that are not in different dense areas. We introduced the Split and Merge algorithm to generate such neurons. This algorithm is a part of the initialization algorithm in the Modified SOM and the numerical experiments show that the initialization algorithm generates neurons close to the optimal solution. Consequently we presented a topology for the SOM to generate neurons in high dense areas of input data space and do not connect neurons from different dense areas. The experiments show that this restriction reduces the quantization and distortion errors. Numerical results demonstrate the superiority of the proposed algorithm over the SOM in the sense of accuracy. These results also show that the Modified SOM converges much faster than the SOM and in all cases the proposed algorithm requires less computational time.

References

Alahakoon, D., Halgamuge, S. K. & Srinivasan, B. (2000), 'Dynamic self-organizing maps with controlled growth for knowledge discovery', *IEEE transactions on neural networks* **11**(3), 601–14.

Appiah, K., Hunter, A., Dickinson, P. & Meng, H. (2012), 'Implementation and Applications of Tri-State Self-Organizing Maps on FPGA', *IEEE Transactions on Circuits and Systems for Video Technology* **22**(8), 1150–1160.

Arous, N. (2010), 'On the Search of Organization Measures for a Kohonen Map Case Study: Speech Signal Recognition', *International Journal of Digital Content Technology and its Applications* **4**(3), 75–84.

Ayadi, T., Hamdani, T. & Alimi, A. (2012), 'Migsom: Multilevel interior growing self-organizing maps for high dimensional data clustering', *Neural Processing Letters* **36**, 235–256.

Bache, K. & Lichman, M. (2013), 'UCI machine learning repository', http://archive.ics.uci.edu/ml.

Berkhin, P. (2006), 'A survey of clustering data mining techniques', *Grouping Multidimensional Data* pp. 1–56.

Bogdan, M. & Rosenstiel, W. (2001), Detection of cluster in self-organizing maps for controlling a prostheses using nerve signals, *in* 'ESANN'01', pp. 131–136.

Brugger, D., Bogdan, M. & Rosenstiel, W. (2008), 'Automatic cluster detection in Kohonen's SOM', *IEEE transactions on neural networks* **19**(3), 442–59.

Cheng, S.-S., Fu, H.-C. & Wang, H.-M. (2009), 'Model-based clustering by probabilistic self-organizing maps', *IEEE transactions on neural networks* **20**(5), 805–26.

Chi, S.-C. & Yang, C. C. (2006), Integration of ant colony som and k-means for clustering analysis, *in* 'Proceedings of the 10th international conference on Knowledge-Based Intelligent Information and Engineering Systems - Volume Part I', KES'06, Springer-Verlag, Berlin, Heidelberg, pp. 1–8.

Chi, S.-C. & Yang, C.-C. (2008), 'A Two-stage Clustering Method Combining Ant Colony SOM and K-means', *Journal of Information Science and Engineering* **24**(5), 1445–1460.

Cottrell, M., Gaubert, P., Eloy, C., Francois, D., Hallaux, G., Lacaille, J. & Verleysen, M. (2009), 'Fault Prediction in Aircraft Engines Using Self-Organizing Maps', *Security* pp. 37–44.

Fiannaca, A., Di Fatta, G., Rizzo, R., Urso, A. & Gaglio, S. (2011), 'Simulated annealing technique for fast learning of som networks', *Neural Computing and Applications* pp. 1–11.

Fiannaca, A., Fatta, G., Gaglio, S., Rizzo, R. & Urso, A. (2007), Improved som learning using simulated annealing, *in* 'Artificial Neural Networks AI ICANN 2007', Vol. 4668 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 279–288.

Ghaseminezhad, M. H. & Karami, A. (2011), 'A novel self-organizing map (SOM) neural network for discrete groups of data clustering', *Applied Soft Computing* **11**(4), 3771–3778.

Goncalves, M. L., de Andrade Netto, M. L. & Zullo, J. (1998), A neural architecture for the classification of remote sensing imagery with advanced learning algorithms, *in* 'Neural Networks for Signal Processing VIII, 1998. Proceedings of the 1998 IEEE Signal Processing Society Workshop', pp. 577–586.

Gorgonio, F. L. & Costa, J. A. F. (2008), Combining Parallel Self-Organizing Maps and K-Means to Cluster Distributed Data, *in* '2008 11th IEEE International Conference on Computational Science and Engineering - Workshops', IEEE, pp. 53–58.

Haese, K. (1998), 'Self-organizing feature maps with self-adjusting learning parameters', *IEEE transactions on neural networks* **9**(6), 1270–8.

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

Kohonen, T. (2001), *Self-Organizing Maps*, Springer Series in Information Sciences, Springer.

Lapidot, I., Guterman, H. & Cohen, a. (2002), 'Unsupervised speaker recognition based on competition between self-organizing maps', *IEEE transactions on neural networks* **13**(4), 877–87.

Reinelt, G. (1991), 'TSPLIB- a Traveling Salesman Problem Library', *ORSA Journal of Computing* **3**(4), 376–384.

Shah-Hosseini, H. & Safabakhsh, R. (2003), 'TASOM: a new time adaptive self-organizing map', *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society* **33**(2), 271–82.

Shi, J. & Malik, J. (2000), 'Normalized cuts and image segmentation', *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(8), 888–905.

Tasdemir, K., Milenov, P. & Tapsall, B. (2011), 'Topology-Based Hierarchical Clustering of Self-Organizing Maps', *IEEE Transactions on Neural Networks* **22**(3), 474–485.

Theodoridis, Y. (1996), 'Spatial datasetsan unofficial collection', http:// www.dias.cti.gr/ ~ytheod/research/ datasets/ spatial.html. Accessed 2013.

Vesanto, J. & Alhoniemi, R. (2000), 'Clustering of the self-organizing map', *IEEE Transactions on Neural Networks* **11**(3), 587–600.

Wong, M., Jack, L. & a.K. Nandi (2006), 'Modified self-organising map for automated novelty detection applied to vibration signal monitoring', *Mechanical Systems and Signal Processing* **20**(3), 593–610.

Xu, P., Chang, C.-H. & Paplinski, A. (2005), 'Self-organizing topological tree for online vector quantization and data clustering.', *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society* **35**(3), 515–26.

Yang, L., Ouyang, Z. & Shi, Y. (2012), 'A Modified Clustering Method Based on Self-Organizing Maps and Its Applications', *Procedia Computer Science* **9**, 1371–1379.

Yen, G. G. & Wu, Z. (2008), 'Ranked centroid projection: a data visualization approach with self-organizing maps', *IEEE transactions on neural networks* **19**(2), 245–59.

Zheng, Y. & Greenleaf, J. F. (1996), 'The Effect of Concave and Convex Weight Adjustments on Self-organizing Maps', *IEEE transactions on neural networks* **7**(1), 87–96.

Sentiment Augmented Bayesian Network

Sylvester Olubolu Orimaye

School of Information Technology, MONASH University Malaysia
sylvester.orimaye@monash.edu

Abstract

Sentiment Classification has recently gained attention in the literature with different machine learning techniques performing moderately. However, the challenges that sentiment classification constitutes require a more effective approach for better results. In this study, we propose a logical approach that augments the popular Bayesian Network for a more effective sentiment classification task. We emphasize on creating dependency networks with quality variables by using a sentiment-dependent scoring technique that penalizes the existing Bayesian Network scoring functions such as K2, BDeu, Entropy and MDL. The outcome of this technique is called Sentiment Augmented Bayesian Network. Empirical results on three product review datasets from different domains, suggest that a sentiment-augmented scoring mechanism for Bayesian Network classifier, has comparable performance, and in some cases outperform state-of-the-art sentiment classifiers.

Keywords: sentiment; classification; Bayesian network

1 Introduction

Sentiment Classification (SC) has recently gained a lot of attention in the research community. This is due to its increasing demand for the analysis of consumer sentiments on products, topic and news related text from social media such as Twitter[1] and online product reviews such as Amazon[2]. In the same manner, Bayesian Network (BN)(Cooper & Herskovits 1992) also known as Bayesian Belief Network plays a major role in Machine Learning (ML) research for solving classification problems. Over the last decade, learning BNs has become an increasingly active area of ML research where the goal is to learn a network structure using dependence or independence information between set of variables (Cooper & Herskovits 1992, Friedman et al. 1997, Cheng & Greiner 2001, Chen et al. 2008). The resulting network is a directed acyclic graph (DAG), with a set of joint probability distributions, where each variable of the network is a node in the graph and the arcs between the nodes represent the probability distribution that signifies the level of dependency between the nodes.

While it is more common to use other ML algorithms such as Support Vectors Machines (SVM), Naïve Bayes (NB) and Maximum Entropy (ME) for SC tasks (Pang & Lee 2004, Boiy & Moens 2009), few research papers have proposed BN as a competitive alternative to other popular ML algorithms. Considering the huge amount of data available from social media and the level of difficulty attached with analysing sentiments from natural language texts, the ability of BN to learn dependencies between words and their corresponding sentiment classes, could undoubtedly produce a better classifier for the sentiment classification task. This paper focusses on constructing a BN classifier that uses sentiment information as one important factor for determining dependency between network variables.

BN has been successfully applied to solve different ML problems with its performance outweighing some of the popular ML algorithms. For example, in Su & Zhang (2006), a full Bayesian Network classifier (FBC) showed statistically significant improvement on state-of-the-art ML algorithms such as SVM-SMO, C4.5 and NB on 33 UCI datasets. In the case of SC, NB, which is a special case of BN (Cheng & Greiner 1999), and one of the leading ML algorithms for SC tasks (Pang & Lee 2004), has surprisingly and repeatedly shown improved performance on movie and product reviews despite its conditional independence assumption. By comparative study, we show that a Sentiment Augment Bayesian Network (SABN) has better or comparable performance with NB and SVM classifiers on popular review datasets such as Amazon product reviews.

Constructing a BN classifier requires learning a network structure with set of Conditional Probability Tables (CPTs)(Cooper & Herskovits 1992). Basically, there are two combined steps involved in the BN construction process. The first is to perform variable search on a search space, and the other is to score each variable based on the degree of fitness (Heckerman 2008). The challenge however, is to ensure that good networks are learned with appropriate parameters using a scoring or fitness function to determine network variables from the given dataset. Thus, much of the research works on BN focus on developing scoring functions for the BN classifier (De Campos 2006). We argue that such scoring functions rely on many assumptions that make them less effective for SC tasks. For example, K2 algorithm, which is based on Bayesian Scoring function relies on the assumptions of parameter independence and assigning a uniform prior distribution to the parameters, given the class (Chen et al. 2008). We believe these assumptions lead to many false positives in the classification results as

[1] https://twitter.com/

[2] https://amazon.com/

sentiment classes are better captured by conditional dependency between words, rather than independent word counts (Airoldi et al. 2006, Bai 2011).

We also suggest that *varying* prior distribution could be assigned to each variable since each word has a natural *prior* probability of belonging to a particular sentiment class, independent of the data. For example, the word "good" is naturally positive and "bad" is naturally negative. Thus, in this work, we propose a sentiment scoring function that leverage sentiment information between variables in the given data. The output of the sentiment scoring function is then used to augment existing BN scoring functions for better performance. Our aim is to ensure sentiment information form part of the fitness criteria for selecting network variables from sentiment-oriented datasets such as reviews.

The proposed scoring function uses a simple but logical multi-class approach to compute the conditional mutual information between local variables in each class of instances. The conditional mutual information for all classes are then combined through a penalization process that uses the Minimum Description Length (MDL) principle. The final entropy score is further used to penalize the score from an existing BN scoring function. The local probabilities used in computing the conditional mutual information is computed using the popular Bayesian probability that uses the prior probability of a variable belonging to a natural sentiment class (i.e. independent of the given data by using individual word sentiment score from SentiWordNet (Esuli 2008)) and the observation of the variable in the selected class of instances and other classes in the dataset (e.g. positive, negative and neutral). The technique takes into account that the dependency score between two local variables x_i and x_j of a SC task would depend on two criteria:

- The posterior probability from multiple evidences that variables x_i and x_j have sentiment dependency.
- The sum of conditional mutual information between the variables for all classes.

The importance of the first criterion is that we are able to avoid the independence assumption made by the existing BN scoring functions. We capture local sentiment dependency between the variables as a joint probability of evidences from each variable and each class in the given data. Existing BN scoring functions uses the conditional *independence* given the data as a whole for determining dependencies between variables (De Campos 2006, Chen et al. 2008). Under such approach in SC, two independent words may occur with the same or similar frequencies in different classes. Thus, training BN classifier without penalizing such occurrences or dependencies, could affect the classifier decision to decide an appropriate sentiment class. Finally, the second criterion allows us to enforce strict *d-separation* policy between the network variables (Pearl 1988). Thus, only quality variables are used to form the dependency network for the BN classifier.

Section 2 of this paper discuss related work and additional motivations. In Section 3, we explain the problem background and then present the proposed sentiment augmentation technique in Section 4. Our experiment is described in Section 5. Finally, Section 6 gives the conclusion to our study and some thoughts on future research directions.

2 Related Work

2.1 Sentiment Classification (SC)

The most prominent of SC work is perhaps Pang et al. (2002) which employed supervised machine learning techniques to classify *positive* and *negative* sentiments in movie reviews. The significance of that work influenced the research community and created different research directions within the field of sentiment analysis and opinion mining (Liu 2012).

Turney & Littman (2002) uses unsupervised learning of semantic orientation to classify reviews based on the number of negative and positive phrases. They achieved an accuracy of 80% over an unlabeled corpus.

Pang & Lee (2004) proposed a subjectivity summarization technique that is based on minimum cuts to classify sentiment polarities in movie reviews. The intuition is to identify and extract subjective portions of the review document using minimum cuts in graphs. The minimum cut approach takes into consideration, the pairwise proximity information via graph cuts that partitions sentences which are likely to be in the same class. This approach showed significant improvement from 82.8% to 86.4% on the subjective portion of the documents. The approach also shows equally good performance when only 60% portion of a review document is used compared to an entire review document.

Choi & Cardie (2008) proposed a *compositional semantics* approach to learn the polarity of sentiments from the sub-sentential level of opinionated expressions. The compositional semantic approach breaks the lexical constituents of an expression into different semantic components.

Wilson et al. (2005) use instances of polar words to detect contextual polarity of phrases from the MPQA corpus. Each phrase detected is verified to be either *polar* or *non-polar* phrase by using the presence of opinionated words from a polarity lexicon. A detailed review of other sentiment classification techniques on different datasets is provided in Liu (2012) and Tang et al. (2009).

2.2 BN Classifiers for Sentiment Classification

Airoldi et al. (2006) and Bai (2011) proposed a two-stage Markov Blanket Classifier (MBC) approach to extract sentiments from unstructured text such as movie reviews by using BN. The approach learns conditional dependencies between variables (words) in a network and finds the portion of the network that falls within the *Markov Blanket*. The *Tabu Search* algorithm (Glover et al. 1997), is then used to further prune the resulting Markov Blanket network for higher cross-validated accuracy. While the use of Markov Blanket has shown to be effective in avoiding *over-fitting* in BN classifiers (Friedman et al. 1997), the MBC approach finds sentiment dependencies based on the ordinary *presence* or *absence* of words in their original sentiment class only. We identify sentiment dependencies by considering multiple sources of evidence. These include multiple sentiment classes in the data and the *natural* sentiment class of each variable which is independent of its sentiment class in the given data.

Similarly, Chen et al. (2011) proposed a parallel BN learning algorithm using MapReduce for the purpose of capturing sentiments from unstructured text. The technique experimented on large scale blog data

and captures dependencies among words using mutual information or entropy, with the hope of finding a vocabulary that could extract sentiments. The technique differs from Bai (2011) by using a three-phase (drafting, thickening and thinning) dependency search technique that was proposed in Cheng et al. (1997). Other than using *mutual information* in the *drafting* phase of the search technique, the work did not capture additional sentiment dependencies using other source of evidence.

Again, we do not focus on developing a search algorithm but a scoring technique that considers multiple sentiment-dependent information as part of the existing state-of-the-art scoring functions.

3 Problem Background

3.1 Bayesian Network (BN)

A Bayesian Network N is a graphical representation of a joint probability distribution between a set of random variables (Friedman & Yakhini 1996). The network consists of two components: (1) a DAG $G = (R_n, M_r)$ that represents the structural arrangement of a set of variables (nodes) $R_n = \{x_1, ..., x_n\}$ and a corresponding set of dependence and independence assertions (arcs) M_r between the variables; (2) a set of conditional probability distributions $P = \{p_i, ..., p_n\}$ between the parent and the child nodes in the graph.

In the DAG component, the existence of an directed arc between a pair of variables x_i and x_j asserts a conditional dependency between the two variables (Cheng & Greiner 2001). The directed arc can also be seen to represent *causality* between one variable and the other (Aliferis et al. 2010), that is, variable x_y is an existential cause of variable x_z, hence $x_y \rightarrow x_z$. The absence of an directed arc between a pair of variables, however, represents a conditional independence, such that, given a subset U of variables from R_n, the degree of information about variable x_i does not change by knowing x_j, thus $I(x_i, x_j|U)$. This also implies that $p(x_i|x_j, U) = p(x_i|U)$. The parent(s) of variable $x_i \in R_n$ is denoted by a set $pa_G(x_i) = x_j \in R_n|x_{ji} \in M_r$, and $pa_G(x_i) = \emptyset$ for the root node.

The conditional probability distributions of the DAG G is represented by its CPT, which contains a set of numerical parameters for each variable $x_i \in R_n$. These numeric parameters are computed as the probability of each variable given the set of parents, $p(x_i|pa_G(X_i))$. Over the set of variables in R_n, the joint probability for the BN is therefore obtained as follows:

$$p(x_1, ..., x_n) = \prod_{X_i \in R_n} p(x_i|pa_G(x_i)) \qquad (1)$$

Thus, for a typical classification task, the BN classifier would learn the numerical parameters of a CPT from the DAG structure G, by estimating some statistical information from the given data. Such information include, *mutual information* (MI) between the variables and *chi-square distribution* (De Campos 2006). The former is based on the *local score metrics* approach and the latter exhibits *conditional independence tests* (CI) approach. For both approaches, different *search* algorithms are used to identify the network structure. The goal is to ascertain, according to one or more search criteria, the best BN that fits the given data by evaluating the weight of the arc between the variables. The criteria for evaluating the fitness of the nodes (variables), and the arcs (parameters) in the BN search algorithms, are expressed as fitting or scoring functions within the BN classifier (De Campos 2006). Our goal is to ensure that those criteria include *sentiment-dependent* information between the variables. We will focus on penalizing existing *local score metrics* with our sentiment augmented scoring function for the BN classifiers, hence the SABN proposed in this paper.

The local score metrics are of particular interest because they exhibit a practical characteristic that ensures the joint probability of the BN is *decomposable* to the sum (or product) of the individual probability of each node (Friedman & Yakhini 1996)(De Campos 2006). To the best of our knowledge, very few research papers have considered sentiment-dependent information, as part of the fitness criteria for capturing dependency between the variables.

3.2 BN Scoring Functions

We focus on the local score metrics functions, K2, BDeu, Entropy, AIC and MDL (De Campos 2006). The functions define a fitness score, and a specified search algorithm searches for the best network that maximizes the score. Each of these functions identifies frequencies of occurrence of each variable x_i in the data D and a network structure N. In this paper, we assume that the scores generated by the scoring functions are somehow naïve, thus, we attempt to mitigate its effect on SC tasks. Firstly, we will define the parameters that are common to all the functions. We will then describe each of the functions with their associated formula and specify their limitations to the SC tasks.

Similar to Bouckaert (2004), we use $r_i (1 \leq i \leq n)$ to denote the size or cardinality of x_i. $pa(x_i)$ represents the parents of x_i and the cardinality of the parent set is represented by $q_i = \prod_{x_j \in pa(x_i)} r_j$. If $pa(x_i)$ is empty (i.e. $pa(x_i) = \emptyset$), then $q_i = 1$. The number of instances in a dataset D, where $pa(x_i)$ gets its jth value is represented by $N_{ij} (1 \leq i \leq n, 1 \leq j \leq q_i)$. Similarly, $N_{ijk} (1 \leq i \leq n, 1 \leq j \leq q_i, 1 \leq k \leq r_i)$ represents the portion of D where $pa(x_i)$ gets its jth value and x_i gets its kth value such that $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$. Obviously, N represents the size of D.

K2: This metric is a type of Bayesian scoring function proposed by Cooper & Herskovits (1992). The function relies on series of assumptions such as parameter independence and uniform prior probability for the network. We reiterate that instead of independent word counts, the sentiments expressed in a given data are better captured using conditional dependency between words and their related sentiment classes (Airoldi et al. 2006). The K2 metric is defined as follows:

$$S_{k2}(N, D) = P(N) \prod_{i=0}^{n} \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(r_i - 1 + N_{ij})!} \prod_{k=1}^{r_i} N_{ijk}! \qquad (2)$$

BDeu: The metric was proposed by Buntine (1991) as a generalization of K2. It resulted from Bayesian Drichlet (BD) and BDe which were proposed by Heckerman et al. (1995). The BD is based on hyperparameters η_{ijk} and the BDe is a result of BD with additional assumptions. BDeu relies on the sample size η as the single parameter. Since BDeu

is a generalization of K2, it carries some of our concerns expressed on K2 earlier. Most importantly, the uniform prior probability assigned to each variable $x_i \in pa(x_i)$ could be replaced by the probability of the variable belonging to a *natural* sentiment class as stated earlier. We suggest that this is likely to improve the performance of the sentiment classifier especially on sparse data distribution. We define the BDeu metric as follows:

$$S_{BDeu}(N,D) = P(N)\prod_{i=0}^{n}\prod_{j=1}^{q_i}\frac{(\Gamma(\frac{\eta}{q_i})}{\Gamma(N_{ij}+\frac{\eta}{q_i})} \prod_{k=1}^{r_i}\frac{\Gamma(N_{ijk}+\frac{\eta}{r_i q_i})}{\Gamma(\frac{\eta}{r_i q_i})} \quad (3)$$

Note that the function $\Gamma(.)$ is inherited from BD, and $\Gamma(c) = \int_0^\infty e^{-u}u^{c-1}du$ (De Campos 2006).

Entropy: Entropy metric measures the distance between the joint probability distributions of the network (De Campos 2006). This allows dependency information to be identified by computing the mutual information (or entropy) between pair of variables. Thus, a minimized entropy between a pair of variables denotes dependency relationship, otherwise, a large entropy implies conditional independence between the variables (Su & Zhang 2006)(Heckerman et al. 1995). While the entropy metric has been successful in measuring dependency information for BN classifiers, the local probabilities involved in the metric is largely computed based on *conditional independence* assumption given the data (i.e. using frequency counts for independent variables). We suggest that a joint probability of multiple evidences could improve the metric in BN classifiers for the SC tasks. The metric is defined as follows:

$$H(N,D) = -N\sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i}\frac{N_{ijk}}{N}\log\frac{N_{ijk}}{N_{ij}} \quad (4)$$

AIC: The AIC metric adds a non-negative parameter penalization to the entropy method (De Campos 2006). The metric is specified as follows:

$$S_{AIC}(N,D) = H(N,D) + K \quad (5)$$

Where K is the number of parameters, such that $K = \sum_{i=1}^{n}(r_i - 1).q_i$.

MDL: The MDL metric is based on the minimum description length principle which selects a minimum representative portion of the network variables through coding (De Campos 2006). Thus, the best BN is identified to minimize the sum of the description length for the data. The metric is defined as follows:

$$S_{MDL}(N,D) = H(N,D) + \frac{K}{2}\log N \quad (6)$$

The use of MDL has been particularly effective for selecting dependency threshold between variables in BN. The study in Friedman & Yakhini (1996), suggests that the mean of the total cross-entropy error is asymptotically proportional to $\frac{\log N}{2N}$, which is why the entropy metric is penalized in Equation 6.

In this paper, the proposed augmented score function is based on a straight forward Information Theory approach. The approach uses the entropy-based conditional mutual information (CMI) technique to measure the dependencies between the variables. The local probabilities for computing the CMI between two variables are derived as joint probability resulting from multiple evidences of both variables belonging to the same sentiment class. This is achieved by using a multiclass approach that measures the CMI in each sentiment class. The sum of the CMIs over the data is thereafter penalized using the MDL principle as suggested in Friedman & Yakhini (1996).

4 Sentiment Augmented Score (SAS)

In this section, we will show how we derived the sentiment augmented score for BN. Given a dataset D containing two or more sentiment classes, we divide D into $|C|$ subsets, where $D_1...D_c$ represent the sentiment classes which are present in D. Note that the process of creating the SASs is similar to the process of creating a CPT which contains the resulting network parameters from a particular search algorithm, given the data. Thus, we will create a SAS table (SAST) from the given data, and at the later stage, we will use the values in SAST to augment existing scores from the original CPT.

Creating an appropriate CPT or SAST is challenging, especially when there is a sheer number of variables in the given data (Cheng et al. 1997). In fact, local search algorithms such as *K2*, *Hill Climbing*, *TAN*, *Simulated annealing*, *Tabu search* and *Genetic search* have been developed to address this challenge (Friedman et al. 1997). Thus, we do not intend to repeat the sophisticated local search process in our augmented scoring technique. We use a straight forward approach that computes CMI as the dependency between a pair of variables, given a subset D_c. The resulting scores for each pair of variables is stored into the SAST. Equation 7 computes the CMI for a pair of variables. Note that this process is equivalent to the *drafting* phase proposed in Cheng et al. (1997) or the Chow and Liu algorithm in Chow & Liu (1968). We can therefore focus on computing the local probabilities $P(x_i)$ and $P(x_j)$ for the CMI. In this work, each local probability encodes the sentiment dependency information as a *joint probability* of multiple sentiment evidences. We suggest that the joint probability is better than using the ordinary variable presence or single frequency count.

$$CMI(x_i,x_j|C) = \sum_{x_i,x_j,c} P(x_i,x_j,c) \log\frac{P(x_i,x_j,c)}{P(x_i|c),P(x_j|c)} \quad (7)$$

4.1 Local probabilities for CMI

In order to compute the local probabilities $P(x_i)$ and $P(x_j)$, we adopt Bayesian probability (Lee 2012), to calculate the joint probability from multiple sentiment evidences. Bayesian probability encodes a *generative model* or *likelihood* $p(D|\theta)$ of the dataset with a *prior* belief $p(\theta)$ to infer a *posterior* distribution $p(\theta|D)$, see Equation 8. The idea is to determine a favourable posterior information of a particular variable belonging to its observed class, such that, the conditional mutual information between two dependent variables x_i and x_j increases when the posterior information for both variables in the same class is large.

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)} \quad (8)$$

However, in sentiment oriented documents such as product reviews, it is very common to observe variables that belong to different classes in one sentiment class. Pang et al. (2002) referred to such scenario as *thwarted expectation*. For example, a "positive" review document may contain certain "negative" words used to express dissatisfaction about an aspect of a product despite some level of satisfaction that the product might offer. With this kind of problem, it is much probable that a dependency network that is learned with ordinary frequency counts of each variable (regardless of the sentiment class) would no doubt leads to inaccurate sentiment classifiers. Figure 1 shows a sample BN resulting from a product review dataset upon performing attribute selection. In that network, variable *After* has a 1.0 probability of belonging to the *negative* and *positive* classes, respectively. Similarly, variable *not* has a 0.723 probability of belonging to a "positive" class rather than "negative". Every other variables in the network, has a split probabilities between both classes. Our aim is to remove the contrasting variables such as *After* and *not* from the dependency network or at least minimize its influence in the network such that the quality of the network is improved for sentiment classification.

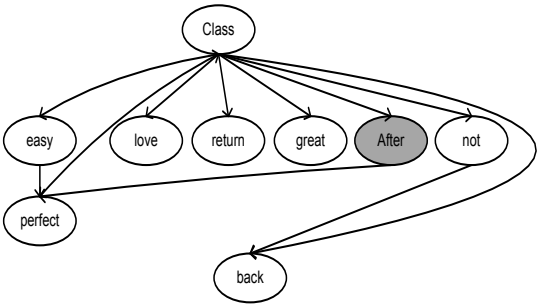


Figure 1: An example Bayesian network from product reviews.

Thus, in this work, we compute the posterior information for each variable by considering its *prior* information and joint *likelihood* or *observation* from all the classes available in the data.

The *prior* information is computed using the *natural sentiment or polarity scores* from SentiWordNet (Esuli & Sebastiani 2006). SentiWordNet gives the polarity scores of corresponding synsets for each English word. However, the polarity scores are often different for each of the synset entries. A synset contains multiple semantic or polarity interpretation of a given word. Each interpretation has three different polarities values. That is, a synset entry (word) would have a *positive*, *negative*, and *neutral* polarity scores which varies depending on the semantic interpretation of the word. An example of such words is *great*. Its fourth synset entry in SentiWordNet has 0.25 *positive*, 0.625 *negative*, and 0.125 *neutral* polarity scores, respectively.

In this work, we focus on the "positive" and "negative" sentiments, thus we will only consider positive and negative polarity scores from SentiWordNet. The challenge however, is to compute an absolute and single polarity score for each word from its multiple synset entries. First, we compute the score for each polarity independently and then find the polarity that maximizes the other. The score for the positive or negative polarity of all synset entries for a given word is computed as follows:

$$score_\phi(w) = \frac{1}{\epsilon}\sum_{i=1}^{\epsilon} E_c(e_i) \quad (9)$$

where $score_\phi(w)$ is the score for each polarity of the given word w, ϵ is the number of synset entries E for the word, c is the polarity or category (i.e. positive or negative) and e_i is each synset entry. Thus, the *prior* or *absolute polarity score* for w is computed as follows:

$$POL_\phi(w) = \underset{c\in C}{\operatorname{argmax}}\, score_\phi(w) \quad (10)$$

where $POL_\phi(w)$ is the maximum polarity score computed with respect to either *positive* or *negative* category c from all the syset entries.

We compute the *likelihood* information using a multi-class approach. Given a set of sentiment classes C, the probability of a variable belonging to its "first" observed sentiment class, is calculated as a joint probability of independently observing the variable in its first observed sentiment class and every other sentiment classes, $C_1...C_n$. Thus, the likelihood information is computed as follows:

$$p(x_1, ..., x_C|D) = \prod_{c=1}^{C} p(x_c|D) \quad (11)$$

Where $p(x_c|D)$ is the probability of a variable x belonging to a class c given the data D.

Given the data, our aim is to minimise the effect of the variables which might have appeared in a wrong (false positive) class as a result of *thwarted expectation* that was suggested in Pang et al. (2002), thereby biasing the dependency structure. Common examples are *negation* and *objective* words such as *not* and *After* as illustrated with Figure 1. If the word "not" for example, has a probability of 0.723 in a first observed "positive" class and a probability of 0.496 in the other negative class, then its *likelihood* of actually belonging to the "positive" class would be 0.359. Note that each probability is independent in this case as both probabilities do not sum to 1.

In addition, the *prior* or *natural sentiment score* (see Equation 10) obtained from SentiWordNet regulates the *likelihood* further, ensuring that the probability of a variable belonging to its first observed class is also conditioned on the natural sentiment class of the word which is independent of the data. With variable *not* having a probability of 0.625 *negative* from SentiWordNet, the *posterior* Bayesian probability is 0.149. This means the probability of the variable belonging to the *negative* class is higher (i.e. 0.85), and thus, should not be allowed to have strong dependency on a "true positive" variable. We suggest that this technique is more reliable than using the highest probability from both classes at the expense of accuracy (e.g. using only 0.723 and without the *prior*).

Thus, using the Bayesian probability defined in Equation 8, we substitute the *likelihood* information $p(x_1, ..., x_C|D)$ to $p(D|\theta)$ and the *prior* information $POL_\phi(w)$ to $p(\theta)$. Note that $P(D)$ is the sum of the two independent probabilities used in the likelihood (i.e. 0.723 and 0.496).

4.2 Sentiment Dependency Score

Having computed the local probabilities $P(x_i)$ and $P(x_j)$ using the Bayesian probability approach, we

compute the conditional mutual information as the dependency information between pair of variables in each class. Thus, we store the dependency information in the sentiment augmented score table, SAST. Again, the SAST is similar to the conventional CPT. The obvious difference is that sentiment information have been used to generate SAST. However, since we are using conditional mutual information to compute dependencies between variables, certain dependency threshold needs to be met in order to generate a reliable sentiment dependencies between each pair of variables in the SAST. As mentioned earlier, Friedman & Yakhini (1996) suggested that the mean of the total cross-entropy (mutual information) error is asymptotically proportional to $\frac{\log N}{2N}$. Using that MDL principle, we defined the threshold value as follows:

$$\Theta_{x_i,x_j} = \frac{\log N_c}{2N_c} \tag{12}$$

where Θ_{x_i,x_j} is the sentiment dependency threshold between a pair of variables x_i and x_j, N_c is the size of the data for a particular training class. Note that we generated individual SAST for each sentiment class in our data. In this work, a pair of variables x_i and x_j have strong sentiment dependency and get stored into the appropriate SAST, if and only if, the conditional mutual information $CMI(X_i, X_j|C) > \Theta_{x_i,x_j}$. Otherwise, we store a zero value to the corresponding slot in the SAST.

Finally, we reiterate our ultimate goal to penalize the dependency score from any of the existing scoring functions described in Section 3.2. Scoring functions such as K2 identifies dependency relationships by computing a *parent-child* score for a pair of variables x_i and x_j and checks if it maximizes a *base score* calculated as the total influence of a variable x_i on other variables in the data. We suggest that, for a sentiment classification task, the base score has highly minimized entropy in its current state, due to *false positive* variables as highlighted earlier. Thus, we penalize the base score of the existing scoring function by the SAST's *sentiment dependency score* between a pair of variables x_i and x_j. Arguably, this method creates reliable dependency network structures for training a sentiment classifier. Hence, we refer to this dependency network as Sentiment Augmented Bayesian Network (SABN). The *sentiment dependency score* for the SABN is defined below and it is computed as the sum of the conditional mutual information scores for the pair of variables x_i and x_j over all the sentiment classes.

$$Score_{SD}(x_i, x_j) = \sum_{c=1}^{C} CMI(x_i, x_j|C) \tag{13}$$

where C is the set of sentiment classes and $CMI(x_i, x_j|C)$ is the conditional mutual information score defined in Equation 7.

4.3 Summary of the SABN Algorithm

Algorithm 1 and Algorithm 2 are the two main algorithms involved in SABN. We will give a summary of the two algorithms as follows.

The purpose of Algorithm 1 is to generate the SAST which contains the CMIs between pairs of variables in the dataset. More importantly, sentiment information have been used to compute the local probabilities for each CMI. The algorithm takes as input a dataset D containing a set of labelled instances that are partitioned into a subset of classes D_c. For each subset D_c, CMI is computed for each pair of variables. Note that each CMI is checked against a MDL threshold. CMIs that are above the threshold are stored into the $SAST_c$ for the corresponding subset D_c. Thus, the algorithm outputs a set of SAST to be used in Algorithm 2. Again, the SAST is similar to the conventional CPT but with encoded sentiment information as part of the local probabilities that compute the CMI.

Finally, Algorithm 2 creates the sentiment dependency network as the BN. The algorithm takes as inputs the generated $SAST_{1,...,C}$ for the set of classes and the dataset D. For each variable in D, a *base score* is calculated using a specified base score function from the existing scoring function. The *parent-child* dependency score is also computed between each pair of variables using the specified parent-child dependency score function. Further, we compute our *sentiment dependency* score using the sum of CMIs for a specified pair of variables over the set of SASTs. The sentiment dependency score is then used to penalize the base score of the specified scoring function. If a parent-child dependency score is larger than the penalized base score, then a dependency exists between the selected pair of variables and then stored in the network. Thus, the output of the algorithm is the sentiment augmented Bayesian network that is used to build the sentiment classifier.

Algorithm 1 SAST(D)

Input : A set of labelled instances D.
Output : A set of Sentiment Augmented Score Tables for all pairs of variables x_i and x_j.

Steps

1: Partition instances D into subsets of classes D_c.
2: $SAST_{1,...,C}$ = empty.
3: **for each** subset D_c in D **do**
4: Compute the local probabilities $P(x_i)$ and $P(x_j)$ with Equation 8.
5: Use the local probabilities to compute CMI for each pair of variables x_i and x_j using Equation 7.
6: Compute the MDL threshold Θ with Equation 12.
7: **if** CMI > MDL threshold Θ_{x_i,x_j} **then**
8: Store the CMI into $SAST_c$ columns x_i, x_j and x_j, x_i, respectively.
9: **else**
10: Store 0 into $SAST_c$ columns x_i, x_j and x_j, x_i, respectively.
11: **end if**
12: **end for**
13: Return $SAST_{1,...,C}$

5 Experiments and Results

We conducted set of experiments using the proposed SABN algorithm on different product reviews. We then compared the accuracy with the ordinary BN classifier and a state-of-the-art sentiment classification technique.

Algorithm 2 SABN($SAST_{1,...,C}$, D)

Input : A set of $SAST_{1,...,C}$, training instances D.
Output : Sentiment Augmented Bayesian Network.

Steps

1: SABN = empty
2: **for each** variable x_i and x_j in D **do**
3: Get BaseScore(x_i) from a specified base score function in the search algorithm.
4: Get ParentChild(x_i, x_j) from a specified parent-child score function in the search algorithm.
5: $Score_{SD} = 0$.
6: **for each** subset $SAST_c$ in $SAST_{1,...,C}$ **do**
7: $Score_{SD} = Score_{SD} + SAST_c(x_i, x_j)$
8: **end for**
9: Penalize BaseScore(x_i) with $Score_{SD}$
10: **if** ParentChild(x_i, x_j) > BaseScore(x_i) **then**
11: Add dependency between x_i and x_j in SABN.
12: **end if**
13: **end for**
14: Return SABN

5.1 Datasets and Baselines

Our datasets consist of Amazon online reviews from three different product domains[3] that were manually crawled by Blitzer et al. (2007). These include *video*, *music*, and *kitchen* appliances. Each product domain consists of **1000** *positive* reviews and **1000** *negative* reviews, hence each domain has **2000** balanced set of instances. According to Blitzer et al. (2007), positive reviews were selected using a star rating of greater than 3 and negative reviews used a star rating of less than 3. Other ratings were discarded due to the ambiguity of their polarities. Note that 60% training and 40% testing sets were used on all domains. Table 1 shows details of the three datasets.

Table 1: Details of the three review datasets.

Dataset	Instances	Neg/Pos	Attributes
Kitchen	2000	1000/1000	1290
Music	2000	1000/1000	1292
Video	2000	1000/1000	1326

As our baseline, we implemented the popular sentiment classification technique in Pang & Lee (2004) using NB and SVM classifiers on our datasets with the same testing-to-training ratio. We also included additional baseline by using the ordinary BN without our proposed algorithm.

5.2 Data preparation

We implemented our algorithm within the *weka.classifiers.bayes* package of the WEKA[4] data mining framework. The SentiWordNet library[5] including the lexicon file were also incorporated into the same Weka directory. Further, we prepared our datasets according to the WEKA's ARFF format by concatenating the positive and negative reviews for each domain and created a string data file in ARFF format. The string data file was then converted to TFIDF data file in ARFF format using StringToWordVector filter with default settings. Note that the TFIDF format is just a processable numerical representation of the text variables that is supported by the *bayes* package. Arguably, the representation still maintains the dependency relationship between the words (variables) as in the original string format.

[3] http://www.cs.jhu.edu/ mdredze/datasets/sentiment/
[4] http://www.cs.waikato.ac.nz/ml/weka/
[5] http://sentiwordnet.isti.cnr.it/download.php

5.3 Results

Table 2: Accuracies of SABN and baseline classifiers on Amazon product reviews.

Dataset	SABN	Baseline-BN	Baseline-NB	Baseline-SVM
Kitchen	75.5%	70.7%	75.3%	76.3%
Music	**74.4%**	68.4%	73.9%	71.5%
Video	**81.5%**	72.6%	80.0%	77.1%

Table 2 shows the accuracies of the proposed SABN alongside other baseline classifiers. Baseline-BN represents the ordinary BN classifier in Weka. Baseline-NB and Baseline-SVM denote the implemented baseline technique on NB and SVM-SMO classifiers respectively. For both NB and SVM, we use the presence and absence of *unigram* features as suggested in Pang & Lee (2004). Note that both SABN and Baseline-BN used the SimpleEstimator with $\alpha = 0.5$ and K2 search algorithm with the Bayes/K2 scoring function.

As emphasised in Table 2, we observed the proposed SABN to have similar and in some cases improved performance compared to the baseline classifiers. For example, SABN recorded better improvements with average of 3.1% and 5.3% over the three baselines on Music and Video domains, respectively. We also note that the accuracies on the Amazon video reviews seems to be lower than the accuracies that were reported on the IMDb video reviews by Pang & Lee (2004). We suggest that this is a trade-off in sentiment classification on different datasets and/or domains as could be observed in our experiment on different Amazon domains. In addition, we believe that increased size of dataset, that is beyond the limited 1000 Amazon reviews, could further improve the accuracy of the SABN classifier.

In further experiments, we evaluated the performance of the SABN with reduced attribute sets since attribute selection tends to improve BN's accuracy (Airoldi et al. 2006). Thus, we ranked and reduced the set of attributes for each of our dataset by using the "AttributeSelection" filter in Weka. Specifically, we used the *InfoGainAttributeEval* evaluator with the *ranker* search algorithm. We used up to top-ranked 50 attributes for each domain and we performed classification with SABN and the Baseline-BN using 10-folds cross validation. For each domain, we report the number of attributes with the best accuracy. Table 3 shows the accuracies of the two classifiers on the three datasets.

With the attribute selection, we see that the accuracies of both SABN and Baseline-BN increased except for the Video domain. Again, we suggest that the accuracy of the SABN on the video domain could be improved with large dataset that may contain more representative attributes. Nevertheless, the accuracy of the SABN is still better than the Baseline-BN on the Video domain with the reduced attributes. We also performed experiment by using SABN with other

Table 3: Accuracies of SABN and Baseline-BN with the best ranked attribute sets.

Dataset	Ranked Attributes	SABN	Baseline-BN
Kitchen	50	75.7%	71.4%
Music	30	74.6%	69.7%
Video	50	77.8%	72.9%

scoring functions reported in Section 3.2 using the reduced attributes. The result in Table 4, shows that those scoring functions did not improve the result for SABN beyond the Bayes/K2 scoring function used in the earlier experiments. This is consistent with the comparative study conducted in De Campos (2006) on BN scoring functions.

Table 4: Experimental results using SABN with different scoring functions.

Function	Kitchen	Music	Video
K2/Bayes	75.7%	74.6%	77.8%
MDL	75.2%	71.7%	73.5%
BDeu	75.3%	71.8%	73.5%
Entropy	73.4%	70.0%	74.4%
AIC	75.2%	71.7%	72.8%

In terms of computational complexity, the SABN classifier has a training time complexity of $O(n^2.D)$ and a testing time complexity of $O(n)$, where n represents the count of the variables in the dataset and D denotes the size of the dataset. We believe this complexity is comparable with those of popular state-of-the-art classifiers, such as reported in Su & Zhang (2006). Overall, we have observed the SABN classifier to have reasonable performance that shows a promising research pathway for using Bayesian Network as a competitive alternative classifier for sentiment classification tasks.

6 Conclusion

In this study, we have proposed a sentiment augmented Bayesian network (SABN) classifier. The proposed SABN uses a multi-class approach to compute sentiment dependencies between pairs of variables by using a joint probability from different sentiment evidences. Thus, we calculated a sentiment dependency score that penalizes existing BN scoring functions and derived sentiment dependency network structure using the conditional mutual information between each pair of variables in a dataset. We performed sentiment classification on three different datasets with the resulting network structure. Experimental results show that the proposed SABN has comparable, and in some cases, improved calcification accuracy with state-of-the-art sentiment classifiers. In future, we will experiment with SABN on cross-domain datasets and large scale sentiment datasets.

References

Airoldi, E., Bai, X. & Padman, R. (2006), Markov blankets and meta-heuristics search: Sentiment extraction from unstructured texts, *in* 'Advances in Web Mining and Web Usage Analysis', Springer, pp. 167–187.

Aliferis, C. F., Statnikov, A., Tsamardinos, I., Mani, S. & Koutsoukos, X. D. (2010), 'Local causal and markov blanket induction for causal discovery and feature selection for classification part i: Algorithms and empirical evaluation', *The Journal of Machine Learning Research* **11**, 171–234.

Bai, X. (2011), 'Predicting consumer sentiments from online text', *Decision Support Systems* **50**(4), 732–742.

Blitzer, J., Dredze, M. & Pereira, F. (2007), Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification, Association of Computational Linguistics (ACL).

Boiy, E. & Moens, M.-F. (2009), 'A machine learning approach to sentiment analysis in multilingual web texts', *Information Retrieval* **12**(5), 526–558.

Bouckaert, R. R. (2004), *Bayesian network classifiers in weka*, Department of Computer Science, University of Waikato.

Buntine, W. (1991), Theory refinement on bayesian networks, *in* 'Proceedings of the Seventh conference on Uncertainty in Artificial Intelligence', Morgan Kaufmann Publishers Inc., pp. 52–60.

Chen, W., Zong, L., Huang, W., Ou, G., Wang, Y. & Yang, D. (2011), An empirical study of massively parallel bayesian networks learning for sentiment extraction from unstructured text, *in* 'Web Technologies and Applications', Springer, pp. 424–435.

Chen, X.-W., Anantha, G. & Lin, X. (2008), 'Improving bayesian network structure learning with mutual information-based node ordering in the k2 algorithm', *Knowledge and Data Engineering, IEEE Transactions on* **20**(5), 628–640.

Cheng, J., Bell, D. A. & Liu, W. (1997), Learning belief networks from data: An information theory based approach, *in* 'Proceedings of the sixth international conference on Information and knowledge management', ACM, pp. 325–331.

Cheng, J. & Greiner, R. (1999), Comparing bayesian network classifiers, *in* 'Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence', Morgan Kaufmann Publishers Inc., pp. 101–108.

Cheng, J. & Greiner, R. (2001), Learning bayesian belief network classifiers: Algorithms and system, *in* 'Advances in Artificial Intelligence', Springer, pp. 141–151.

Choi, Y. & Cardie, C. (2008), Learning with compositional semantics as structural inference for subsentential sentiment analysis, *in* 'Proceedings of the Conference on Empirical Methods in Natural Language Processing', Association for Computational Linguistics, Honolulu, Hawaii, pp. 793–801.

Chow, C. & Liu, C. (1968), 'Approximating discrete probability distributions with dependence trees', *Information Theory, IEEE Transactions on* **14**(3), 462–467.

Cooper, G. F. & Herskovits, E. (1992), 'A bayesian method for the induction of probabilistic networks from data', *Machine learning* **9**(4), 309–347.

De Campos, L. M. (2006), 'A scoring function for learning bayesian networks based on mutual information and conditional independence tests', *The Journal of Machine Learning Research* **7**, 2149–2187.

Esuli, A. (2008), 'Automatic generation of lexical resources for opinion mining: models, algorithms and applications', *SIGIR Forum* **42**(2), 105–106.

Esuli, A. & Sebastiani, F. (2006), 'Sentiwordnet: A publicly available lexical resource for opinion mining', *Proceedings of LREC* .

Friedman, N., Geiger, D. & Goldszmidt, M. (1997), 'Bayesian network classifiers', *Machine Learning* **29**, 131–163. 10.1023/A:1007465528199.
URL: *http://dx.doi.org/10.1023/A:1007465528199*

Friedman, N. & Yakhini, Z. (1996), On the sample complexity of learning bayesian networks, *in* 'Proceedings of the Twelfth international conference on Uncertainty in artificial intelligence', Morgan Kaufmann Publishers Inc., pp. 274–282.

Glover, F., Laguna, M. et al. (1997), *Tabu search*, Vol. 22, Springer.

Heckerman, D. (2008), *A tutorial on learning with Bayesian networks*, Springer.

Heckerman, D., Geiger, D. & Chickering, D. M. (1995), 'Learning bayesian networks: The combination of knowledge and statistical data', *Machine learning* **20**(3), 197–243.

Lee, P. M. (2012), *Bayesian statistics: an introduction*, John Wiley & Sons.

Liu, B. (2012), 'Sentiment analysis and opinion mining', *Synthesis Lectures on Human Language Technologies* **5**(1), 1–167.

Pang, B. & Lee, L. (2004), A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts, *in* 'Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics', Association for Computational Linguistics, Barcelona, Spain, p. 271.

Pang, B., Lee, L. & Vaithyanathan, S. (2002), Thumbs up?: sentiment classification using machine learning techniques, *in* 'Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10', Association for Computational Linguistics, pp. 79–86.

Pearl, J. (1988), *Probabilistic reasoning in intelligent systems: networks of plausible inference*, Morgan Kaufmann.

Su, J. & Zhang, H. (2006), Full bayesian network classifiers, *in* 'Proceedings of the 23rd international conference on Machine learning', ACM, pp. 897–904.

Tang, H., Tan, S. & Cheng, X. (2009), 'A survey on sentiment detection of reviews', *Expert Systems with Applications* **36**(7), 10760–10773.

Turney, P. & Littman, M. L. (2002), Unsupervised learning of semantic orientation from a hundred-billion-word corpus, Technical report.

Wilson, T., Wiebe, J. & Hoffmann, P. (2005), Recognizing contextual polarity in phrase-level sentiment analysis, *in* 'Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing', Association for Computational Linguistics, Vancouver, British Columbia, Canada, pp. 347–354.

A Concept-based Retrieval Method for Entity-oriented Search

Jun Hou and Richi Nayak
School of Electrical Engineering and Computer Science,
Queensland University of Technology, Brisbane, Australia
jun.hou@student.qut.edu.au,r.nayak@qut.edu.au

Abstract

Entity-oriented retrieval aims to return a list of relevant entities rather than documents to provide exact answers for user queries. The nature of entity-oriented retrieval requires identifying the semantic intent of user queries, i.e., understanding the semantic role of query terms and determining the semantic categories which indicate the class of target entities. Existing methods are not able to exploit the semantic intent by capturing the semantic relationship between terms in a query and in a document that contains entity related information. To improve the understanding of the semantic intent of user queries, we propose concept-based retrieval method that not only automatically identifies the semantic intent of user queries, i.e., Intent Type and Intent Modifier but introduces concepts represented by Wikipedia articles to user queries. We evaluate our proposed method on entity profile documents annotated by concepts from Wikipedia category and list structure. Empirical analysis reveals that the proposed method outperforms several state-of-the-art approaches.

Keywords: Query Reformulation, Concept-based Retrieval, Entity-oriented Retrieval.

1 Introduction

When people use a retrieval system for focused information, they often intend to look for particular information rather than finding an entire document or a long text passage. Entities play a central role in answering such information needs. This type of retrieval can be called as entity-oriented retrieval. Considering the query "Formula 1 drivers that won the Monaco Grand Prix", the desired result will be the list of names, instead of, finding the documents related to "Formula 1", or "Monaco Grand Prix". Research efforts such as Expert finding track (chen et al. 2006), INEX entity ranking track (Rode et al. 2009) and TREC entity track (Balog, Serdyukov and Vries 2010) emphasize that entities are an important information unit, in addition to a search unit, for providing exact answers to user queries.

In fact, user queries often consist of keywords and tend to be short, which makes it difficult for a search system to understand the query intent. In order to return accurate answers for user queries, researchers have proposed methods to interpret queries by classifying them into semantic categories which indicate the class of target entities (li et al. 2008; Manshadi and Li 2009). Query categorisation provides the ranking engine with a capability of filtering out irrelevant candidate entities. However, these methods fail to describe the semantic structure of user queries, i.e. the semantic role of individual query terms. The semantic role of individual query terms gives an insight of query of target entities. Considering the query "Formula 1 drivers that won the Monaco Grand Prix", a search system needs to return target entities of class "Formula 1 drivers" that won the "Monaco Grand Prix" rather than other Grand Prix or just drivers competing in "Monaco Grand Prix". Recent researches (Pound, Ilyas and Weddell 2010; Unger et al. 2012) have shown that understanding the semantic structure of a user query can be beneficial to entity-oriented retrieval.

However, simply applying query category classification or query semantic structure identification on traditional Bag-of-Words (BOW)-based information retrieval (IR) method (Moriceau and Tannier 2010; Demartini et al. 2010) fails to capture and exploit the semantic relationships that exist between terms of a query and of a document which contains entity related information. As known, different terms may be used to describe the same concept in a query and a document (the synonymy problem), and even the same term may be used to express different concepts (the polysemy problem). Consequently, a search system may return inaccurate and incomplete results without identifying the semantic concepts of related terms (Egozi, Markovitch and Gabrilovich 2011).

In this paper, solving these issues, we propose a novel concept-based retrieval method that integrates the semantic structure of a query and concept-based query expansion to expand query terms with concepts. More specifically, we define the semantic structure embedded in a query as Intent Type (IT) and Intent Modifier (IM) and automatically identify them using a Conditional Random Field (CRF) model. The semantic structure enables the retrieval system to focus only on entities that belong to the required class, i.e., Intent Type. In addition, the concept-based query expansion introduces concept-based features from the massive human knowledge base, i.e. Wikipedia, to related query terms. Considering the query "Formula 1 drivers that won the Monaco Grand Prix", the semantic structure can be identified as: $[\text{Formula 1drivers}]_{\text{IT}}$ $[\text{that won the Monaco Grand Prix}]_{\text{IM}}$. The concept-based query expansion detects the following concepts:

IT: Formula One, Formula racing, List of Formula One drivers.

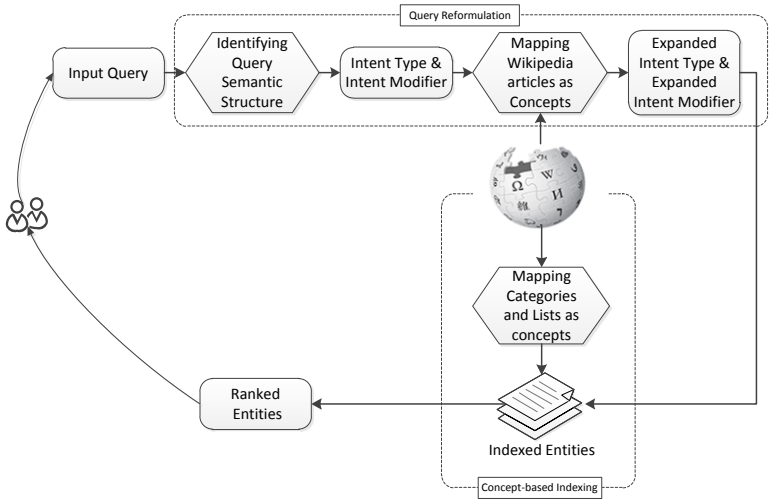


Figure 1: Overview of The Proposed Method of Concept-based Retrieval

IM: Monaco, Monaco Grand Prix, Grand Prix motorcycle racing.

This paper also presents a process of applying the proposed concept-based query structure analysis method on a collection annotated by concepts (and concept-based indexed) for retrieving target entities. Empirical analysis reveals that the proposed method outperforms several state-of-the-art approaches i.e., BOW-based model and conventional concept-based query expansion.

2 Related work

Entity-oriented retrieval (ER) aims to find a list of entities as the answer of a query. Different from the traditional document retrieval, ER includes extra processes such as Named Entity Recognition (NER) and Information Extraction (IE) for processing documents. ER can be classified into two groups based on how entities are ranked: (1) Profile-based models where entities are represented and ranked by related documents (profiles) that contain these entities; and (2) Document-based models where top-ranked entities are found from the top relevant documents (Balog, Serdyukov and Vries 2010). Our research is similar to the profile-based model where each entity has a corresponding document. Authors (Rode, Serdyukov and Hiemstra 2008) have proposed a profile-based retrieval model combing document and paragraph score to rank entities. Authors (Moriceau and Tannier 2010) proposed a lightweight document-based search system that combines entity-related syntactic information, i.e., syntactic dependency relations into classical search engine indexing. External entity repositories such as Yago (Demartini et al. 2010) and Freebase (Bron et al. 2010) have been integrated into entity-oriented search to utilize more entity-related information. However, the existing methods do not fully exploit the query semantic structure and semantic relationship between query and document terms.

The proposed method is also related to studies understanding the semantic structure of user queries that annotate individual query terms with semantic roles. Research work (Li 2010) studied the problem of labelling semantic class and attribute/value for noun phrase queries by learning a CRF-based tagger. A question analysis approach based on a set of heuristics (Moriceau and Tannier 2010) was proposed to identify answer type for different types of questions in Question Answering (QA). Researchers (Unger et al. 2012) developed a hybrid and generative query structure template framework for a similar task. Recently, a statistic-based method (Pound et al. 2012) has been proposed by learning query semantic structures from Web query log and Knowledge Base. Our work analyses the semantic structure of a query and further introduces concepts for individual components of the semantic structure.

Concept-based Retrieval (CR), another related work, integrates concept-based features derived from external resources into the keyword-based search model. CR consists of two major processes: (1) concept-based query expansion and (2) concept-based indexing. The CR methods introduce concepts to keyword-based representation and run a retrieval algorithm to return ranked results. The CR methods can be grouped into different categories based on how they augment keyword-based representation with external resources, e.g., by using manually built thesaurus (Grootjen and Weide 2006), by relying on term co-occurrence data (Schuetze and Pedersen 1995), or by extracting latent concepts from massive human knowledge base (Egozi, Markovitch and Gabrilovich 2011; Milne, Witten and Nichols 2007). However, CR methods only augment queries and documents with concepts without modelling the semantic structure embedded in a query.

In this paper, we propose a novel concept-based retrieval method integrating concept-based query expansion and semantic structure analysis of a query, i.e., Intent Type and Intent Modifier, for entity-oriented retrieval. The proposed approach not only models the semantic structure of a query for searching target group of entities but it also introduces concepts to solve the problem of semantic relationship between query and document terms. The proposed method, therefore, is able to improve the performance of retrieval system.

3 The Proposed Concept-based Retrieval Method

Figure 1 gives an overview of the proposed approach. An input query, formulated in natural language or in noun phrases, is analysed and represented in the form of

semantic structure, i.e., Intent Type and Intent Modifier. Next, the input search query is mapped to related concepts (represented by Wikipedia articles). Based on the semantic structure information expanded with concepts, a list of ranked entities are retrieved over the concept-based indexing annotated by concepts (Wikipedia categories and lists), and presented as search answers to users. We now present the various processes of the proposed method.

3.1 Query Reformulation

In this section, we discuss how to identify the semantic structure of an input query and introduce concepts related to the input query.

3.1.1 Identification of Query Semantic Structure

In this section, we define the semantic structure of an input query and present a method to identify the semantic roles of individual constituents of input queries. We assume an input query Q to be a sequence of query terms $Q = \{q_1, q_2, \dots q_{|V|}\}$ over the vocabulary V. The semantic structure for an input query is defined as Intent Type (IT) and Intent Modifier (IM):

Definition 1. *Intent Type*, t is a query segment $s = \{q_i, q_{i+1}, \dots, q_k\}$ where $|k| \leq |v|$ that has a conditional probability of implying the class of target entity:

$$t = p(t|s) \tag{1}$$

Definition 2. *Intent Modifier*, m is a query segment s' ($s' = \{Q - s\}$) that has a conditional probability of imposing constraints on the Intent Type:

$$m = p(m|s') \tag{2}$$

Let the semantic structure be a label sequence $L = \{[q_1]_{l_1}, [q_2]_{l_2}, \dots, [q_{|V|-1}]_{l_{|V|-1}}, [q_{|V|}]_{l_{|V|}}\}$ with labels $l_i \in \{t, m\}$. Our goal is to obtain:

$$L = \underset{l_i}{argmax}\, p(l_i|q_i) \tag{3}$$

In this paper, we propose to use Conditional Random Field (CRF) to maximizes the conditional likelihood $p(l_i|q_i)$. More specifically, when a labelled training query instance $(L'|Q')$ comes, we learn a set of feature functions $f(l'_{j-1}, l'_j, Q', j)$ (j is the position of label l'_j) and corresponding weight vector λ. To classify a testing query instance ($L|Q$), CRF selects the labelled semantic structure on Q that maximizes:

$$p(L|Q) = \frac{1}{Z(q_i)} exp\{\sum_{i=1}^{|V|+1} \lambda \cdot f(l_{i-1}, l_i, Q, i)\} \tag{4}$$

where $Z(q_i)$ is normalization function.

In order to train CRF, we define feature functions containing following features for query term q_i with input position x_i: (1) lexical features (all query terms and query term bigrams within a five size window around x_i), (2) part-of-speech (PoS) features (all PoS tags, PoS tag bigrams and PoS tag trigrams within a five size window around x_i), and (3) feature of a pair of adjacent labels l_{i-1}, l_i. The input query is first decomposed into lexical tokens and annotated with part-of-speech (PoS) tags. Research has shown that PoS tagging has an accurate performance even on keyword queries (Barr, Jones and Regelson 2008). Here, lexical features and PoS features exploit the syntactic constraints on query terms being IT or IM. For example, IT, which implies the class of target entity, is often expressed using noun or noun phrases. For IM, named entities and adjectives are often used as context modifier. In addition, the feature of a pair of adjacent labels captures the transition pattern between adjacent labels. For example, the boundary between IT and IM is often described by prepositions such as "on" and "by", verbs such as "won" and "situated", and "WH" words such as "which" and "that".

One simple example of feature function which produces binary value can be: if the current word q_i is "italian" with its PoS tag p_i ""JJ" (which means adjective word) and current label l_i is "IM":

$$f(l\_(i-1), l\_i, Q, i) = \begin{cases} 1 & if\ \ q_i = italian, p_i = JJ\ and\ l_i = IM \\ 0 & otherwise \end{cases} \tag{5}$$

In addition, the corresponding weight vector λ_i for a feature function f_i is controlled by CRF during training stage: if $\lambda_i > 0$, the probability of label l_i is increased ($f_i = 1$); if $\lambda_i < 0$, the probability of label l_i is decreased ($f_i = 0$).

3.1.2 The Concept-based Query Expansion

Concept-based query expansion is an automatic process of adding related concepts to a given query. The relevance of added concepts determines the quality of retrieved results. In this paper, we propose to select semantically-related concepts from Wikipedia. A concept is generated from a single Wikipedia article and represented by different forms, for example, the concept "earth" is referred to by different anchor texts: "the planet", "the world" and "the globe". Meanwhile, a query is considered as a set of key-phrases. The key-phrases are obtained by extracting word n-grams that appears in a query. The query expansion process is triggered when a Wikipedia concept appears as a key-phrase in a query.

Given the labelled query Q that consists of $|P|$ key-phrases and Wikipedia concept set C, the simplest expansion is the exact mapping that one key-phrase only maps to one Wikipedia concept, which we refer to as context concept C'. However, a key-phrase, for example, "bar", may refer to many Wikipedia concepts, "bar (establishment)", "bar (law)" or "bar (computer science)". Here, we select the most appropriate Wikipedia concept c_i from candidate concept set C'' for key-phrase p_i by a function:

$$F(c_i|p_i) = \underset{c_i}{argmax} \sum_{j=1}^{|C'|} f(c_i, c'_j))\ \ c'_j \in C', c_i \in C'' \tag{6}$$

where $f(c_i, c'_j)$ is a similarity function. According to

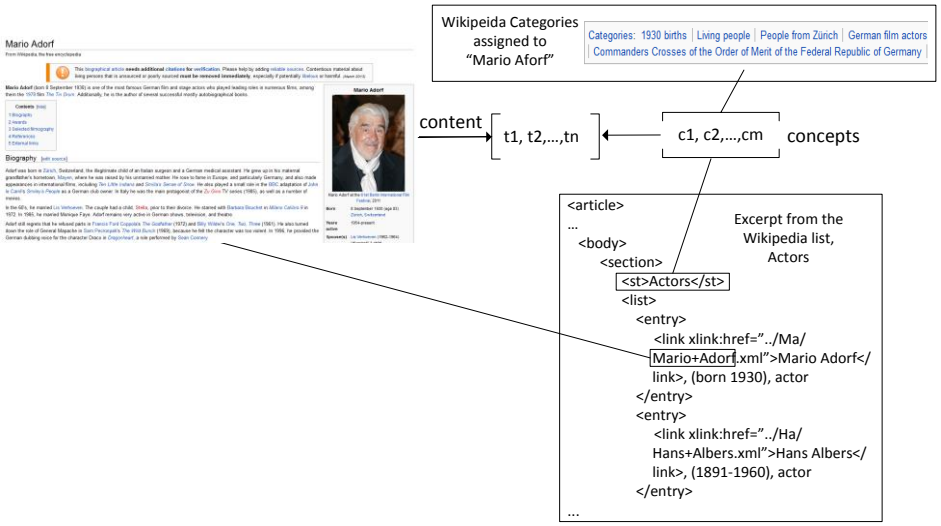


Figure 2: Overview of Concept-based Indexing

(Medelyan and Witten 2008), the similarity-based disambiguation method can accurately select the most appropriate concept. The similarity function $f(c_i, c_i')$ calculates the similarity between candidate concept c_i and context concept c_j' based on the overlap of hyperlinks between them (Milne and Witten 2008):

$$f(c_i, c_j') = 1 - \frac{max(log|c_i|, log|c_j'|) - log|c_i \cap c_j'|}{C - min(log|c_i|, log|c_j'|)} \quad (7)$$

where$|\cdot|$ is the number of hyperlinks. In addition, if a user query contains few text segments and there is no context evidence ($C' = \emptyset$) for selecting appropriate concept, the most commonly-used concept will be selected by a weight function w_{c_i} and equation 6 becomes:

$$F(c_i|p_i) = \begin{cases} \underset{c_i}{argmax}\left(c_i \left| \sum_{j=1}^{|C'|} f(c_i, c_j')\right.\right), if\ C' \neq \emptyset \\ w_{c_i} = \underset{c_i}{argmax}(\frac{f_{c_i}}{df_{c_i}}), \quad otherwise \end{cases} \quad (8)$$

where $f_{c_i}$is the frequency of c_i being an anchor text in Wikipedia and df_{c_i} is the total number of Wikipedia concepts in which c_i appears. Weight function w_{c_i} measures the commonness of the concept $c_i$in Wikipedia.

3.2 Concept-based Indexing

In this section, we define our concept-based indexing as a hybrid of unstructured and structured information consisting of entity profile documents and concepts introduced to them. Generally, entities and their related information are located in two categories of Web sources: unstructured information such as text in a Web document and structured information such as, records of RDF data or entries of a Web table. Research (Nie et al. 2007) found that balancing such unstructured and structured information is more suitable and is superior for entity-oriented retrieval. In this paper, the concept-based indexing adopts unstructured text (a unique Wikipedia article) as entity profile document and introduce structured information (i.e., Wikipedia category and list structure) as concepts.

More specifically, we represent an entity as the title of a Wikipedia article and use the Wikipedia article as entity profile document. This is because Wikipedia covers most entities of general domains and each Wikipedia article contains relatively complete and much cleaner information than Web pages (Gabrilovich and Markovitch 2006). In addition, we use structured information in Wikipedia, i.e., Wikipedia category and list structure, as concept features. Generally, a concept indexing mechanism needs scanning of natural language text within a document to find related concepts. It is time-consuming for millions of entities (Wikipedia articles), especially when the article length is relatively large. Therefore, we extracted Wikipedia categories and list structure (Schenkel, Suchanek and Kasneci 2007) as concepts. For example, as shown in Figure 2, a Wikipedia category is added as a concept to an entity if the Wikipedia category ("Living people") is assigned to the entity's corresponding Wikipedia article ("Mario Adorf"). In addition, if an entity's corresponding Wikipedia article is included as child element ("Mario Adorf") in a list structure (Figure 2), the parent element ("Actors") is introduced as a concept to the entity. As a result, the concept-based indexing E includes content terms of entity profile document and introduced concepts.

3.3 Concept-based Retrieval

Figure 3 summarizes the concept-based entity retrieval algorithm. Upon receiving an input query (q), the proposed method first identifies the query semantic structure (i.e., query Intent Type t and query Intent Modifier m by function $Struc(q)$). We then expands the query Intent Type and the query Intent Modifier with concepts (i.e., q' with the expanded query Intent Type t' and query Intent Modifier m' by function $Con(\cdot)$).

As returned results, we favour entities that match concept-expanded query terms and document score (W_d) measures the score between concept-expanded query (q') over concept-based indexing E (by function $InvIndexScore(\cdot)$ that represents the standard inverted index function that scores a document's match to a

INEX XER query topic 113	
Query	Formula 1 drivers that won the Monaco Grand Prix
Semantic structure	• $[\text{Formula 1drivers}]_{\text{IT}}$ • $[\text{that won the Monaco Grand Prix}]_{\text{IM}}$
Introduced concepts	• IT: Formula One, Formula racing, List of Formula One drivers; • IM: Monaco, Monaco Grand Prix, Grand Prix motorcycle racing.
Relevant results	Ayrton Senna, Michael Schumacher, Fernando Alonso, etc.
Class of target entity	racecar drivers, formula one drivers

Table 1: An example benchmark query from INEX XER

#Retrieve concept-based results
Input: Query q
Output: Ranked list of entities e
Procedure *Concept-Retrieval* (q)
$(t, m) \leftarrow Struc(q)$
$q' = (t', m') \leftarrow Con(t, m)$
Return *Ent-Retrieve* (q')
#Retrieve ranked entities for query q' from concept-based index
Procedure *Ent-Retrieve* (q')
For each $e \in E$
$W_d \leftarrow InvIndexScore(q', e)$
For each $c \in concept(e)$
$W_t \leftarrow InvIndexScore(t', c)$
$W_e \leftarrow W_d + W_t$
Return ranked list of entities, e based on W_e

Figure 3: The Proposed Concept-based Entity-oriented Retrieval Algorithm

query.). However, entities which contain matching concept-expanded query terms may be context entities serving as query intent modifier rather than target type of entities. Therefore, we add extra score to these entities that match the query Intent Type t using boosting factor. Boosting factor can be set either as a constant factor, or proportional to the relevance of certain query terms to an entity. Here, we implemented the latter basedon concept-expanded query Intent Type t. Intent Type score W_t scores concept-expanded query Intent Type t over concepts introduced to the entity profile document e (returned by function $concept(\cdot)$). We obtain the entity's score (W_e) as the sum of document score (W_d) and Intent Type score (W_t). Finally, entities are sorted and output to users according to the combined score.

4 Experiments and Evaluation

4.1 Document Collection and Query Set

We implemented the commonly used Lucene [1] search engine to build our concept-based indexing as discussed in Section 3.2. The document collection (Wikipedia) consists of about 2.6 million English XML articles with a total size of 50.7 GB. To evaluate our proposed concept-based retrieval, we constructed a benchmark query set from the INEX XER track (Rode et al. 2009). Relevant results for each individual XER query topic was manually assessed and used as ground truth. In addition, each benchmark query was annotated the class of target entities for evaluating the identified semantic structure, i.e., query Intent Type and query Intent Modifier. We selected 16 queries from XER query set containing keyword and natural language queries. Table 1 shows an example of benchmark query.

4.2 Experiments Setup

In order to test the effectiveness of the proposed concept-based entity retrieval method (Coss), we implemented three baseline methods for comparison. The first baseline approach (BOW) is the keyword-based search with the bag of words representation of documents without using concepts. The second baseline is the classical concept-based retrieval model that uses expanded concepts on query along with query terms, i.e., concept-based query expansion (Coq). The third baseline (Con) is modified version of our approach, i.e., using the algorithm in Figure 3 but without using Intent Type Score to rank the target entities, i.e., $W'_d \leftarrow W_d$ only in Procedure *Ent-Retrieve* $(F_q(t, m))$.

We evaluate the results of the proposed method and baseline methods using P@n (*n=10*) and Mean Average Precision (MAP). P@n is the fraction of the documents retrieved that are relevant to the user's information need:

$$P@n = \frac{Number\ of\ relevant\ documents}{Retrieved\ n\ documents}$$

where n is the number of retrieved documents.

Mean average precision (Map) for a set of queries is the mean of the average precision scores for each query:

$$MAP = \frac{\sum_{q=1}^{Q} AveP(q)}{Q}$$

where Q is the number of queries and $AveP(q)$ is calculated as follows:

$$AveP(q) = \frac{\sum_{k=1}^{n} P@(k) \times Rel(k)}{Number\ of\ relevant\ documents\ for\ q}$$

where $Rel(k)$ is an indicator function equalling 1 if the item at rank k is a relevant document, zero otherwise. For evaluation of the identified semantic structure, we use Label Accuracy (Acc) that is measured by the total number of labels divided by the total number of true positive predicted by CRF model. The label of a query term is true positive if the label assigned by the trained CRF model matches with its correct label.

4.3 Preliminary Results

Table 2 presents the Label Accuracy (Acc) for the identified semantic structure. As we can see, almost every query term is assigned with correct label. This is because CRF utilizes a rich set of feature functions to maximize

[1] http://www.lucene.apache.org

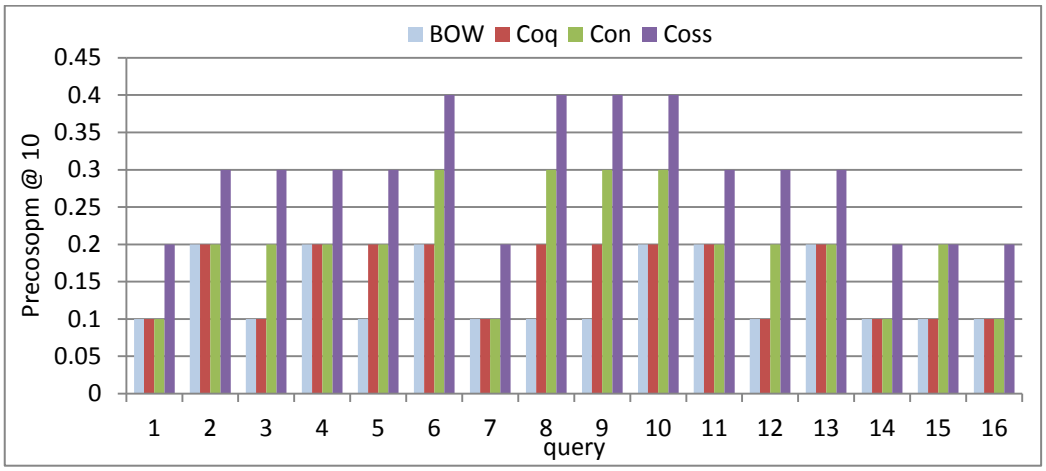


Figure 4: P@10 of Baseline Methods (BOW, Coq, Con) and the Proposed Method (Coss)

the conditional likelihood $p(l_i|q_i)$ for the label of query terms. The main reason for incorrect labels is the sporadic failure of named entity recognition. For example, query "National Parks East Coast Canada US", the phrase "East Coast" is not recognized as named entity, which leads to false labelling "National Parks" as Intent Modifier rather than Intent Type. As a final consideration, if it is the case proposed method (Coss) can not use Intent Type as boosting factor and it is equal to baseline method (Con) but we still report the result (i.e., query 15).

	Acc	Incorrect
CRF	94.7%	5.3%

Table 2: Label Accuracy (Acc) of the Identified Semantic Structure

4.4 Results and Discussion

In Figure 4, we present the p@10 results returned by four retrieval methods for each query. Table 3 shows the Mean Average Precision and Average Precision@10 of all methods.

	BOW	Coq	Con	Coss
MAP	0.12	0.13	0.16	0.19
Avg P@10	0.14	0.16	0.20	0.30

Table 3: Mean Average Precision (MAP) and Average Precision at rank=10 (Avg P@10) for All Methods

We observe that the proposed method (Coss) performs significantly better than all other methods. For a number of queries, query expansion with concepts do not improve the performance over the BOW approach, however the proposed method is still able to improve the retrieval result. This demonstrates that introducing concepts to the identified query intent can effectively help search system return relevant exact answers for user queries. For the query semantic structure, we find that since method Con does not fully exploit query intent in the ranking process, the Con has limited improvement in comparison to the proposed method Coss.

Comparing concept-based methods (Coq and Con) with keyword-based only method (BOW), we observe that introducing concepts can yield better results. This shows the effectiveness of introduced concepts in terms of bridging the gap of semantic relationships between query and document terms. Among these three methods, Con achieved the best performance. This is because concepts can be better used in both query and indexing stage, rather than just used in query expansion only.

In addition, we accumulated the number of relevant results among all queries. Figure 5 shows the sizes of the sets of results regarding to the "percentage of queries answered". We can see that the proposed method is capable of covering more percentage of the queries than all other baseline methods. The pure bag of words and classical concept based retrieval approaches is limited to retrieving relevant results over the query set. It confirms that analysing and using the structure of query with concept-based query expansion improves the performance of entity-oriented retrieval.

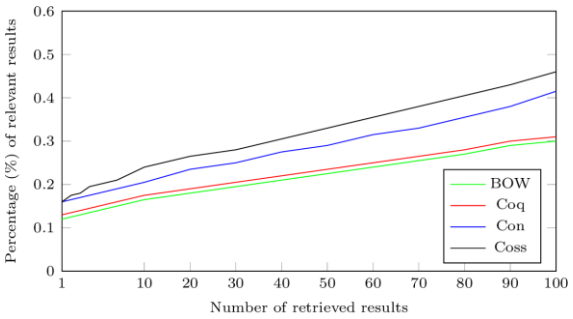


Figure 5: Percentage of Relevant Result Returned when Increasing the Number of Top Retrieved Results

4.5 Time Cost

To study the time cost of our approach, we conducted experiments on a machine with Intel Xeon X5690 3.47 GHz CPU, 96 GB memory. The average result time for query semantic structure analysis and concept-based expansion for all queries is less than 1 second.

The main time cost lies in the stage of extracting and loading statistics of Wikipedia concepts for concept-based query expansion, which took roughly 600 seconds. This is because that training CRF model for query semantic structure analysis requires less time than concept-based query expansion as it implemented smaller size of feature functions. However, these processes only need to be done once and can be performed offline beforehand. Such an overhead becomes acceptable.

5 Conclusions and Future Work

In this paper we proposed a novel concept-based search system for entity-oriented retrieval. We developed a query analysis framework which integrated query semantic structure and concept-based query expansion identification. For searching target entities, we presented a concept-based retrieval system which applied query analysis results to search over annotated documents. Experiment results showed that the proposed search system significantly improved the search performance over the traditional BOW-based method and the classical concept-based method. In future, we will apply our proposed retrieval model over structured and semi-structured data, such as relational database and RDF data, since entity-related information are increasingly extracted and represented as such data.

6 References

Balog, K., Serdyukov, P. and Vries, A. de (2010): Overview of the trec 2011 entity track. In *TREC 2011 Working Notes*, Maryland, USA, NIST.

Barr, C., Jones, R. and Regelson, M. (2008): The linguistic structure of english web-search queries. In *Proc. Conference on Empirical Methods in Natural Language Processing*, 1021-1030, Honolulu, USA.

Bron, M., He, J., Hofmann, K., Meij, E., Rijke, M. de, Tsagkias, M. and Weerkamp, W (2010): The Uni versity of Amsterdam at TREC 2010: Session, Entity and Relevance Feedback. In *Proceedings of the Nineteenth Text REtrieval Conference*, Maryland, USA, NIST.

Chen, H., Shen, H., Xiong, J., Tan, S. and Cheng, X. (2006): Social network structure behind the mailing lists: ICT-IIIS at TREC 2006 expert fnding track. In *Text REtrieval Conference (TREC)*, Maryland, USA, NIST.

Demartini, G., C. Firan, et al. (2010): Why finding entities in Wikipedia is difficult, sometimes. *Information Retrieval* **13**(5): 534-567.

Egozi, O., Markovitch, S. and Gabrilovich, E. (2011): Concept based information retrieval using Explicit Semantic Analysis. *ACM Transactions on Information Systems* **29**(2):8:1–8:34.

Gabrilovich, E. and Markovitch, S. (2006): Overcoming the Brittleness bottleneck using Wikipedia: enhancing text categorization with encyclopedic knowledge. *Proc. American Association for Artificial Intelligence*, 1301-1306, Boston, USA.

Grootjen, F. A. and Weide, van der T. P. (2006): Conceptual query expansion. *Data & Knowledge Engineering* **56**(2):174-193.

Li, Fangtao, Zhang, Xian, Yuan, Jinhui and Zhu, Xiaoyan (2008): Classifying what-type questions by head noun tagging. In *Proceedings of the 22nd International Conference on Computational Linguistics*, 481–488, Manchester, UK.

Li, X. (2010): Understanding the semantic structure of noun phrase queries. In *Proc. 48th Association for Computational Linguistics*, Uppsala, Sweden, 1337-1345, ACM press.

Manshadi, M. and Li, X. (2009): Semantic tagging of web search queries. In *Proc. Joint Conference of the 47th ACL and the 4th Intl. Joint Conference on Natural Language Processing*, 861-869, Singapore.

Medelyan, O., Witten, I., Milne, D. (2008): Topic indexing with wikipedia. In: *Proc. Of AAAI* , Chicago, USA, ACM press.

Milne, D., Witten, I.H. and Nichols, D.M. (2007): A Knowledge-Based Search Engine Powered by Wikipedia. In *Proceedings of the ACM Conference on Information and Knowledge Management*, Lisbon, Portugal, ACM press.

Milne, D. and Witten, I. H. (2008): An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In *Proceedings of the Workshop on Wikipedia and Artificial Intelligence: An Evolving Synergy at AAAI-08*, 25–30, Chicago, USA.

Moriceau, V. and Tannier, X. (2010): FIDJI: using syntax for validating answers in multiple documents. *Information Retrieval* **13**(5): 507-533.

Nie, Z., Ma, Y., Shi, S., Wen, J.-R., and Ma, W.-Y. (2007): Web object retrieval. In *Proceedings of the 16th international conference on world wide web*, 81–90, Banff, Canda.

Pound, J., Hudek, A., Ilyas, K. I. F. and Weddell, G. (2012): Interpreting keyword queries over Web knowledge bases. In *CIKM2012*, Maui, USA.

Pound, J., Ilyas, I. F. and Weddell, G. E. (2010): Expressive and Flexible Access to Web-extracted Data: a Keyword-based Structured Query Language. *SIGMOD*, Indianapolis, USA, ACM press.

Rode, H., Hiemstra, D., Vries, A., & Serdyukov, P. (2009): Efficient XML and entity retrieval with PF/Tijah: CWI and University of Twente at INEX'08. In *Advances in focused retrieval: 7th international workshop of the initiative for the evaluation of XML retrieval*, Dagstuhl Castle, Germany, 207–217, Heidelberg: Springer.

Rode, H. , Serdyukov, P. and Hiemstra, D. (2008): Combining document- and paragraph-based entity ranking. *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 851-852, Singapore, ACM press.

Schuetze, H. AND Pedersen, J. O. (1995): Information retrieval based on word senses. In *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*, 161–175, Las Vegas, USA.

Schenkel, R., Suchanek, F. and Kasneci, G. (2007): YAWN: A Semantically Annotated Wikipedia XML Corpus. In *Proceedings of Datenbanksysteme in Business, Technologie und Web*, 277-291, Aachen, Germany.

Unger, C., L. Buhmann, J. Lehmann, A.-C. Ngonga Ngomo, D. Gerber, P. Cimiano (2012): SPARQL Template-Based Question Answering. In *Proc. of the 22nd International World Wide Web Conference*, Lyon, France, ACM press.

Two Stage Similarity-aware Indexing for Large-scale Real-time Entity Resolution

Shouheng Li[1] **Huizhi Liang**[2] **Banda Ramadan**[3]

Research School of Computer Science, College of Engineering Computer Science
Australian National University,
Canberra ACT 0200,
[1]Email: sohey33@gmail.com

[2]Email: huizhi.liang@anu.edu.au

[3]Email: banda.ramadan@anu.edu.au

Abstract

Entity resolution is the process of identifying records in one or multiple data sources that represent the same real-world entity. How to find all the records that belong to the same entity as the query record in real-time brings challenges to existing entity resolution approaches. The challenge is especially true for large-scale dataset. In this paper, we propose to use a two-stage similarity-aware indexing approach for large-scale real-time entity resolution. In the first stage, we use locality sensitive hashing to filter out records with low similarities for the purpose of decreasing the number of comparisons. Then, in the second stage, we pre-calculate the comparison similarities of the attribute values to further decrease the query time. The experiments conducted on a large-scale dataset with over 2 million records shows the effectiveness of the proposed approach.

Keywords: Entity Resolution, Real-time, Blocking, Locality Sensitive Hashing, Scalability, Dynamic Data.

1 Introduction

With the utilisation of databases and information systems, businesses, governments and organisations are able to collect massive information without much difficulty. However, the raw data might be *dirty*, containing data that are incomplete, inconsistent and noisy. So the raw data is often required to be *pre-processed* or *cleaned* before further use. One of the important steps in data pre-processing is called entity resolution, or data integration, which is the process that identifies and matches data records that refer to the same real world entity.

Entity resolution can help to reduce the noise in data and improve data quality (Elmagarmid et al. 2007). Currently, most available entity resolution techniques conduct the resolution process in offline or batch mode, while the dataset is usually static (Christen 2012). However, in real world scenarios, many applications require rapid real-time responses. For example, online entity resolution based on personal identity information can help a bank to identify fraudulent credit card applications (Christen 2012). The requirement of dealing with large-scale dynamic data and providing rapid responses brings challenges to current entity resolution techniques.

Typically, pair-wise comparison is used to find the records that belong to the same entity. However, the number of comparisons increases dramatically when the size of the dataset grows quickly. Blocking or canopy formation can help to significantly decrease the number of comparisons (Christen 2012). Blocking divides the data into blocks and only compares the query record with all other records within the same block. For example, Soundex and Double-Metaphone are commonly used blocking approaches (Christen 2012). Moreover, another technique called Locality Sensitive Hashing (LSH) can find approximate similarity records quickly via hashing. It provides a similarity based filtering that "hashes" similar data records together.

Recently, Christen et al. (2009) proposed an Similarity-aware Indexing technique that improves the performance of traditional indexing by pre-calculating similarities of attribute values. Nevertheless, this approach needs to compare every record that has one or more encoding values that are the same with the query record, even though the compared record has a low overall similarity with the query record. If we can filter out those low similarity records first, then we can reduce the number of comparisons to speed up the query time. In this paper, we propose a two-stage similarity-aware indexing approach. At the first stage, locality sensitive hashing is adopted to approximately filter out those data records that have low similarity with the query record and allocate the potential matches in the same block. At the second stage, similarity-aware indexing is used to compare potential matches to obtain a precise and accurate result.

The rest of the paper is organized as below. Firstly, the related work will be briefly reviewed in Section 2. Then, the proposed approaches will be discussed in Section 3. In this section, the two-stage similarity-aware indexing approach is presented. In Sections 4 and 5, the design of the experiments, experimental results, and discussions will be presented. Finally, the conclusions of this work will be given in Section 6.

2 Related Work

The purpose of entity resolution is to find records in one or several databases that belong to the same real-world entity. Such an entity can be a person (e.g. cus-

tomer, patient or student), a product, a business, or any other object that exists in the real world. Entity resolution aims to link different databases together (in which case it is known as Record Linkage or Data Matching) and can also identify duplicate records in one database (known as De-duplication) (Christen 2012). Entity resolution is widely used in various applications such as identity crime detection, estimation of census population statistics, and retrospective construction of samples of persons for health research (Elmagarmid et al. 2007, Christen 2012). Currently, most available entity resolution techniques conduct the resolution process in offline or batch mode. Only limited research into using entity resolution at query time (Lange & Naumann 2012) or in real-time (Christen et al. 2009, Ramadan et al. 2013) has been conducted.

Indexing techniques can help to scale-up the entity resolution process. Commonly used indexing approaches include standard blocking based on inverted indexing and phonetic encoding, q-gram indexing, suffix array based indexing, sorted neighborhood, multi-dimensional mapping, and canopy clustering (Christen 2012). Typically, these existing approaches index one or more attribute values manually selected based on expert domain knowledge. Some recent research has proposed automatic blocking mechanisms (Das Sarma et al. 2012) or learning algorithms to find the best blocking schemes for record linkage (Michelson & Knoblock 2006).

Christen et al. (2009) proposed a similarity-aware indexing approach for real-time entity resolution. However, this approach fails to work well for large-scale databases, as the number of similarity comparisons for new attribute values increases significantly as the sizes of blocks increases with the growing number of records. Recently, Ramadan et al. (2013) extended this approach to facilitate a dynamic approach to index whereby the index is extended as a database grows. They showed that the insertion of new records into the index, as well as querying the index for real-time entity resolution grows sub-linearly as the size of the database index grows. However, this approach still requires a large number of pair-wise comparisons as it compares every record that has one or more encoding values that are the same with the query record, while the comparisons of records has a low overall similarity with the query record are not necessary.

Locality Sensitive Hashing (LSH) can help to return approximate similar records of a query quickly. Such approaches are widely used in Nearest Neighbor and Similarity Search in applications such as image search (Dong et al. 2008), recommender systems (Li et al. 2011), and entity resolution (Kim & Lee 2010). More recently, the work of Gan et al. (2012) proposed to use a hash function base with n basic length-1 signatures rather than using fixed l length-k signatures or a forest to represent each record. The records that are frequently colliding with a query record across all the signatures are selected as the approximate similarity search results. However, this approach needs to scan all the data records in the blocks to get the frequently colliding records each time, which results in the difficulty of returning results quickly when the sizes of the blocks are big for large-scale datasets. This approach can be used in real-time scenario, if we use the dynamic collision counting method for length-k ($k > 1$) blocks.

Although both LSH and indexing approaches (e.g., Similarity-aware Indexing(Christen et al. 2009, Ramadan et al. 2013)) can be effectively used in real-time entity resolution, how to combine them together to facilitate large-scale real-time entity resolution still needs to be explored.

Record ID	Entity ID	First name	Last name	Suburb	Zipcode
r_1	e_1	halle	bryant	turner	2612
r_2	e_2	kristine	jones	city	2601
r_3	e_3	hailey	pitt	turner	2612
r_4	e_4	christy	greg	belconnen	2617
r_5	e_2	christine	jones	city	2601

Table 1: Example records, r_5 is the query record

Record ID	h_1	h_2	h_3	h_4
r_1	1	2	3	4
r_2	5	6	7	8
r_3	9	10	3	11
r_4	12	13	14	15
r_5	5	16	7	8

Table 2: Length-1 minHash signatures, generated using four minHash functions h_1, h_2, h_3 and h_4

3 A Two-stage Similarity-aware Indexing Approach

We propose a two-stage similarity-aware indexing approach called LSI. LSI includes three indexes: a locality sensitive hashing index named LI, a similarity index named SI, and a standard blocking index named BI. At the first stage, we use the locality sensitive hashing index LI to filter out records with low similarities with the query records. Records with high similarities are preserved and stored in same blocks in the LI. Then at the second stage, records in the same block in the LI are considered as candidate matches and are compared pair-wisely. Candidate matches' attribute values are grouped together using encoding techniques and stored in the blocking index BI. A attribute value is compared with other attribute values which have the same encoding value, their similarities are pre-calculated and stored in the similarity index SI. The proposed approach performs well in real-time entity resolution scenarios for two reasons: firstly, comparisons on record pairs with low-similarities are avoided; secondly, most similarities are pre-calculated and can be retrieved from the SI therefore time of comparisons are saved. The proposed indexing approach LSI will be discussed in details in Section 3.1 and 3.2. Then in Section 3.3, we will discuss how the proposed indexing approach LSI can be applied in real-time entity resolution.

[Example 1] Table 1 shows an example of five records r_1, r_2 , r_3, r_4, and r_5 with four attributes: First Name, Last Name, Suburb, and Zipcode. They belong to four different entities e_1, e_2, e_3 and e_4. Suppose r_5 is a query record, the entity resolution process for r_5 is to find r_2 based on the four attribute values.

3.1 First stage: Locality Sensitive Hashing

A locality sensitive hashing family can help to find approximate results. Let h denote a hash function for a given distance measure approach D, $Pr(x)$ 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, if for any two records r_x and r_y, the following conditions hold:

1. if $D(r_x, r_y) \leq d_1$ then $Pr(h(r_x) = h(r_y)) \geq p_1$

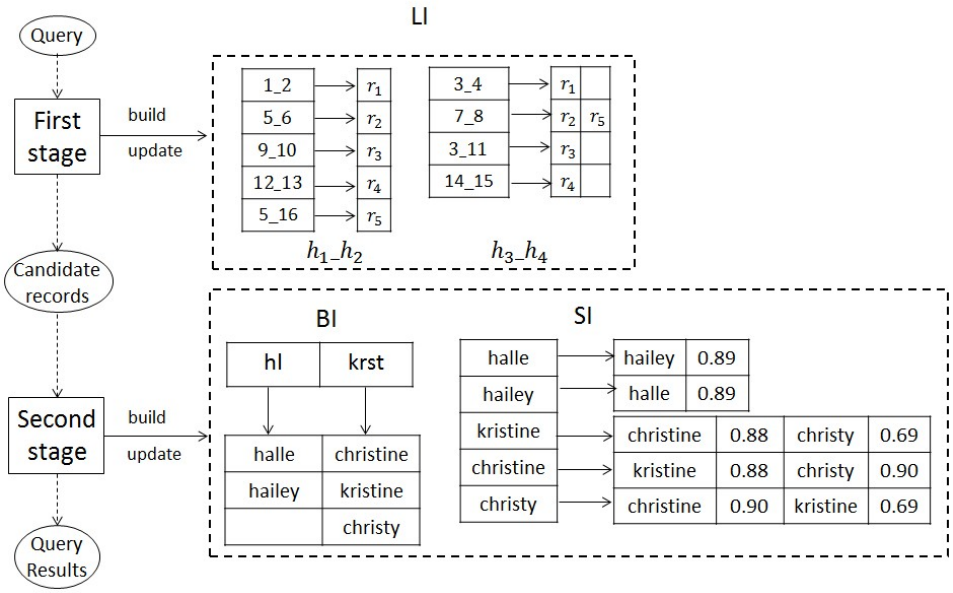


Figure 1: An example LSI index for the example records in Table 1

Record ID	First name	Encoding
r_1	halle	hl
r_2	kristine	krst
r_3	hailey	hl
r_4	christy	krst
r_5	christine	krst

Table 3: Double-Metaphone Encoding of the First name attribute values in Table 1

2. if $D(r_x, r_y) > d_2$ then $Pr(h(r_x) = h(r_y)) \leq p_2$

Minwise hashing (minHash) is a popularly used locality sensitive hashing approach that estimates the Jaccard similarity. Let $J(r_x, r_y)$ denote the Jaccard similarity for any two records r_x and r_y. The minHash method applies a random permutation π on the attribute values (i.e., elements) of any two records (i.e., sets) r_x and r_y and utilizes

$$p = Pr(min(\pi(r_x)) = min(\pi(r_y))) = J(r_x, r_y) = \frac{r_x \cap r_y}{r_x \cup r_y} \quad (1)$$

to estimate the Jaccard similarity (Gionis et al. 1999) of r_x and r_y (Gionis et al. 1999), where $min(\pi(r_x))$ denotes the minimum value of the random permutation of the attribute values of record r_x. p denotes the hash collision probability $Pr(min(\pi(r_x)) = min(\pi(r_y)))$. It represents the ratio of the size of the intersection of the attribute values of the two records to that of the union of the attribute values of the two records. To avoid unnecessary comparisons with low similarity records and secure the response time for query records in real-time, we use locality sensitive hashing to filter out those low similarity records at the first stage.

Each record is assigned with a set of minHash signatures via a family of minHash functions. For example, Table 2 shows the minHash signatures of the example records in Table 1. In the example of Table 2, four hash functions h_1, h_2, h_3, h_4 are used. They generate length-1 signatures for each record. However, having common length-1 minHash values does not necessarily mean two records are similar. In situations where some attribute values are frequent, dissimilar records usually have common length-1 minHash values too, which means most records will be assigned with same length-1 minHash values and minHashing does not really narrow down the range of candidate records.

[Example 2] (Length-1 minHash signatures). Table 2 shows the length-1 minHash values of the example records given in Table 1. Records that have common attribute values are likely to have same length-1 minHash values, e.g., r_1 and r_3 have the same minHash value 3. However, although r_1 and r_3 have same length-1 minHash signatures, they actually belong to different entities.

Therefore, in order to filter out the noisy matches, a technique called *banding* is introduced to enable a rigid filtering. Banding tunes the strictness of minHash filtering by combining multiple minHash functions to form a *band*. The minHash values generated by minHash functions in a band are combined to form a minHash signature. Banding enhances the filtering strictness by applying a logic "AND" on minHash values. The number of minHash values in a band is known as bits denoted as k. Together with the And-construction, Or-constructions are conducted to increase the collision probability. More Or-constructions will introduce more hash tables denoted as l.

At the first stage, for each record, k hash functions are used to generates a length-k signature through And-construction. To increase the collision probability, each record is hashed l times to conduct Or-construction and form l hash tables (i.e., l length-k signatures), $n = k \times l$. Each record is indexed by l length-k minHash signatures.

After we get the minHash signature, we store the records' identifiers and their minHash signatures into an index named Locality Sensitive Hashing Index (i.e., LI). For a query record, the records with the same minHash signatures, also known as candidate records, can be quickly found by looking up the LI index.

[**Example 3**] **(LI: Locality Sensitive Hashing Index)**. Left figure in Figure 1 shows the locality sensitive hashing index of the example records in Table 1 using the parameters $k = 2$ and $l = 2$. r_2 and r_5 are put in the same block in the LI with minHash signature 7_8.

After we get the candidate records in each locality sensitive hashing block, we need to compare the similarity of each candidate record with the query record. The pre-calculation of the similarity of two attribute values can help to avoid large number of comparisons in real-time. This will be discussed in Section 3.2.

3.2 Second stage: Similarity-aware indexing

At the second stage, we adopt the idea of Similarity-aware Inverted Indexing (Christen et al. 2009, Ramadan et al. 2013) to conduct pair-wise comparisons.

As mentioned in Section 3.1, an input query's candidate matches can be obtained by looking up the query's minHash signatures in the LI. After that, the query record needs to be compared with each of the candidate records in order to get a precise list of true matches. The pair-wise comparisons are done by comparing the two records' attribute values accordingly with approximate string comparison approaches such as the Winkler function (Ramadan et al. 2013). For large datasets, there are often thousands of candidate records to compare, thus the pair-wise comparisons are computationally expensive if they have to be done in real-time. However, in real-world situations, attribute values may appear frequently, such as the names and zipcodes of populous suburbs, some popular personal names, etc. For this reason, in the Similarity-aware Inverted Indexing, the similarities of attribute values are pre-calculated, so that the similarities between attribute values can be retrieved from the Similarity Index (SI) rather than calculated online and thus the comparison time can be saved.

The Similarity-aware Inverted Indexing works in the following manner. A record is firstly processed using encoding techniques, each attribute of the record generates a encoding blocking key value. Attribute values are then stored in an index call Blocking Index (BI) under the corresponding encoding blocking key values. The BI is introduced for the purpose of reducing the number of comparisons between attribute values as an attribute value is only compared with other attribute values that have the same encoding blocking key value (same block in BI).

[**Example 4**] **(BI: Blocking Index)**. The attribute "First name" in Figure 1 is used to illustrate the process. Attribute values that have the same Double-Metaphone encoding are put in the same block in the BI (shown in Figure 1). So *halle* and *hailey* are put in the block of key *sm0*; *christine*, *kristine* and *christy* are put in the block of key *krst*.

The comparisons are conducted via calculating each pair's similarity using comparison functions. The calculated similarities are then stored in an index call Similarity Index (SI) for future retrieval purpose.

[**Example 5**] **(SI: Similarity Index)**. Each attribute values in the BI is compared with others in the same block using the Winkler comparison function. So *halle* is compared with *hailey*, *zach* is compared with *zack*, *christine*, *kristine* and *christy* are compared with each other. The calculated similarities are then stored in the SI. For instance, the similarities among of *kristine* and *christine* and *christy* are stored as shown in Figure 1.

3.3 Real-time Entity Resolution

The LSI includes three indexes: LSH Index (LI), Block Index (BI) and Similarity Index (SI). This section describes how the proposed approach is applied to real-time entity resolution. Similar to other indexing techniques (Christen 2012), the proposed indexing approach has two phases: building phase and querying phase.

3.3.1 Building phase

In the building phase, every record is treated as a new record and is used to build the indexes. In the beginning, all the three indexes are initialised to be empty. While a record is processed, it is firstly inserted into the LI according to the record's minHash signatures. Each minHash signature corresponds to a unique "bucket" in the LI, empty "buckets" are initialised for new minHash signatures. The record's identifier is then inserted into every bucket that the minHash signatures correspond to. Afterwards, the record is used to build the BI. Encoding blocking key values are generated based on attribute values of the record. Similar to the LI, each encoding blocking key value corresponds to a "block" in the BI. Since phonetic encoding is used, attribute values in a same "block" are similar in pronunciations. The attribute values of the inserted record are then added into the "blocks" which the encoding blocking key values correspond to in the BI. Finally, each attribute value of the inserted record is compared with other attribute values in the same "block" in the BI. The calculated similarities are then stored in the third index, SI. The building process is briefly described in Algorithm 1. A record **r**'s identifier **r**.0 is firstly inserted into the LI. Then the *insertion* subroutine is called to insert **r** into the BI and SI.

In the building phase, every record is treated as a new record and is used to build the indexes. In the beginning, all the three indexes are initialized to be empty. While a record is processed, it is inserted into the locality Sensitive Hashing Index (i.e., LI) based on the record's minHash signatures at the first stage. Each minHash signature corresponds to a unique block in the LI, empty blocks are initialised for new minHash signatures. The record's identifier is then inserted into every bucket that this record's minHash signatures correspond to.

Then, at the second stage, for each attribute value of a record in every Locality Sensitive Hashing block, we build the standard encoding block (i.e., BI) based on their encoding blocking key value. Afterwards, the record is used to build the BI. encoding blocking key values are generated based on attribute values of the record. Similar to the LI, each encoding blocking key value corresponds to a block in the BI. If phonetic encoding is used, then the attribute values in the same block are similar in pronunciations. The attribute values of the inserted record are then added into the blocks which the encoding blocking key values correspond to in the BI. After we build the BI, each attribute value of the inserted record is compared with other attribute values in the same block in the BI. The calculated similarities are then stored in the Similarity Index SI. The building process is briefly described in Algorithm 1. A record **r**'s identifier **r**.0 is firstly

Algorithm 1: ***Building phase***

input : Input dataset **D**; number of attributes n; minHash function **H**; encoding functions **E**; similarity functions **S**; **r** is a record in **D**, **r**.0 is the record identifier, **r**.i is **r**'s attribute value, $i = 1, ..., N$
output: Indexes **SI**, **BI** and **LI**

```
1  Initialise SI = {}
2  Initialise BI = {}
3  Initialise LI = {}
4  for r ∈ D do
       // First stage, insert record ID into the LI.
5      Sig = H(r)
6      for sig ∈ Sig do
7          if sig ∉ LI then
8              Initialise bk = []
9              Append r.0 to bk
10             LI[sig] = bk
11         else
12             Append r.0 to LI[sig]
       // Second stage, insert attribute values into the BI
          and SI.
13     Insert(r, n, E, S, SI, BI, LI)
```

Algorithm 2: ***Insertion***

input : Record **r** number of attributes n; encoding functions **E**; similarity functions **S**; indexes **SI**, **BI** and **LI**.
output: Updated indexes **SI**, **BI** and **LI**

```
   // Second stage, insert attribute values into the BI and
      SI.
1  for i = 1...n do
2      if r.i ∉ SI then
3          c = E_i(r.i)
4          b = BI[c]
5          Append r.i to b
6          BI[c] = b
7          Initialise si = []
8          for v ∈ b do
9              s = S_i(r.i, v)
10             Append (v, s) to si
11             oi = SI[v]
12             Append (r.i, s) to oi
13             SI[v] = oi
14         SI[r.i] = si
```

inserted into the LI. Then the *insertion* subroutine is called to insert **r** into the BI and SI.

The most important part of the building phase is the insertion subroutine, it is shown in Algorithm 2.

The insertion subroutine takes a record **r**, attribute values of **r** are checked to see if they have been indexed. If an attribute value **r**.i has not been indexed previously, e.g. not in the SI, the attribute value will be inserted into both SI and BI. For instance, if **r**.i is not previously indexed, the inserting process will firstly compute its encoding value c, and add **r**.i into BI using c as encoding blocking key value. The block list **b** which contains other attribute values with the same encoding blocking key value c will then be retrieved. The similarities (denoted as s) between **r**.i and other attribute values v in **b** will be calculated using the comparison function **S**. In the SI, each attribute value has a list that stores its similarity with other attribute values. So a new similarity list si will be initialised and similarities will be stored in it in the form of tuples (v, s) where v is other attribute value and s is the similarities between **r**.i and v. Next, for each of the attribute values v in **b**, its similarity list oi will be retrieved and the similarity will be added to it in the form of tuple (**r**.i, s). Finally, the updated indexes SI, BI and LI are returned.

Algorithm 3: ***Querying phase***

input : Dataset **D** query record **q** number of attributes n; minHash function **H**; encoding functions $\mathbf{E}_i$ and similarity functions $\mathbf{S}_i$ for $i = 1...n$; indexes **SI**, **BI** and **LI**.
output: Ranked match list **M**

```
1  Initialise M = []
   // First stage
2  Sig = H(q)
3  for sig ∈ Sig do
4      if sig ∉ LI then
5          Initialise bk = []
6          Append q.0 to bk
7          LI[sig] = bk
8      else
9          bk = LI[sig]
           // Second stage, pair-wise comparisons
10         for r.0 ∈ bk do
11             if r.0 ∉ M then
12                 Retrieve r from D using r.0
13                 Initialise s = 0
14                 for i = 1...n do
15                     if q.i ∉ SI then
16                         Insert(q, n, E, S, SI, BI, LI)
17                     sl = SI[q.i]
18                     s = s + sl[r.i]
19                 Append (r.0, s) to M
20         Append q.0 to bk
21         LI[sig] = bk
22     Sort M according to similarity
```

3.3.2 Querying Phase

At querying phase, the main aim is to return the most similar records that match with a query. As mentioned before, in many scenarios, queries are required to be processed in real-time. In the LSI, real-time querying is implemented using the three indexes built in the building phase. Also, similar to the extended similarity-aware inverted indexing proposed by Ramadan et al. (2013), querying in the LSI is performed in a dynamic manner, which means every single query is regarded as a new record and is used to enrich the three indexes. The main idea of the LSI querying is to use minHash to filter out low similarity records. Thus, rather than using all the records, only the records that share the same minHash signatures with the query are considered as candidate matches for pair-wise comparisons. Because the minHash filtered out most non-matches, an enormous number of unnecessary comparisons are avoided. For the necessary comparisons, since previously appeared attribute values are all indexed and their pair-wise similarities are stored in the SI, we can get their similarity values directly rather than on-line calculation (Christen et al. 2009). Because retrieving similarities from the SI is computationally cheaper than on-line comparisons, the querying is much faster.

The querying phase is briefly described in Algorithm 3 where a query record is denoted by **q**. **q** is firstly processed using minHash to obtain its minHash signatures (denoted as **Sig**). If a minHash signature corresponds to a empty block in the LI, the query record's identifier **q**.0 will be added to the empty block. If the block is not empty, records that have been hashed into the same block will be retrieved from the LI and considered as candidate matches. Next, the query **q** will be compared to every single candidate record. The comparison is done through comparing each attribute value of **q** and the candidate record **r** accordingly. Since the similarities are pre-calculated and stored in the SI, they can be retrieved directly. However, there

are cases that the attribute values of **q** are not previously indexed and cannot be found in the SI. For such cases, **q** is treated as a whole new record and the insertion function used in the building phase is called to insert it into indexes. As the querying is performed in a dynamic environment where queries are also considered as new records, **q**'s identifier is inserted into the LI for future querying.

[**Example 6**] Suppose r_5 in Table 1 is a query record, and the querying is performed based on the three indexes shown in Figure 1. In the work of Christen et al. (2009) and Ramadan et al. (2013), r_5 is compared with both r_2 and r_4 because they share the same Double-Metaphone encoding "hrst". In the LSI, r_5 is no longer compared with the noisy record r_4 because they are in different blocks in the LI (Figure 1). Thus, the number of comparisons can be decreased.

The overall similarity between two records is the sum of the similarities of all the attribute values of two records. The candidate result records are ranked based on their overall similarity values. We can set a similarity threshold to return those candidate results that have similarities higher than the threshold as the query results. Alternatively, we can select top N highly ranked records as query results.

4 Experiment

4.1 Dataset

To evaluate the approach, we conducted experiments on North Carolina Voter Registration Dataset. This dataset is a large real-world voter registration database from North Carolina (NC) in the USA (*North Carolina State Board of Elections: NC voter registration database* Last accessed 11 December 2012). We downloaded this database every two months since October 2011 until December 2012. This data set contains the names, addresses, and ages of more than 2.4 million voters. The attributes used in our experiments are: first name, last name, city, and zip code. The entity identification is the unique voter registration number. This data set contains 2,567,642 records. There are 263,974 individuals (identified by their voter registration numbers) with two records, 15,093 with three records, and 662 with four records. Examination of the record sets for individuals with multiple records shows that many of the changes in the first name attribute contain nicknames and small typographical mistakes. The changes in last name and address attributes are mostly real changes that occur when people get married or move address.

4.2 Evaluation Approaches

To evaluate the effectiveness of the proposed approximate blocking approach, we employ the commonly used Recall, Memory Cost and Query Time to measure the effectiveness and efficiency of the whole real-time top-N entity resolution approach. We divided each dataset into training (i.e., building) and test (i.e., query) set. Each test dataset contains 50% of the whole dataset. For each test query record, the entity resolution approach will generate a list of ordered result records. The top N records (with the highest rank scores) will be selected as the query results. If a record in the results list has the same entity identification as the test query record, then this record is counted as a hit (i.e., an estimated true match). The recall value is calculated as the ratio of the total number of candidate recrods of all the test queries to the total number of true matches in the test query set. We compared the performance produced by the following approaches:

- ***LSI***. This is the proposed two-stage similarity-aware indexing approach. It contains two stages, at the first stage, we use locality sensitive hashing to filter out records with low similarities for the purpose of decreasing the number of comparisons. Then, at the second stage, we pre-calculating the comparison similarities of the attribute values to further decrease the query time.
- ***LSH***. This is the Locality Sensitive Hashing approach. It generates l length-k signatures for each data record. MinHash is used to generate the signatures of each record. This is an improved locality sensitive hashing approach that uses dynamic collision counting (Gan et al. 2012) in real-time scenario. Work (Gan et al. 2012) uses a base of length-1 basic hash functions to represent each data record. The similar data records are ranked based on the dynamic collision counting number with the query record. As blocks with length-1 signatures usually have very large blocks for large-scale datasets, LSH uses length-k signatures ($k > 1$) rather than length-1 ones.
- ***SAI***.This is the Similarity-Aware Indexing approach for real-time entity resolution discussed in (Ramadan et al. 2013). It pre-calculated the similarity of each record value pairs of the same encoding block to decrease the number comparisons in real-time querying.

The above techniques are all built for dynamic indexing where queries are regarded as new records and are inserted into indexes for future quering. All the techniques are implemented using Python (version 2.7.3). Experiments are ran on a server with 128 GBytes of main memory and four 6-core 64-bit Intel Xeon CPUs running at 2.4 GHz.

4.3 Parameter setting

For the encoding (blocking) functions, the Double-Metaphone technique was used for the first three attributes (first name, last name, and suburb), while the last 4 digits were used for the zip code attribute. For the string comparison functions, the Winkler function was used for the first three attributes, while for the zip code the similarity was calculated by counting the number of matching digits divided by the length of the zip code. In order to simulate an intensive query environment, 50% data records of each dataset is used for indexes building and the other 50% are used for indexes querying.

The number of hash functions and number of bits in each band are crucial parameters, they together control the Jaccard similarity threshold of the minHash filter by tuning the logic combination of "AND" and "OR" (Gan et al. 2012). Dividing the number of hash functions by the number of bits, we get the number of minHash signatures for a record, which is also the number of buckets the record will be assigned to in the LI. Normally, a big number of minHash signatures leads to larger memory usage and more pairwise comparisons but better query accuracy, while a small number of minHash signatures leads to less

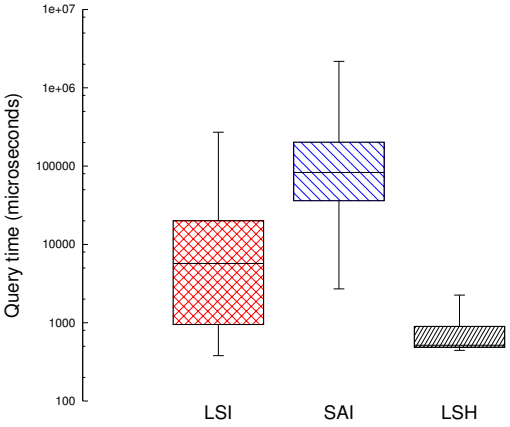


Figure 2: Summary of query time distribution, y axis is the query time in logarithmic scale $N = 100$. (boxplot with whiskers with maximum 1.5 IQR, outliers are not plotted).

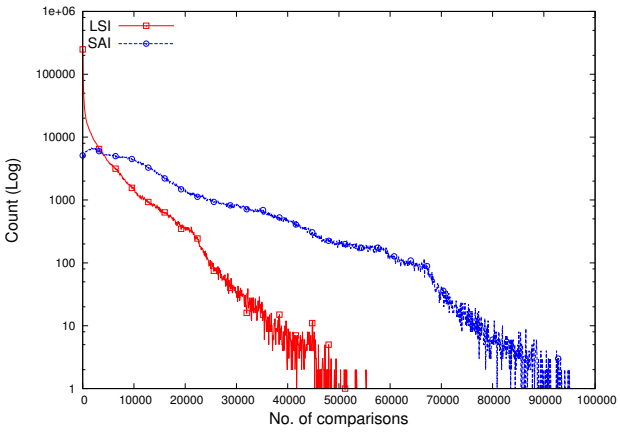


Figure 3: Distribution of the number of comparisons, $N = 100$, y axis is the count of queries in logarithmic scale and x axis is the number of comparisons. (LSH does not involve pair-wise comparisons, therefore it is not included in this figure).

query time, less memory usage and less accuracy. Different parameters can be chosen based on different scenarios. After extensive experiments, we set $k = 4$, $l = 15$ for LSH and LSI.

5 Results and Discussions

The experimental results show that the LSI's average processing time for a single query is 13.67 milliseconds, which is almost 10 times faster that the SAI (Figure 2). The improvement can be explained by decrease in the number of pair-wise comparisons as shown in Figure 3. Although the LSH is the fastest in terms of query processing, its recall is relatively low: less than 0.6 while $N = 50$. The LSI shows a good recall of around 0.7 when a small number of query results are returned. If a larger N is allowed, the recall of the SAI increases and surpasses the LSI at $N = 27$ (Figure 4). Consequently, the LSI requires more time for building indexes and more memory for storing indexes (Figure 5 and 6).

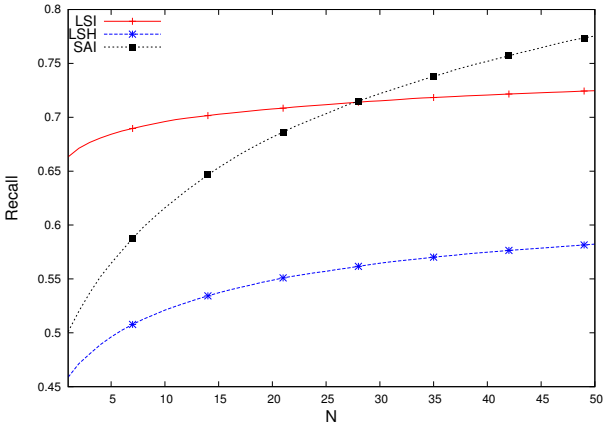


Figure 4: Top N recall results ($N = 1, ..., 50$).

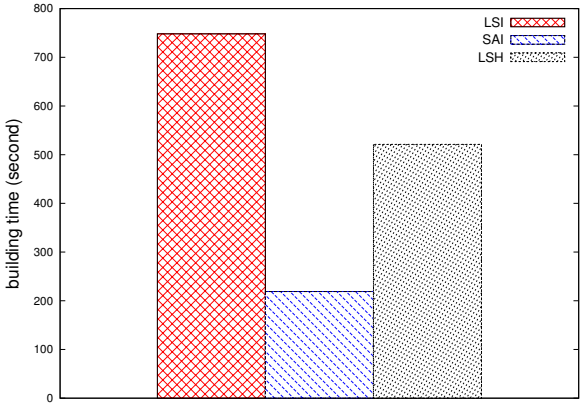


Figure 5: Building time

Discussions:

1) Query time

The distribution of query time is summarised in Figure 2 using boxplot with the range of maximum 1.5 IQR. IQR, shorted for interquartile range, equals to the difference between the upper and lower quartiles, i.e. $IQR = Q_3 - Q_1$.

We can see that the LSI approach performs better than the SAI in this aspect. The median query time for the LSI is 5.71 milliseconds, which is more than 10 times faster than the 83.03 milliseconds of the SAI. Also, as expected, the query time of the LSI is longer than that of the LSH. This is because the LSH only gives a approximate result for queries and does not involve pair-wise comparisons.

The distribution of number of comparisons can be used to explain the improvement of the LSI in query time. Processing time is much faster for queries that requires little comparison times and much slower for queries that need to be compared for many times. As shown in Figure 3, a significant number of queries in the two-stage approach are distributed in the lower range of the number of comparisons. 90% of the total queries are in the range of less than 1,000 comparison times. This is because dissimilar records are filtered out based on Jaccard similarity using minHash in the first stage, the number of candidate matches are relatively small and thus less pair-wise comparisons are needed. For many queries, pair-wise comparisons are not even needed because no candidate matches are found for them in the first stage (i.e., their minHash signatures are not found in the LI). There are

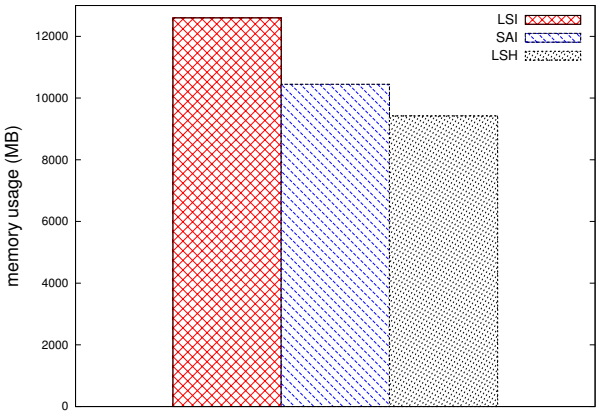


Figure 6: Memory usage, $N = 100$.

also a small proportion of queries that requires massive numbers of comparisons. This is because these queries' minHash signatures direct to large blocks in the LI, so a large number of candidate records are found for them in the first stage. As a result, the distribution of the LSI's query time shows a big range as seen in Figure 2

Comparatively, for the SAI, the number of comparisons is distributed more evenly and the maximum number of comparisons reaches almost 100,000. In the SAI, a query's comparison number totally depends on whether or not the Double-Metaphone encodings of the query's attribute values is common in the dataset. If the attribute values direct to large blocks in the BI, the query will be compared with a lot of candidate records, and vice versa. Although blocks in the BI of the SAI are of a variety of sizes, the number of Double-Metaphone encodings are limited, which makes most blocks in the BI huge. As a result, because most queries in the SAI are compared for much more time than those in the LSI, queries are processed slower in the SAI.

2) Recall

The recall distribution is given in Figure 4. As it shows, while the number of query results (i.e., N) is less than 27, the LSI's recall is higher than the SAI's. Starting from 0.66 while $N = 1$, the LSI's recall increases and slowly stabilised at 0.72. The SAI's recall grows steeply from 0.47 to 0.76 and surpasses the LSI at $N = 27$. The recall of the LSH is relatively low: less than 0.6 while $N = 50$.

In the LSH, the candidate data records are ranked based on the collision counting number with the query record. Because approximate comparison functions such as Winkler are not used in this approach, it fails to capture the spelling variations of attribute values and results in missing this type of true match records.

In the SAI, records have the same Double-Metaphone encodings with the query are all considered as candidate records and compared with the query using the Winkler function. Because attribute values with the same Double-Metaphone encoding often share common characters, they are likely to have high Winkler similarities and cannot be differentiated via similarity ranking. The top ranked records can be false matches as they may have analogical similarities. Therefore, when only a small N is allowed, false matches are likely to be included, which leads to low recalls. However, as N increases, more true matches are included than false match and the recall increases sublinearly.

The situation of the SAI does not happen to the LSI because most false matches are already eliminated by minHash based on their Jaccard similarities at the first stage, which ensures query results are of high quality while N is small. At the same time, some true matches with relatively low Jaccard similarities are also eliminated by minHash at the first stage. Consequently, the recall of the LSI grows slowly and soon settles at around 0.72 while the SAI's recall grows all the way to 0.76. So, while a small number of query results is required, the LSI outperforms the SAI in terms of recall. But if a big number of query results is acceptable, the SAI provides a better matching recall.

3) Building time and memory usage

The building times of the tree approach are shown by histograms in Figure 5. As expected, the LSI takes much longer time for building indexes. 748 seconds are used by the LSI to build indexes, which is 3 times longer than the SAI and 5 times longer than LSH. The main reason for the difference is the introduction of the LI in the LSI. The experiment setting is 60 hash functions with 4 bits in a band, which means each record is "hashed" into 15 buckets in the LI. As a result, the LI becomes the largest indexes and thus takes much longer time to build. Considering building is done off-line, the increase in building time does not have a big impact on real-time entity resolution scenarios.

While the LSI processes queries faster, it consumes more memory than other two techniques as shown in Figure 6. The LSI used more than 12,598 MB memory which is 2,158 MB more than the SAI and 3,179 MB more than the LSH. Similar to building time. the large LI plays an big part in the large memory usage of the LSI too. Additionally, for the purpose of avoiding signature collisions, minHash signatures are often large integer numbers. Storing millions of large integers in the LI also consumes a lot of memory.

6 Conclusion and Future Work

In this paper, a two-stage similarity-aware indexing approach named LSI has been presented for large-scale real-time entity resolution. LSI firstly filter out records with low similarities using locality sensitive hashing, and then pre-calculating the comparison similarities of the attribute values to further decrease the query time.

This approach is evaluated experimentally on a large-scale datasets taken from a real-world database. The experimental results demonstrated the effectiveness of the proposed approach.

Like other similarity-aware indexing techniques, the two-stage similarity-aware indexing approach requires to store pre-calculated similarities in memory, which consumes a large proportion of memory as query records are being added to the indexes continuously. Improving upon the memory consumption by adopting other indexing techniques such as sorted neighbourhood indexing is one of the future research directions of this approach. Additionally, exploring the possibility of applying this approach to other application areas such as real-time recommender system is another direction for future work.

References

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* 'Proceedings of the 14th international conference on World Wide Web', ACM, pp. 651–660.

Baxter, R., Christen, P. & Churches, T. (2003), 'A comparison of fast blocking methods for record linkage', *ACM SIGKDD Workshop on Data Cleaning, Record Linkage and Object Consolidation, pages 2527, Washington DC* .

Broder, A. Z. (1997), On the resemblance and containment of documents, *in* 'Compression and Complexity of Sequences 1997. Proceedings', IEEE, pp. 21–29.

Christen, P. (2012), 'A survey of indexing techniques for scalable record linkage and deduplication', *Knowledge and Data Engineering, IEEE Transactions on* **24**(9), 1537–1555.

Christen, P. & Gayler, R. (2008), 'Towards scalable real-time entity resolution using a similarity-aware inverted index approach', *AusDM '08 Proceedings of the 7th Australasian Data Mining Conference* .

Christen, P., Gayler, R. & Hawking, D. (2009), Similarity-aware indexing for real-time entity resolution, *in* 'Proceedings of the 18th ACM conference on Information and knowledge management', ACM, pp. 1565–1568.

Das Sarma, A., Jain, A., Machanavajjhala, A. & Bohannon, P. (2012), An automatic blocking mechanism for large-scale de-duplication tasks, *in* 'Proceedings of the 21st ACM international conference on Information and knowledge management', ACM, pp. 1055–1064.

Dasgupta, A., Kumar, R. & Sarlós, T. (2011), Fast locality-sensitive hashing, *in* 'Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 1073–1081.

Dong, W., Wang, Z., Josephson, W., Charikar, M. & Li, K. (2008), Modeling lsh for performance tuning, *in* 'Proceedings of the 17th ACM conference on Information and knowledge management', ACM, pp. 669–678.

Elmagarmid, A. K., Ipeirotis, P. G. & Verykios, V. S. (2007), 'Duplicate record detection: A survey', *Knowledge and Data Engineering, IEEE Transactions on* **19**(1), 1–16.

Gan, J., Feng, J., Fang, Q. & Ng, W. (2012), Locality-sensitive hashing scheme based on dynamic collision counting, *in* 'Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data', ACM, pp. 541–552.

Gionis, A., Indyk, P., Motwani, R. et al. (1999), Similarity search in high dimensions via hashing, *in* 'VLDB', Vol. 99, pp. 518–529.

Hernandez, M. A. & Stolfo, S. J. (1995), 'The merge/purge problem for large databases', *ACM SIGMOD95, San Jose* .

Ioffe, S. (2010), Improved consistent sampling, weighted minhash and l1 sketching, *in* 'Data Mining (ICDM), 2010 IEEE 10th International Conference on', IEEE, pp. 246–255.

Kim, H.-s. & Lee, D. (2010), Harra: fast iterative hashed record linkage for large-scale data collections, *in* 'Proceedings of the 13th International Conference on Extending Database Technology', ACM, pp. 525–536.

Lange, D. & Naumann, F. (2011), Efficient similarity search: arbitrary similarity measures, arbitrary composition, *in* 'Proceedings of the 20th ACM international conference on Information and knowledge management', ACM, pp. 1679–1688.

Lange, D. & Naumann, F. (2012), 'Cost-aware query planning for similarity search', *Information Systems* .

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.

Lv, Q., Josephson, W., Wang, Z., Charikar, M. & Li, K. (2007), Multi-probe lsh: efficient indexing for high-dimensional similarity search, *in* 'Proceedings of the 33rd international conference on Very large data bases', VLDB Endowment, pp. 950–961.

Michelson, M. & Knoblock, C. A. (2006), Learning blocking schemes for record linkage, *in* 'Proceedings of the National Conference on Artifiical Intelligence', Vol. 21, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, p. 440.

North Carolina State Board of Elections: NC voter registration database (Last accessed 11 December 2012).
URL: *ftp://www.app.sboe.state.nc.us/*

Ramadan, B., Christen, P., Liang, H., Gayler, R. W. & Hawking, D. (2013), Dynamic similarity-aware inverted indexing for real-time entity resolution, *in* 'Trends and Applications in Knowledge Discovery and Data Mining', Springer, pp. 47–58.

Slaney, M. & Casey, M. (2008), 'Locality-sensitive hashing for finding nearest neighbors [lecture notes]', *Signal Processing Magazine, IEEE* **25**(2), 128–131.

Sood, S. & Loguinov, D. (2011), Probabilistic near-duplicate detection using simhash, *in* 'Proceedings of the 20th ACM international conference on Information and knowledge management', ACM, pp. 1117–1126.

Yan, S., Lee, D., Kan, M. Y. & Giles, L. C. (2007), 'Adaptive sorted neigh-borhood methods for efficient record linkage', *ACM/IEEE-CS joint conference on Digital Libraries* .

To Learn or to Rule: Two Approaches for Extracting Geographical Information from Unstructured Text

Philipp Katz **Alexander Schill**

Dresden University of Technology
Faculty of Computer Science
Institute of Systems Architecture
01072 Dresden, Germany
Email: philipp.katz@tu-dresden.de

Abstract

Geographical data plays an important role on the Web: recent search engine statistics regularly confirm that a growing number of search queries have a locale context or contain terms referring to locations. Assessing geographical relevance for Web pages and text documents requires information extraction techniques for recognizing and disambiguating geographical entities from unstructured text. We present a new corpus for evaluation purposes, which we make publicly available for research, describe two approaches for extracting geographical entities from English text—one based on heuristics, the other relying on machine learning techniques—and perform an extensive discussion of those two approaches. Furthermore, we compare our approach to other publicly available location extraction services. Our results show, that the presented approaches outperform current state of the art systems.

Keywords: Toponym Resolution, Toponym Recognition, Toponym Disambiguation, Machine Learning, Feature Mining, Dataset

1 Introduction

According to recent statistics, between 30 and 40 % of the users' queries at Google are now related to physical places (Parsons 2012). Therefore, correctly recognizing and extracting geographic information from unstructured text can be considered a crucial step for offering more appropriate answers to users' information needs. Domains where geographic information plays an important role are, for example, the daily news; methods for searching and organizing news articles can greatly benefit from place information, as it is one of the fundamental parts of the *Five Ws* (Who, What, When, Where, Why) employed in journalism to describe events. As a further example, consider advertising: In *Geotargeting*, geographic information extracted from available data can be used to deliver more individual and context relevant content to the user.

The roots of geographical information extraction lie in Named Entity Recognition (NER) as it was defined by the Message Understanding Conference 6 in 1996 (Grishman & Sundheim 1996). However, general NER usually neglects spatial properties in favor of recognizing a broad range of different entity types (e. g. in MUC-6, the types "Organization", "Person", "Location", "Date", "Time", "Money", "Percent" were used). Later, dedicated approaches focussed explicitly on extracting geographic data from text and associating extracted location references with models of the real world by providing spatial and/or topological properties such as coordinates or administrative/part-of relations.

The task of extracting geographical information from text can be divided into the following three disciplines: *Toponym Recognition* (TR), *Toponym Disambiguation*[1] (TD) and *Scope Resolution* (SR). TR identifies and marks occurring toponyms in a text and is therefore closely related to NER. A central challenge lies in the so called "geo/non-geo ambiguity" (Amitay et al. 2004). Considering the sentence "Mary is in Turkmenistan.", it is unclear without further background information, whether "Mary" refers to a person or to the city which is located in the south east of the country. The process of TD associates identified toponyms with entries in a database serving as a so called gazetteer. Here, we face the problem of the "geo/geo ambiguity" (Amitay et al. 2004). Consider the sentence "San Antonio is a place in California.". While looking up the term "San Antonio" in a gazetteer would yield matches all over the world, using the determiner "California" as filter, the number of potential places can be greatly reduced, although there are still multiple places called "San Antonio" in California.

SR on the other hand identifies a spatial scope for a document as a whole, summarizing all geographical evidence to one appropriate abstraction. While the presented approach can be further used as foundation for SR, the focus of this work is TR and TD.

During our research, we found that existing datasets for geographic information extraction are usually not publicly available, limiting the possibilities to compare different approaches with each other. Thus, the first contribution of this work is a novel dataset with NER-style type annotations for locations and associated geo coordinates, which we make freely available for research purposes on the research platform Areca. The creation and properties of this dataset, called "TUD-Loc-2013" hereupon, will be described in detail in Section 3. As a second contribution, we present two new approaches for combined TR and TD: A heuristic approach relying on several rules which will be described in Section 4.2, and an approach using machine learning (ML) techniques, based on a plethora of extracted features, which will be discussed in detail in Section 4.3. To the current

[1] Others, such as Leidner (2006), Lieberman & Samet (2012) refer to this task as "Toponym Resolving" or "Resolution".

state of the art, none of the existing approaches as presented in Section 2 has applied ML with such an extensive feature set for this task. Section 5 presents our gazetteer, which is aggregated from multiple freely available location sources. In Section 6, we outline our experiments for fine-tuning the approaches. In particular, we review the features which serve as input for the machine learning approach and determine, which of them are actually valuable for the task. In Section 7 we compare our approaches to other services for extracting locations from unstructured text.

2 Related Work

This section describes related work for TR and TD. General all-purpose NER approaches which perform only TR, but do not associate recognized toponyms to geographic representations are not considered here, neither do we present approaches for SR such as Andogah (2010) or Wing & Baldridge (2011).

One of the first notable works in this area can be attributed to Smith & Crane (2001), who did TR and TD for a digital library with historical content. After a rule-based identification and filtering of toponym candidates, their approach calculates a centroid coordinate for all location candidates and continuously elmiminates those which are more than a specified distance away from this centroid.

Li et al. (2002, 2003) build a weighted graph of all location candidates in a text. The edge weights are determined using a set of rules which rely on topological properties between the represented locations beside some intrinsic text metrics. Using Kruskal's algorithm, a Maximum Weight Spanning Tree is calculated from which the final disambiguation is derived.

Rauch et al. (2003) describe a confidence-based mechanism which, on the one hand relies on intrinsic textual properties, such as the proximity between two toponyms within the text and the actual geographic proximities of their potential locations. On the other hand, for locations with same names, higher confidence values are assigned to those with larger population figures.

Smith & Mann (2003) employ a Naïve Bayes classifier for a simplified TD task, where the aim is to recover the correct U. S. state or country for a given text. The classifier is trained using phrases from texts, which disambiguate place names by giving explicit cues, such as in "[. . .] Nashville, Tennessee [. . .]", and relies on text features (unfortunately, the work gives no deeper information about the feature types they use, e. g. n-grams, tokens, etc.). However, their results yield in only minimal improvements over a baseline which simply assumes the most frequently occurring location.

Similar to Rauch et al. (2003), Amitay et al. (2004) assign confidence values to locations based on some heuristics. Additionally, they address the problem of geo/non-geo ambiguities with a large curated corpus of over 1 million Web pages. Location names from their gazetteer, which do not occur as proper nouns in the corpus frequently, or where the number of mentions in the corpus is strongly disproportional to the population of the place, are considered as non-locations, unless explicit evidence is given in the text.

Leidner (2007) presents an algorithm for toponym resolution using a spatial minimality assumption. Besides the heuristic to resolve each toponym to a country, if such exists, and exploiting explicitly given disambiguations in text, a cross product is computed of all locations for remaining unresolved toponyms. From all potential combinations, each containing one potential location for the toponyms given in the text, the combination which spans the smallest area is selected.

Da Graça Martins (2008) use a set of manually created contextual rules for TR. Furthermore, an exclusion list is used to remove common terms which are not usually locations. The TD also makes use of the context rules and further applies a set of heuristics which exploit the topological relations between pairs of potential locations.

Buscaldi & Rosso (2008) employ classical methods for word-sense disambiguation and rely on WordNet to recognize toponyms. They note the poor coverage of WordNet in regards to geographical information compared to classical gazetteers.

Lieberman & Samet (2012) are the first to evaluate a machine learning-based TD approach. They present seven features, two of which are extracted from a so called "adaptive context". The adaptive context is characterized by a window breath and depth. The first denotes a context of a specified number of toponyms before and after the currently considered toponym, while the depth limits the number of potential location candidates to consider for each toponym. These two figures are motivated by the requirement to allow for a fast TD computation. Using labeled data, a classifier is trained, which carries out a classification for each potential location assignment to a toponym to be either correct or incorrect.

While most of the aforementioned works rely on heuristics, only the more recent approach of Lieberman & Samet (2012) successfully takes up the idea of a feature-based machine learning technique. However, they rely on a comparatively small feature set and only employ them for the TD phase. In this work, in contrast, we are going present a comprehensive set of different features, which we will then evaluate for a machine learning-based, combined TR and TD approach. In addition, we will draw a comparison with our new heuristic-based method.

3 Datasets

Despite the growing attention of the research community during the last years in toponym recognition, the lack of publicly available datasets makes it difficult to compare different approaches to each other. In the following, we will shortly give an overview over datasets which have been employed in the past and outline our motivations to create the novel, freely available "TUD-Loc-2013" dataset.

3.1 Existing Datasets

"GeoSemCor", as presented by Buscaldi & Rosso (2008), is a freely available dataset, where Toponyms have been annotated with WordNet senses. It contains only annotations for the relatively popular toponyms which exist in WordNet and has no geographical referents. The same applies to "CLIR-WSD"[2].

"TR-ConLL", which was presented in Leidner (2006) is based on English news texts from the Reuters CoNLL corpus. It contains 946 documents with 6,980 toponym instances, of which 1,299 are unique. The dataset can be purchased from the author and costs 550 US$ for an academic license.

The "ACE 2005 English SpatialML Annotations"[3] consists of 428 documents from a broad range of different sources (news, blogs, newsgroups). The dataset is

[2] http://ixa2.si.ehu.es/clirwsd/

[3] http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2011T02

not publicly available; non-members of the Linguistic Data Consortium pay 500/1,000 US$.

The "Local-Global Lexicon" (LGL) corpus from Lieberman et al. (2010) was obtained from 78 newspapers with a mainly local focus, yielding in 588 documents. The sources were selected with an explicit focus on ambiguities (for example, containing the cities Paris in Texas, Tennesse, and Illinois). The dataset therefore seems well suited for evaluating location extraction in a local domain, but does obviously not represent realistic properties. LGL is not publically available, but the author kindly supplied us with the corpus. During a thorough inspection, however, we noticed several issues and inaccuracies concerning the annotations: Plenty of locations in the dataset have not been marked[4] or for many annotations, more specific locations exist[5]. Besides, demonyms and adjectives are marked as locations. We have seen, that these drawbacks favor extraction approaches with low recall and penalize approaches with very fine-grained toponym resolution. That is why we will not include LGL in our evaluation given in Section 7.

"TR-CLEF" and "TR-RNW" are manually anotated corpora used for evaluations in Andogah (2010). They are not published however and we never received a reply from the author to our inquiry.

3.2 TUD-Loc-2013

Motivated by our experiences as described in Section 3.1, we will describe the creation of "TUD-Loc-2013"; a novel dataset for evaluating TR and TD approaches. The dataset described here is published on the open research platform Areca[6] to increase transparency and to allow future research to be compared on this publicly available dataset.

The dataset consists of 152 English text documents retrieved from different URLs. An index file within the dataset package gives the original URLs from which the pages were obtained. We focused on content-oriented pages such as news and blog articles, but excluded start pages which combine multiple topics. The main text content was manually extracted, removing elements such as banners, navigation menus, comments, headers, and footers.

	Total		Unique	
Type	**#**	**%**	**#**	**%**
CONTINENT	72	1.89	6	0.43
COUNTRY	1,486	38.96	147	10.49
CITY	1,031	27.03	401	28.62
UNIT	242	6.35	131	9.35
REGION	139	3.64	83	5.92
LANDMARK	281	7.37	183	13.06
POI	454	11.90	355	25.34
STREET	55	1.44	45	3.21
STREETNR	37	0.97	33	2.36
ZIP	17	0.45	17	1.21
# All	3.814		1.401	

Table 1: Counts of annotations in TUD-Loc-2013

We initially defined ten location types for annotation as depicted in Table 1. We tried to make different location types clearly distinguishable and wanted to avoid a too broad type variety. While the first three of the given types should be self-explanatory, we want to stress the difference between `UNIT` and `REGION`; the first refers to administrative entities, such as federal states, counties or cities' districts (e. g. "California", "Bavaria", or "Manhattan"). The latter, `REGION`, on the other hand is used to designate areas without political or administrative meaning (e. g. "Midwest"). While locations annotated as `LANDMARK` refer to geographic entities, such as rivers, lakes, valleys, or mountains (e. g. "Rocky Mountains"), the type `POI`[7] indicates buildings (e. g. "Stanford University" or "Tahrir Square").

The annotation of the extracted texts was done manually in XML style. This means, that relevant parts of the text were surrounded by tags denoting the appropriate types. Additionally, we allowed the attribute `role="main"` once per document, indicating the document's geographic scope (relevant for SR tasks, see Section 1). The following paragraph shows an example snippet from the dataset:

```
Tiny <LANDMARK>Heir Island</LANDMARK> -- one of the
many isles that are scattered across County
<CITY>Cork</CITY>'s <LANDMARK>Roaring Water
Bay</LANDMARK> in <COUNTRY
role="main">Ireland</COUNTRY>'s southwest -- is one
of the country's go-to gourmet spots.  So you will
need to book months in advance to dine at <POI>Island
Cottage</POI>, a restaurant run by the
husband-and-wife team John Desmond and Ellmary Fenton.
[...]
```

In a second step, annotations were associated with actual locations. Through a dedicated Web-based annotation app, we allowed to query GeoNames[8] and—as a fallback—the Google Geocoding API[9] with the annotations' values. All results were displayed as markers on a map, including additional properties such as type, population, etc. and were then manually selected. As not all values can be found directly, we also provided the possibility to modify the queries (e. g. the term "Atlantic" needs to be corrected to "Atlantic Ocean" to give any results). Locations, which could not be found could be marked as "non resolved" explicitly. The result is a separate CSV file with pointers to the annotated text files (filename, running index and character offset of annotation), coordinates and source-specific identifiers. The following lines give an excerpt of the CSV file:

```
docId;idx;offset;latitude;longitude;sourceId
text1.txt;0;0;53.00000;-8.00000;geonames:2963597
text1.txt;1;28;53.00000;-8.00000;geonames:2963597
text1.txt;2;399;51.49475;-9.43960;google
text1.txt;3;469;51.96667;-8.58333;geonames:2965139
text1.txt;4;476;;;
text1.txt;5;497;53.00000;-8.00000;geonames:2963597
text1.txt;6;619;;;
text1.txt;7;755;51.51077;-9.42505;google [...]
```

From the given 3,814 annotations, 3,452 were manually disambiguated (90.51 %). The remaining non-disambiguated locations are mostly of type `POI`; here, only 50,88 % of the annotations could be assigned with coordinates. This is due to the fact, that the dataset contains several little-known locations such as restaurants, etc. which could not be found in the considered databases. Figure 1 shows the distribution of the disambiguated locations on a map.

[4] In doc. 38765806 for example, 20 annotations are present in the dataset, but we counted 44 toponyms.

[5] In doc. 38543488 for example, in the phrase "[...] building permit for the new Woodstock General Hospital [...]", the term "Woodstock" is marked, but we feel that "Woodstock General Hospital" is the more accurate location.

[6] `http://areca.co/21/TUD-Loc-2013-location-extraction-and-toponym-disambiguation-dataset`

[7] Point of Interest

[8] `http://www.geonames.org`

[9] `https://developers.google.com/maps/documentation/geocoding/`

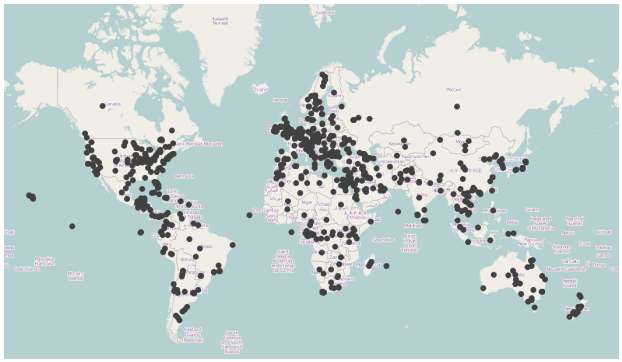


Figure 1: Coordinate distribution in TUD-Loc-2013

The final dataset is split in the three disjoint sets *training* (40 %), *validation* (20 %) and *test* (40 %).

4 Approaches

In the following Sections 4.2 and 4.3, our two strategies for TR and TD are described. The common processing steps to both of the strategies, which include candidate extraction, filtering and preprocessing, are described as follows.

4.1 Preprocessing

In contrast to various other approaches, which rely heavily on a full-fledged preprocessing pipeline, including PoS tagging, deep parsing or NER, we only use very basic mechanisms for TR as described below. We focus on correctly detecting entity candidates of all types (i. e. also potential non-locations) and only filter out those ones, where we can be sure that they do not represent a toponym. Our assumption is, that wrongly classified candidates can be better removed in the TD phase, where we can apply more knowledge such as the gazetteer and information about further candidates in the document. All kinds of lingustic processing on the other hand, especially NER, are strongly domain-specific and introduce additional chances for errors under suboptimal conditions.

Candidate extraction takes place using a rule-based tagger, which marks sequences of capitalized expressions in the text. Several exceptions are applied to correctly annotate term spans containing lowercased prepositions or special characters (such as in "United States of America", "Rue de Rivoli", or "Grand Tradition Estate & Gardens"). This approach guarantees high recall and was successfully applied by Urbansky (2012).

A problem which arises from the rule-based tagging is that tokens at the beginning of sentences are always considered as candidates because of their capitalization. Given the sentence "Tiny Heir Island is one of the country's go-to gourmet spots.", our tagger extracts the candidate "Tiny Heir Island", although "Tiny" represents an attribute for the actual location. A pre-generated case dictionary is used to remove or correct candidates at sentence beginnings. The idea has been described by Millan et al. (2008). The case dictionary[10] consists of a list of tokens with their occurrence frequencies as uppercase and lowercase variant. In case, a token occurs clearly more frequent in lowercase form, we remove it or correct the candidate's offset (so that in the given example, we would end up with the correct form "Heir Island").

Our experience shows, that many incorrectly extracted locations are actually person names. Therefore, we remove *sure*-negative candidates, using a well curated set of person-centric prefixes such as "Mr.", "Minister", "Officer", etc. in a first step. In a second pass, we classify *potentially* negative candidates using candidates' text contexts. Contexts are surrounding tokens before or after a candidate and give clues about its type. For example, in the sentence "Georgia attended the conference", we can conclude from the suffix "attended", that the entity is most likely a person, whereas in the case of "Georgia president concedes election defeat", the suffix "president" gives a strong clue, that the preceding candidate is a location. We have accumulated a massive amount of location and person specific texts building on the foundations as described in Urbansky (2012) to extract a corpus of characteristic contexts for both types. For a list of 800 manually compiled seed entities for each type "person" and "location", we queried Bing[11] and obtained at most 100 URLs per seed, yielding in 29,642 HTML pages for persons and 33,454 pages for locations. We tried to extract the main content block of each page using the Palladian toolkit (Urbansky et al. 2012) and filtered texts under 100 characters and short fragments within texts. The final context dataset consists fo 126,377 person entities and 184,841 location entities. Contexts for persons serve as negative, contexts for locations as positive indicators to build a context dictionary which we use for lookup during classification. To avoid misclassifications and thus decrease recall, the dictionary is created using a very conservative strategy: Only those contexts, where probability is over 90 % of being one of either type are incorporated into the dictionary[12]. We experimentally evaluated different context token sizes and a fuzzy matching (whether a term occurs within a window around the entity candidate), but found, that a fixed lookbehind and ahead of one token provides the most reliable classification results. We assume a "one sense per discourse" (Gale et al. 1992) and thus consolidate context classifications of identical entities within the document.

In contrast to the first filtering pass, we make no instant decision whether to remove the candidate here. The rationale behind this deferred commitment strategy is, that a distinction between location and person type is not perfect and complicated by common figures of speech such as metonymy. For example, for the phrase "U.S. says Rwanda aids Congo rebels", our context classifier would cleary label "U.S." as being a person. By postponing the decision whether to drop or to keep the candidate in question to the TD phase, we can make use of the gazetteer information to apply appropriate exceptions in case of prominent locations such as countries or capitals.

The last step in the preprocessing phase is the lookup in our gazetteer (see Section 5). For each unique annotation value a_n, we query our database to retrieve a set L_n of potential location candidates. The TD strategies as described below take a list of all annotations $A = \{a_1, \ldots, a_n\}$ with their corresponding locations $AL = \{L_1, \ldots, L_n\}$. Then, per annotation a_n, either one location candidate $l \in L_n$ is selected, or the annotation is discarded as being a "non-location" by the TD.

[10] We created our case dictionary from English Wikipedia articles, consisting of approximately 91,000 tokens.

[11] http://www.bing.com

[12] The context dictionary used within this work consists of 318 prefix and suffix contexts for the type "person" or "location".

4.2 Heuristic TD

As a first step, the heuristic eliminates unlikely annotations by applying the following rules: Those annotations, which were marked as being likely of type "person" are removed, in case they do not have a location candidate which is of type `CONTINENT` or `COUNTRY`, or where the population is above a *unlikelyPopulationThreshold*.

Subsequently, the approach makes use of a concept which we call "anchor locations"[13]. The underlying idea is to first only extract those locations, where we can guarantee high precision and use them as reference points in a second extraction step. The mechanism for extracting anchor locations is outlined in Figure 2. First, we assume those locations as anchors, which are either of type `CONTINENT` or `COUNTRY` or those which exceed a high population count as specified with *anchorPopulationThreshold*. To extract further anchor locations, we employ the following criteria: We create groups of locations with equal names and determine the largest distance[14] between each pair in the group. In case, the largest distance is below *sameDistanceThreshold* (which is set to a two-digit value in kilometers), we suppose multiple location candidates to denote the same place[15]. Locations complying to this condition and having either a population of more than *lowerPopulationThreshold*, or a distinctive name consisting of more than one token (e. g. "Santa Catarina Federal University") as defined by *tokenThreshold* are added to the set of anchor locations. We have seen, that the probability for geo/non-geo ambiguities strongly decreases for terms with two or more tokens.

```
funct getAnchors(L) ≡
  Anchors := {}
  for l in L do
    if type(l) ∈ {CONTINENT, COUNTRY} ∨
      population(l) ≥ anchorPopulationThreshold
      then Anchors ← l; fi
  end
  for g in groupByName(L) do
    if largestDistance(g) ≤ sameDistanceThreshold
      then
          l := getBiggestLocation(g);
          p := population(l);
          t := |tokenize(name(l))|;
          if p ≥ lowerPopulationThreshold ∨
            t ≥ tokenThreshold
            then Anchors ← l; fi
    fi
  end
  return Anchors.
```

Figure 2: Algorithm for extracting anchor locations

In case we could not determine any anchor locations using the given strategy, we use a stepwise convergence approach comparable to a lasso, which is increasingly tightened. Figure 3 shows the progress, where we continuously remove the most outlying location from the center point of a given set. The idea is adopted from Smith & Crane (2001), but we stop the convergence, as soon as the maximum distance between any pair in the remaining set is below a specified *lassoDistanceThreshold*. We only take the locations remaining in the set as anchors, if it contains at least two differently named candidates. This way, we can avoid converges into the wrong direction. In case, still no anchor could be established, we simply select the location with the highest population from the candidate set as a fallback.

```
funct getLasso(L) ≡
  Lasso ← L
  while |Lasso| > 1 do
      maxDistance := 0;
      maxDistanceLocation := null;
      midpoint = midpoint(Lasso);
      for l in Lasso do
          distance := distance(midpoint, l);
          if distance > maxDistance
            then
                maxDistance := distance;
                maxDistanceLocation := l; fi
      end
      if maxDistance < lassoDistanceThreshold
        then break; fi
      Lasso := Lasso \ maxDistanceLocation;
  end
  if |groupByNames(Lasso)| ≤ 1
    then return ∅;
    else return Lasso; fi.
```

Figure 3: Algorithm for extracting lasso locations

The identified anchor locations are used as reference points in the final disambiguation phase. For all remaining location candidates, which are not in the set of anchor locations, we check the spatial distance between the candidate and all anchors. Locations either falling below a *anchorDistanceThreshold* or being child of a given anchor location and exceeding a *lowerPopulationThreshold* are added to the final result set. In case multiple location candidates with the same name fulfill the given criteria, we select the one with the biggest population, or—in case the locations are in a hierarchy—the deepest, i. e. the most specific one.

4.3 Machine Learning TD

The second approach is based on the findings of the heuristic. We have experienced, that adding more rules to further improve the approach described in Section 4.2 becomes increasingly complicated and bears the risk of overfitting the algorithm. We therefore present a more flexible approach in this section using machine learning mechanisms. It relies on a number of features and a classifier which is trained using manually annotated and disambiguated training documents. As initally outlined, the classifier is used to perform a binary classification for each location candidate for an annotation. The probability value assigned by the classifier is used to rank the location candidates. In case the probability exceeds a specified *probabilityThreshold*, we assign the highest ranked candidate location to the annotation, elsewise we discard all candidates, i. e. we discard the annotation as "non-location". Obviously, the probability threshold allows to adjust the results of the approach into a more precision- or recall-oriented direction.

Table 2 gives an overview over the multitude of features we extract for the classification. We give a general motivation for those features in the following and describe selected features in more detail. The

[13] Other approaches such as Rauch et al. (2003) or Lieberman et al. (2010) also use the term "anchor", their definitions are different, however.

[14] All distance calculations in this work use the haversine function (Sinnott 1984), which denotes the shortest distance between two points on an idealized sphere with a radius of $r = 6,371$ km.

[15] We have, for example, multiple entries for the location "Armstrong Atlantic State University" in our gazetteer, which is due to the fact, that multiple facilities exist. Still, they lie close together, so we treat the locations as one.

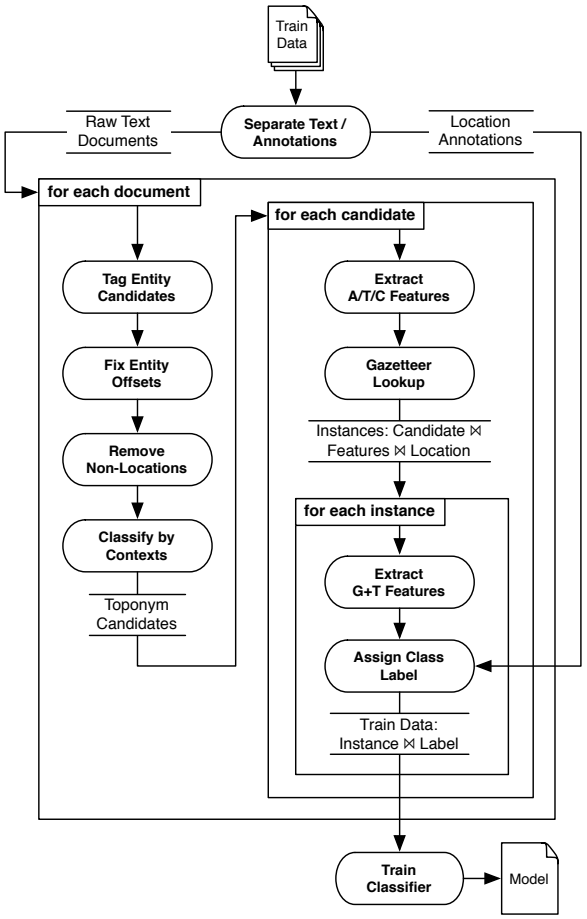
Feature Name	Type	Description
Annotation Features		
`numCharacters`	num.	Number of characters in name
`numTokens`	num.	Number of tokens in name
`acronym`	bin.	Name is acronym, e. g. "USA", "U.A.E.", etc.
`stopword`	bin.	Name is on stopword list, e. g. "Or"
`caseSignature`	nom.	Upper/lowercase signature, e. g. "Aa Aa" for "New York"
`containsMarker(M)`	bin.	Name contains marker token `M`, such as "city", "mountain", etc.
Text Features		
`count`	num.	Occurrence count in text
`frequency`	num.	Count, normalized by highest occurence count
Corpus Features		
`unlikelyCandidate`	bin.	Annotation was classified as being unlikely a location during preprocessing
Gazetteer Features		
`locationType`	nom.	Type of location (see Table 1)
`population`	num.	Population count of location
`populationMagnitude`	num.	Order of magnitude of population
`populationNorm`	num.	Population, normalized by highest population of location candidates
`hierarchyDepth`	num.	Depth of location in topologic hierarchy
`nameAmbiguity`	num.	Occurrence count of name in gazetteer, calculated as $1 / \mid sameNameLocations \mid$
`leaf`	bin.	Location has no child with same name
`nameDiversity`	num.	Diversity of alternative names for location, calculated as $1 / \mid namesForLocation \mid$
`geoDiversity`	num.	Spatial distribution of locations with given name
Text and Gazetteer Features		
`contains(p|a|s|c|d)`	bin.	Text contains parent/ancestor/sibling/child/descendant location as candidate
`num(a|s|c|d)`	num.	Number of ancestor/sibling/child/descendant location candidates in text
`numLocIn(R)`	num.	Number of location candidates in text within distance `R`
`distLoc(P)`	num.	Minimum distance to other locations with minimum population `P`
`populationIn(R)`	num.	Sum of population count of other locations within distance `R`
`locSentence(R)`	bin.	Location with maximum distance `R` occurs in same sentence
`uniqueLocIn(R)`	bin.	Location has a uniquely named location in maximum distance `R`

Legend: bin. = binary, nom. = nominal, num. = numeric feature

Table 2: Features for machine learning TD

types of features can be grouped in different categories describing the origin from which they were extracted. *Annotation Features* are directly extracted from the candidates' text values and describe simple string properties. The `caseSignature` describes the upper/lower case combination of a candidate. The motivation behind this feature is, that specific location types have characteristic upper/lowercase mixtures (such as "University of California", "Isle of Man" which case signature is "Aa a Aa"), whereas other capitalization variants such as "McDonald" or "InterCity" (case signature "AaAa") might indicate a non-location. `containsMarker(M)` can be determined for a predefined set of indicative tokens, which give a strong clue that the current candidate is a location, such as "city", "river", "mountain", "university" etc. *Text Features* are extracted by regarding the whole document's text. `count` and `frequency` denote, how often an annotation occurs within the text. The *Corpus Feature* `unlikelyCandidate` is determined using the contexts as described in Section 4.1. *Gazetteer Features* are retrieved from the location database. We incorporate the `locationType` so that the classifier can potentially adjust its decisions to different types of locations. The `nameAmbiguity` signifies the number of locations with the respective name in the database; the more locations exist, the bigger the chance for misclassifications. `geoDiversity` follows a similar motivation, but here we measure, how widespread locations with identical names are spatially; we consider the chance for a misclassification higher, in case potential candidates are scattered throughout the whole world. In contrast, the risk should be smaller, in case the potential candidates are close to each other. *Text and Gazetteer Features* combine properties from the text with properties from the database. The `contains` feature signifies, whether location candidates with a specific topologic relation such as "parent", "ancestor", etc. occur within the text. The `num` feature gives the counts of topologically related location candidates. Thus, we can indicate that a specific location is disambiguated through a superior instance within the text, such as "Houston, Texas", or we have an enumeration pattern with locations of similar types, such as in "Stuttgart, Frankfurt, Munich". The remaining features describe spatial proximities to other candidates and thus exploit the fact, that usually, spatially related locations occur together. For example, `numLocIn(R)` signifies the number of location candidates within a maximum distance `R` to the considered location candidate occurring in the text. Note, that some of the employed features can be parametrized and thus replicated and evaluated with different configurations, which we will discuss within Section 6.

To train our classifier, we depend on manually annotated and disambiguated training documents, such as the TUD-Loc-2013 as presented in Section 3. We also use the classifier to detect non-locations, this means, that all toponyms in the texts must be annotated, as the non-annotated entities are considered negative examples for training. Figure 4 shows the training process. First, the annotated training documents are split into their original text and a set of annotations. We then extract potential location candidates using the mechanisms as described in Section 4.1. In a first step, we extract annotation, text and corpus specific features. For each toponym candidate we perform a gazetteer lookup to retrieve potential location candidates and add gazetteer specific features. The combination of location candidate extracted from text, its features and the location information from the gazetteer forms an instance for the classifier. Each



Legend: ⋈ = join on candidate/location name, A/C/G/T = annotation/corpus/gazetteer/text features

Figure 4: Training process of the machine learning-based approach

instance is checked against the manually assigned location annotations from the training data. We mark those instances as positive samples, which have a corresponding location of same type, same name and a small distance. The remaining instances are marked as negative samples for training.

We make no limitations on the actual classification algorithm, as long as it supports numeric and nominal input features and provides a probability value with its classification output. Our implementation relies on a Bagging classifier (Breiman 1996) which creates multiple decision trees using bootstrap sampling of the training data. The classification is carried out by letting all decision trees vote and taking the portion of each result class as its probability. Bagging decision trees avoids the problem of overfitting and improves classification accuracy compared to a single tree.

The learned model can then be used for classification. The preprocessing and feature extraction steps are identical to those employed during the training phase. All instances are classified, resulting in probability value of being the correct location. Annotations, where all candidates' probability values are below a *probabilityThreshold* are discarded as "non-locations", for annotations where the threshold is exceeded, the candidate with the highest probability is taken as result.

4.4 Postprocessing

The postprocessing phase, which is carried out for the heuristic and the machine learning method as well, extracts location types which we do not cover through our gazetteer (yet): `STREET`, `STREETNR`, and `ZIP`. Our current approach is very rudimentary and involves great potential for future improvements. We start by extracting street names using a set of prefix and suffix rules (such as "*street", "*road", "rue *", etc.). We then try to match street numbers occurring before or after those street names. Similarly, we proceed for ZIP codes, which are searched right before or after location entities marked as `CITY`. A disambiguation of entities of the three types is not carried out currently.

5 Gazetteer

Our gazetteer has been aggregated from different sources and currently consists of circa 9.2 million locations. While other approaches employ only comparatively small gazetteer databases, our aim is to achieve a high recall from our gazetteer and guarantee precision through our algorithms. For each entry, we keep the following information: unique identifier, primary name, alternative names (a list of alternative names for the location, optionally including a language), a type (see Figure 1), latitude and longitude coordinates, population count if applicable and the topologic hierarchy (expressing the list of parent locations within the database, such as "Federal Republic of Germany → Europe → Earth").

While the major portion (8.5 million entries) of our data comes from GeoNames, we have further enriched our database from the following sources: The dataset from HotelsBase[16] provides about 500,000 hotels, protectedplanet.net[17] contributes about 200,000 protected areas.

To further complement our database, we extract locations from the English Wikipedia[18]. Using the Palladian toolkit's MediaWiki parser (Urbansky et al. 2012), we are able to extract 500,000 entries. Similar to universal information extraction approaches such as DBPedia[19], we therefore rely on so called infoboxes (see Figure 5), table-like templates which are used to describe entities of different types in a standardized manner. We use a manually created mapping between infobox types and our own location types to filter relevant pages (for example, the infobox in Figure 5 is of type `protected area`, which is mapped to the type `POI` in our schema) and add those pages to our database, which provide geographic coordinates.

As a further step, we tried to exploit Wikipedia-internal redirects to obtain alternative names for the extracted locations (for example, when trying to access the Wikipedia article "Alcatraz", one is redirected to "Alcatraz Island"). However, we came to the conclusion, that often very obscure redirects exist, which are not in general language use and therefore degrade the quality of our database. As an alternative for future improvement, we suggest to only extract those alternative names, which are explicitly mentioned and highlighted in an article's introduction.

As we perform no explicit deduplication, we have no exact figures on how many previously unknown locations we actually retrieve through the additional sources to GeoNames, but in our experiments we achieved a noticable recognition improvement. Our

[16] `http://www.hotelsbase.org`
[17] `http://protectedplanet.net`
[18] `http://en.wikipedia.org/wiki/Main_Page`
[19] `http://dbpedia.org/About`

Figure 5: Infobox on the English Wikipedia article "Alcatraz Island" providing geographic coordinates

extraction mechanisms can cope with semantically duplicated location data, so eliminating them is not important for us, but could be achieved using record linkage strategies.

6 Experiments

In this section, we will present our findings during the optimization of both presented approaches. The heuristic approach (see Section 4.2) can be fine-tuned using the presented threshold values. For the machine learning approach (see Section 4.3), we presented a set of features which can be used for the classifier. However, not all of those features might be necessary or useful, therefore we are presenting a backward feature elimination to narrow down our full feature set to a reduced necessary subset. We use QuickDT[20] with its Bagging implementation for the machine learning-based approach.

We use precisio, recall, and F1 measure as harmonic mean in the evaluation. First, we evaluate the TR and classification results following the "MUC" evaluation scheme (two dimensional evaluation with separate scoring for correct type and correct boundaries, giving one point for each, and two points in case both were identified correctly). For the TD, we evaluate the spatial distance between the location given in the dataset and the location given by the extractor and assume a correct disambiguation in case the distance is below 100 km. In the following, we also give precision, recall and F1, denoted as "Geo". We do not use a scoring based on actual distances' values such as RMSE[21], as this would pose a disadvantage to other approaches using different gazetteer data which will be compared in Section 7. The TD is only evaluated for locations of the types `CITY` and `POI`, as the shapes of other locations are generally to broad to perform a point-based matching.

6.1 Parameter Optimization for Heuristic TD

The presented heuristic was developed using the training set of TUD-Loc-2013 (see Section 3). The heuristic is based on seven threshold values which were selected intuitively at first. Consecutively, we will examine the impact on the extraction results when varying each of those threshold values, while keeping the rest of the values to the initial default value as given in Table 3. The analysis is performed on the validation set.

[20] `https://github.com/sanity/quickdt`
[21] Root-mean-square error

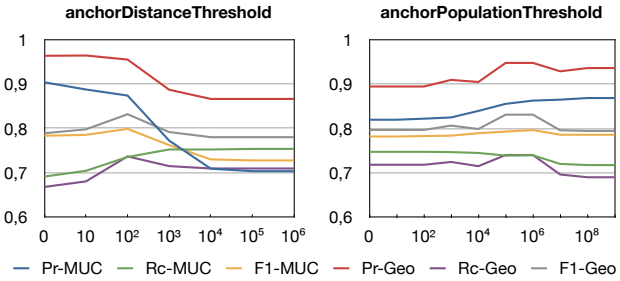


Figure 6: Influence of selected threshold settings on extraction performance of the heuristic approach

The results of our analysis show, that only the variation of *anchorDistanceThreshold* and *anchorPopulationThreshold* have a noticable influence on the extaction results. The value of 2 for the *tokenThresholds* gives best results, values above/below yield a lower F1 measures. A variation of the remaining parameters has no significant influence, which indicates on the other side, that our approach does not overfit to the given data and generalizes well. While we had initially planned to use more sophisticated optimization mechanisms, such as genetic algorithms, for parameter tuning, the results clearly indicate, that this is not necessary.

As Figure 6 shows, the given default values already provide good results in terms of F1 measure, while the *anchorDistanceThreshold* allows for a further adjustment towards more precision or recall. Intuitively, the lower the distance threshold between anchor locations and potential further candidates, the higher the precision. Increasing this threshold allows the recall to rise, however, this results in a comparatively strong decrease of precision.

6.2 Feature Elimination for Machine Learning TD

In Table 2, we describe various features which we use for classification. In total, we extracted 70 features. We used 24 tokens for extracting the binary `containsMarker(M)` feature such as "city", "river", "county", etc. For the features `numLocIn(R)`, `populationIn(R)`, `locSentence(R)`, `uniqueLocIn(R)`, which are to be parametrized with a distance `R`, we used values of 10, 50, 100, and 250 km for each. The feature `distLoc(P)` was extracted for values of 1,000, 10,000, 100,000, and 1,000,000 for `P`. Figure 7 shows the results of the backward feature elimination. The process is as follows: We start with the complete feature set. In each iteration, we remove each of the remaining features once and train the classifier using the training set and test the classifier using the validation set (see Section 3). This means, we ran $n(n+1)/2 = 2485$ train/test cycles. We evaluate the classification results using the F1 measure (note,

Threshold	Value	
unlikelyPopulationThreshold	100,000	
anchorPopulationThreshold	1,000,000	
sameDistanceThreshold	50	km
lowerPopulationThreshold	5,000	
tokenThreshold	2	
lassoDistanceThreshold	100	km
anchorDistanceThreshold	100	km

Table 3: Default threshold values for heuristic TD

that we only evaluated the binary classification performance, and did not employ the MUC or Geo F1 measure). After each iteration, we finally remove the feature, where elimination achieved the best results in F1. This means, that with each step to the right on the x-axis in Figure 7, the mentioned feature was removed in addition to the features on the left.

While the results of the backward feature elimination give no direct evidence of how strong or weak a feature is (for that, chi-squared or information gain tests should be employed), our results show, that we can remove a significant amount of features without harming the classification quality. In contrast—the values indicate, that F1 slightly improves, beginning in the last third of the elimination phase. The oscillation of the results is due to statistical properties and could be eliminated through cross validation, however, we wanted to stick to the predefined training/validation set for better reproducibility.

The results of the feature elimination show, that we can build a robust classifier for TD using a comparatively small feature set. For our following comparison, we rely on a set of the top 15 features, starting on the right of Figure 7. It is interesting to observe, that intuitive indicators, which we already employed in our heuristic, such as `populationNorm` or `nameAmbiguity` are among the leading features. Also, spatial proximity-based features such as `populationIn(R)`, `uniqueLocIn(R)`, `locSentence(R)`, as well as the features `num(c)` and `contains(c)`, based on topologic relations, are among the top 15.

A further question is, how to set the probability threshold, above which candidates are classified as locations. Figure 8 shows the evaluation measures with an increasing probability threshold. Intuitively, a lower threshold classifies more candidates as being locations, resulting in high recall, whereas higher threshold values achieve better precision. The best F1 measure is achieved for a probability threshold of 0.2, whereas higher thresholds lead to decreasing F1 values due to the high loss of recall.

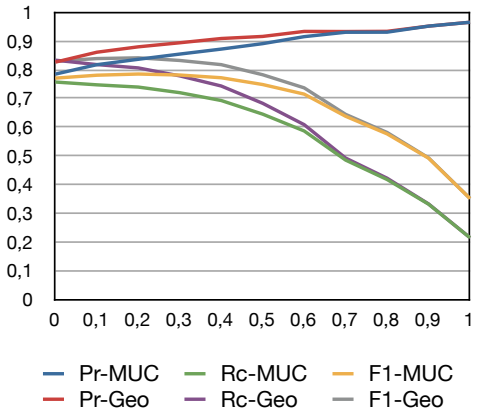


Figure 8: Results of the threshold analysis

7 Comparison

We evaluate our two approaches using TUD-Loc-2013 and the methods already described in Section 6 and compare them to publicly available state of the art approaches for location extraction. In particular, we consider the following Web-based APIs: Yahoo! BOSS GeoServices[22], Unlock Text[23], OpenCalais[24], AlchemyAPI[25], and Extractiv[26]. While all of the mentioned services perform a TR, only Yahoo and Unlock provide a full TD by returning geographical coordinates with each extracted location. OpenCalais, and Extractiv at least retured coordinates for some of the extracted toponyms, while AlchemyAPI does not deliver any coordinates at all.

Naturally, each of the service relies on its own set of location types. We therefore perform a mapping to the location types used in the dataset (see Table 1). The mapping was evaluated and optimized in advance using the training set to ensure a fair comparison.

Unlock is the only service which does not categorize the extracted locations, which is why we exclude it from the TR evaluation.

We compare the results to a baseline TD approach, using a "maximum population" heuristic, which either disambiguates candidates by taking the `CONTINENT` or `COUNTRY` locations, if such exist, or selects the location with the highest population count. The preprocessing and postprocessing phases are identical to the ones described in Section 4.1 and 4.4.

Figure 9 shows the comparison between the baseline, our approaches and state of the art services for the TR task. It is noteworthy, that the baseline already gives comparatively good results and even beats Yahoo in F1, which is due to the strong recall. Alchemy, OpenCalais and Extractiv perform considerably better, but are still outperformed, both by the heuristic and the machine learning TD method. Through machine learning, we can improve F1 to 77,09 % compared to the 76,02 % achieved via the heuristic approach. The runner-up Alchemy achieved 74,12 % F1.

While the previous comparison evaluated the accuracy of the TR, i. e. the correct identification and classification of location entities in text, the TD evaluation investigates, how well the different approaches identify the correct geographical locations (see Figure 10). In contrast to the comparison above, we do not consider Alchemy, OpenCalais and Extractiv here, as they do not provide coordinates for most extractions. On the other hand, we add Unlock to the comparison, which does not categorize extracted locations, but provides coordinates for all of them.

Even more than in the TR comparison, we can see that the baseline already performs considerably well and outperforms Yahoo and Unlock. Yahoo provides a high precision for the TR, but suffers from the comparatively poor results achieved during TD. In contrast

[22] `http://developer.yahoo.com/boss/geo/`
[23] `http://unlock.edina.ac.uk/home/`
[24] `http://www.opencalais.com/documentation/calais-web-service-api/api-metadata/entity-index-and-definitions`
[25] `http://www.alchemyapi.com/api/entity/`
[26] `http://extractiv.com`

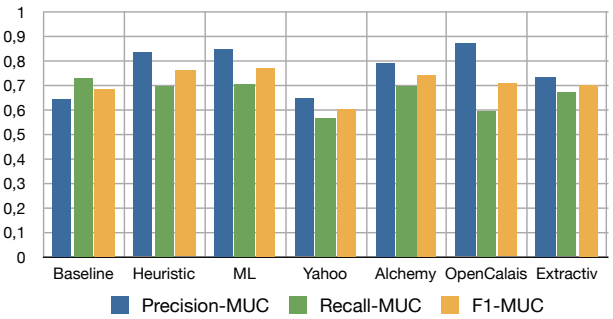


Figure 9: Comparison of TR on TUD-Loc-2013

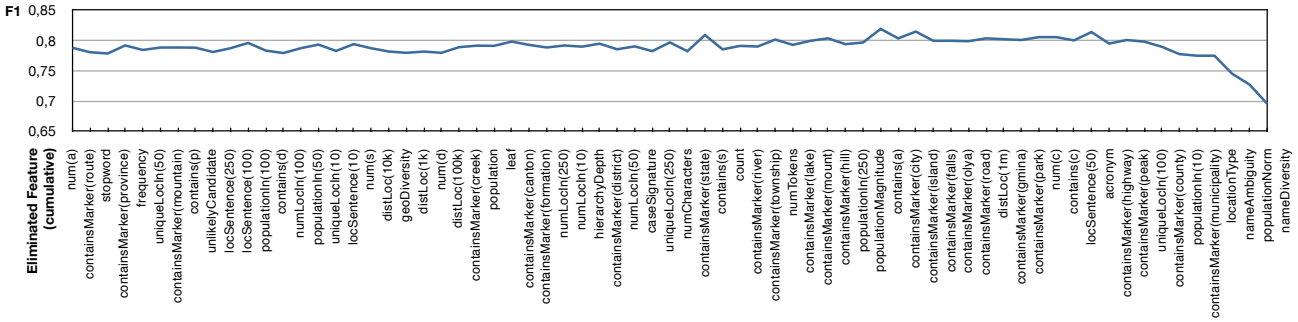


Figure 7: Results of the backward feature elimination

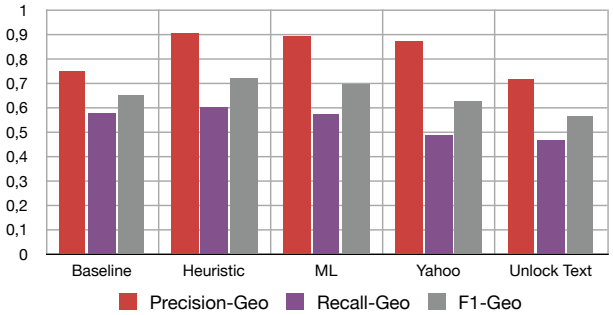


Figure 10: Comparison of TD on TUD-Loc-2013

to the results above, the machine learning-based approach falls short with 69,79 % F1 in comparison to the heuristic, which achieves an F1 value of 72,32 %.

8 Conclusions and Future Work

In this paper we have presented a new dataset for evaluating TR and TD approaches. TUD-Loc-2013 is publicly available for research purposes on the open research platform Areca and therefore facilitates the comparison of future approaches with the results presented here. We have presented two new methods for TR and TR in detail, one relying on a set of heuristics, the other using a classifier trained with machine learning. We have described a comprehensive set of features and seen, that a small subset of those features is sufficient for a well-performing classification-based method.

We have described an aggregated gazetteer database that improves extraction results. Incorporating further sources for obtaining street and ZIP information might further improve those results, but also possibly introduce inaccurate information. An alternative approach might try to exploit map APIs such as Bing or Google when necessary.

On the other hand, the recall achieved during TR can be further improved by extracting more location entities not found in the database. Currently, our approach only allows extraction of address-specific information not in the gazetteer, such as ZIP codes, street names and numbers. We have seen, that currently, the heuristic and machine learning approach deliver a neck-and-neck race, with each one winning either the TR and the TD competition.

In our comparison we have shown, that we can outperform other publicly available Web APIs for extracting location data, using both—heuristic and machine learning—approaches. Thus, we provide a strong foundation for improving current and future applications relying on location-specific data, such as search, geo-targeted advertising, data mining and anlysis, and many more.

Beside the presented dataset TUD-Loc-2013, the methods described within this paper are available as ready-to-use implementations in the Java-based information retrieval toolkit Palladian[27], which is freely available for non-commercial, scientific applications (Urbansky et al. 2012).

References

Amitay, E., Har'El, N., Sivan, R. & Soffer, A. (2004), Web-a-where: Geotagging web content, *in* 'Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval', pp. 273–280.

Andogah, G. (2010), Geographically constrained information retrieval, Dissertation, Rijksuniversiteit Groningen.

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

Buscaldi, D. & Rosso, P. (2008), 'A conceptual density-based approach for the disambiguation of toponyms', *International Journal of Geographical Information Science* **22**(3).

da Graça Martins, B. E. (2008), Geographically aware web text mining, Dissertation, Universidade de Lisboa, Faculdade de Ciências, Departamento de Informática.

Gale, W. A., Church, K. W. & Yarowsky, D. (1992), One sense per discourse, *in* 'Proceedings of the workshop on Speech and Natural Language', HLT '91, pp. 233–237.

Grishman, R. & Sundheim, B. (1996), Message understanding conference – 6: A brief history, *in* 'Proceedings of the 16th International Conference on Computational Linguistics', Vol. 1 of *COLING '96*, pp. 466–471.

Leidner, J. L. (2006), 'An evaluation dataset for the toponym resolution task', *Computers, Environment and Urban Systems* **30**(4), 400–417.

Leidner, J. L. (2007), Toponym resolution in text: Annotation, evaluation and applications of spatial grounding of place names, Dissertation, Institute for Communicating and Collaborative Systems, School of Informatics, University of Edinburgh.

Li, H., Srihari, R. K., Niu, C. & Li, W. (2002), Location normalization for information extraction, *in* 'Proceedings of the 19th international conference on Computational linguistics', Vol. 1 of *COLING '02*, pp. 1–7.

[27] http://palladian.ws

Li, H., Srihari, R. K., Niu, C. & Li, W. (2003), Infoxtract location normalization: a hybrid approach to geographic references in information extraction, *in* 'Proceedings of the HLT-NAACL 2003 Workshop on Analysis of Geographic References', pp. 39–44.

Lieberman, M. D. & Samet, H. (2012), Adaptive context features for toponym resolution in streaming news, *in* 'Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval', SIGIR'12, pp. 731–740.

Lieberman, M. D., Samet, H. & Sankaranarayanan, J. (2010), Geotagging with local lexicons to build indexes for textually-specified spatial data, *in* 'Proceedings of the 26th International Conference on Data Engineering', ICDE 2010, pp. 201–212.

Millan, M., Sánchez, D. & Moreno, A. (2008), 'Unsupervised web-based automatic annotation', *Proceedings of the 4th STAIRS Conference: Starting AI Researchers' Symposium* .

Parsons, E. (2012), 'Ed Parsons at Google PinPoint London 2012', `https://plus.google.com/110553637244873297610/posts/8SYwR5Ze6nB`.

Rauch, E., Bukatin, M. & Baker, K. (2003), A confidence-based framework for disambiguating geographic terms, *in* 'Proceedings of the HLT-NAACL 2003 workshop on Analysis of geographic references', Vol. 1, pp. 50–54.

Sinnott, R. W. (1984), 'Virtues of the haversine', *Sky and Telescope* **68**(2), 159.

Smith, D. A. & Crane, G. (2001), Disambiguating geographic names in a historical digital library, *in* 'Proceedings of the 5th European Conference on Research and Advanced Technology for Digital Libraries', ECDL '01, pp. 127–136.

Smith, D. A. & Mann, G. S. (2003), Bootstrapping toponym classifiers, *in* 'Proceedings of the HLT-NAACL 2003 workshop on Analysis of geographic references', Vol. 1 of *HLT-NAACL-GEOREF '03*, pp. 45–49.

Urbansky, D. (2012), Automatic extraction and assessment of entities from the web, Dissertation, Technische Universität Dresden, Faculty of Computer Science.

Urbansky, D., Muthmann, K., Katz, P. & Reichert, S. (2012), 'TUD Palladian Overview'.

Wing, B. P. & Baldridge, J. (2011), Simple supervised document geolocation with geodesic grids, *in* 'Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies', Vol. 1 of *HLT '11*, pp. 955–964.

Searching Frequent Pattern and Prefix Trees for Higher Order Rules

Ping Liang, John F. Roddick and Denise de Vries

School of Computer Science, Engineering and Mathematics
Flinders University,
PO Box 2100, Adelaide, South Australia 5001
{ping.liang, john.roddick, denise.devries}@flinders.edu.au

Abstract

Since the search for rules that can inform business decision making is the ultimate goal of data mining technology, problems such as the interpretation of interestingness for discovered rules is an important issue. However, the search for rules that adhere to a user's definition of interesting remains somewhat elusive, in part because rules are commonly supplied in a low, instance-level format.

In this paper we argue that rules with more useable semantics can be obtained by searching for patterns in the intermediate data structures such as frequent pattern or prefix trees. This paper discusses this approach and present a proof-of-concept system, Horace, that shows that the approach is both useable and efficient.

1 Introduction

Since the early work of Agrawal, Srikant and others (Agrawal et al. 1993, Agrawal & Srikant 1994, Srikant & Agrawal 1995) association mining research has become a mature field (Ceglar & Roddick 2006) and has been applied in a variety of industry sectors including commerce, defence, health, manufacturing, exploration and engineering. One form of data mining algorithm, association mining algorithms, have the capacity to rapidly discover sets of co-occurring items or events in very large databases and the time complexity of most algorithms is generally close to linear in the size of the dataset (Zaki & Ogihara 1998). A variety of extensions have been proposed that enable, for example,

- temporal (Ale & Rossi 2000, Li et al. 2003, Rainsford & Roddick 1999) and spatial (Han et al. 1997, Koperski & Han 1995) semantics to be accommodated,
- closed sets to be identified (Pasquier et al. 1999, Zaki 2000),
- fuzzy and incomplete data to be handled (Chan & Au 1997, Kuok et al. 1998),
- the accommodation of domain-specific concept hierarchies (Cheung, Ng & Tam 1996, Fortin & Liu 1996, Han & Fu 1995, Shen & Shen 1998), and
- the application of visualisation techniques (Ong et al. 2002).

Clearly, in order to create useable systems, problems such as the interpretation of interestingness for discovered rules are an important issue and need to be resolved. Unfortunately, the search for rules that adhere to a user's definition of interesting (and indeed, even the user's definition of interesting) remains some-what elusive (Geng & Hamilton 2006), in part because rules are generally supplied in an instance-level format, such as

$$DigitalTV \wedge DVDPlayer \rightarrow Cables \quad \sigma(20\%)\gamma(65\%) \tag{1}$$

where the σ (support) and γ (confidence) values are examples of some quality metric for the rule.

Such low-level rules, while useful, provide knowledge only about the coincidence of elementary values and can be termed zero-order rules. Higher order semantics can be derived when sets of rules are inspected to determine patterns of interest between rules. For example, two competitor items a and b may be discovered by observing a set of rules such that:

$$\{a\} \rightarrow \{c\} \quad \sigma(x) \tag{2}$$
$$\{b\} \rightarrow \{c\} \quad \sigma(y) \tag{3}$$
$$\{a, b\} \rightarrow \{c\} \quad \sigma(z) \tag{4}$$
$$\sigma(z) < \sigma(x) \times \sigma(y) \tag{5}$$

That is, the observed value for Eq.(5) is considerably lower than one would have expected with independent items. Other patterns of rules can also be detected in this way including catalysts, and others.

In the past specific algorithms have been developed to search for each case. For example, Teng (2002) outlines a mechanism for learning dissociations (aka competitors) from source data.

Since frequent pattern and prefix trees are (generally speaking) isomorphic with the resulting ruleset, our approach here is to search such data structures directly for patterns. The (higher order) semantics are expressed as an FP-tree pattern and our algorithm is thus able to find a variety of higher order rules in one pass of the FP-tree or prefix tree.

The rest of the paper is as follows. Section 2 discusses other work in higher order mining and section 3 provides a description of some basic notation and concepts used in this paper. Section 4 defines patterns in ruleset while Section 5 outlines our approach in broad detail. Section 6 discusses a particular variant in which FP-trees are searched. Section 7 discusses Horace, our proof-of-concept system and Section 8 provides a discussion of future work.

2 Related Work

Direct current work in the area is relatively limited. With a few notable exceptions, data mining research has largely focused on the extraction of knowledge directly from the source data. However, in many cases such mining routines are beginning to encounter problems as the volume of data requiring analysis grows disproportionately with the comparatively slower improvements in I/O channel speeds. That is, many data mining routines are becoming heavily I/O bound and this is limiting many of the benefits of the technology. Methods of reducing the amount of data have thus been discussed in the literature including statistical methods such as sampling or stratification, reducing the dimensionality of the data by, for instance, ignoring selected attributes, or by developing incremental maintenance methods by analysing the changes to data only (Cheung, Han, Ng & Wong 1996).

Moreover, while unconstrained exploratory data mining is useful, the diversity of datasets and dimensions used in data mining means that some form of guidance is often useful. Most approaches cater for this by allowing users to either specify filters or by restricting the input data. Our approach is to allow users to specify a high-level (i.e. user-oriented) descriptions of the knowledge required. For example, analysts looking to reduce hospital costs may look for situations where potential alternatives exist - i.e. pairs of items which rarely occur together but almost always occur with the same other items.

One approach is to plug the output of one mining algorithm into another. For example, Lent et al. (1997) show how association rules may be clustered. Gupta et al. (1999) extends this work by looking at distance based clustering of association rules and Perrizo & Denton (2003) outline a framework based on partitions to unify various forms of data mining algorithm. Higher order mining more generally was discussed by Roddick et al. (2008).

3 Preliminaries

Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of items, and D be a dataset containing a set of transactions, where each transaction t_i is a set of items such that $t_i \subseteq I$. Each transaction may have subsets which are called itemsets. An association rule is an implication expression of the form

$$X \to Y \quad \sigma, \gamma \tag{6}$$

where $X, Y \subset I$ and $X \cap Y = \Phi$. We call X the antecedent and Y the consequent of the rule. An association rule commonly has two measurements: support (often denoted by σ) and confidence (often denoted by γ). Support (σ) is the ratio of the number of transactions containing both the antecedent and consequent to the number of transactions in D, defined as $\frac{P(X \cup Y)}{|D|}$, where $P(X \cup Y)$ denotes the number of transactions containing X and Y. Confidence (γ) is defined as the ratio $\frac{P(X \cup Y)}{P(X)}$.

For brevity, we denote a rule r_i with antecedent X, consequent Y and support σ as:

$$r_i : X \to Y(\sigma) \tag{7}$$

We call $X \cup Y$ the itemset of rule r_i and $r_i.ac$, $r_i.cs$ be its antecedent and consequent, thus in this case, $r_i.ac = X$, $r_i.cs = Y$. The support of r_i is denoted as $\sigma(r_i)$.

Given two rules r_i and r_j, if $r_i.cs = r_j.cs \wedge r_i.ac \cap r_j.ac = \Phi$, i.e., they have the same consequent but disjoint antecedent, we call r_i a sibling of r_j and vice versa. We denote them as $Sib(r_i, r_j)$. Given multiple sibling rules we denote them as $Sib(r_1, r_2, \ldots, r_n)$.

If $r_i.cs = r_j.cs \wedge r_i.ac \subset r_j.ac$, i.e., they have the same consequent but the antecedent of r_j contains the antecedent of r_i, we call r_j the parent of r_i and r_i a child of r_j, which are denoted as $Par(r_j, r_i)$. Given a parent rule r_p with a set of sibling rules as its children, $Sib(r_1, r_2, \ldots, r_n)$, we denote them as $Par(r_p, Sib(r_1, r_2, \ldots, r_n))$.

To illustrate, consider the ruleset below:

$$r_1 : \{a\} \to \{c\} \quad (60\%) \tag{8}$$
$$r_2 : \{b\} \to \{c\} \quad (70\%) \tag{9}$$
$$r_3 : \{a, b\} \to \{c\} \quad (10\%) \tag{10}$$
$$|D| = 1000$$

Since $r_1.cs = r_2.cs = \{c\} \wedge r_1.ac \cap r_2.ac = \{a\} \cap \{b\} = \Phi$ then r_1 and r_2 are siblings. Also, since $r_3.cs = r_1.cs = \{c\}, r_1.ac = \{a\} \subset r_3.ac = \{a, b\}$, r_3 is the parent of r_1. Similarly we find that r_3 is a parent of r_2. The three rules can thus be denoted as $Par(r_3, Sib(r_1, r_2))$.

Given a set of sibling rules $R = Sib(r_1, r_2, \ldots, r_n)$ we can define a relative support ρ of rule $r_i \in R$ as follows:

$$\rho(r_i) = \frac{P(r_i.ac \cup r_i.cs) - Q(r_i.ac \cup r_i.cs)}{|D|} \tag{11}$$

where $Q(r_i.ac \cup r_i.cs)$ denotes the number of transactions containing the antecedent and consequent of other rules in R in all transactions containing the antecedent and consequent of r_i.

Relative support represents the occurrence of the antecedent of a sibling rule without the existence of other sibling rules' antecedent, when occurring together with their consequent.

To illustrate, consider the relative support of the two sibling rules r_1 and r_2 in the example above. According to the definition of support, we have:

$$\sigma(r_1) = \frac{P(r_1.ac \cup r_1.cs)}{|D|} \tag{12}$$
$$\sigma(r_2) = \frac{P(r_2.ac \cup r_2.cs)}{|D|} \tag{13}$$
$$\sigma(r_3) = \frac{P(r_3.ac \cup r_3.cs)}{|D|} \tag{14}$$

Thus, we have

$$\begin{aligned} P(r_1.ac \cup r_1.cs) &= P(\{a, c\}) \\ &= \sigma(r_1) \times |D| \\ &= 60\% \times 1000 = 600 \end{aligned} \tag{15}$$

Similarly

$$P(r_2.ac \cup r_2.cs) = 700 \tag{16}$$
$$P(r_3.ac \cup r_3.cs) = 100 \tag{17}$$

Result (15) shows there are 600 transactions containing items a and b, which is the antecedent and consequent of r_1 respectively. In order to calculate $\rho(r_1)$, we need to calculate $Q(r_1.ac \cup r_1.cs)$ which is the number of transactions containing the antecedent and consequent of its sibling, r_2, from result (15).

Since $r_2.ac = \{b\}, r_2.cs = \{c\}$ and result (15) contains all transactions with $\{a, c\}$, it is clear that

$$Q(r_1.ac \cup r_1.cs) = P(\{a, b, c\}) = 100 \quad (18)$$

and therefore,

$$\rho(r1) = 0.5 \quad (19)$$
$$\rho(r2) = 0.6 \quad (20)$$

That is, item a occurs in 50% transactions together with item c without the existence of item b and that b occurs in 60% transactions together with item c without the existence of item a.

4 Defining Patterns in Rulesets

Let $R = \{r_1, r_2, \ldots, r_n\}, n > 2$, be a set of rules. A pattern in R is denoted as

$$RP = \{Rset|P\} \quad (21)$$

where $Rset = \{r|r \in R\}$ and P is the condition(s) that $Rset$ holds.

Definition 1 (Competitor Pattern):
Let $R = \{r_1, r_2, \ldots, r_n\}, n > 2$, be a set of rules. Given a user specified threshold $minH$ and $maxL$, where $minH \geq maxL > 0$, a competitor pattern is denoted as

$$\begin{aligned} CoPatt &= \{Par(r_p, Sib(r_i, r_j))| \\ &\quad \rho(r_i) \geq minH, \\ &\quad \rho(r_j) \geq minH, \\ &\quad \sigma(r_p) \leq maxL \\ &\quad \sigma(r_p) < \sigma(r_i) \times \sigma(r_j) \\ &\quad r_i, r_j, r_p \in R\} \end{aligned} \quad (22)$$

As shown in the above definition, a competitor pattern contains a parent rule r_p with two child rules r_i and r_j. The Pattern requires that the relative support of r_i and r_j is higher than user specified threshold $minH$ and the support of the parent rule is lower than threshold $maxL$. It also requires that the itemsets of rules r_i and r_j are statistically negatively correlated.

Competitor patterns illustrate the relationship between the antecedent of rules r_i and r_j, where one suppresses the other when occurring together with their consequent resulting in unexpected low support. To illustrate, given $minH = 0.4$, $maxL = 0.2$, and take our running example. From the results above, we have $\rho(r_1) = 0.5 > minH, \rho(r_2) = 0.6 > minH$. Since $\sigma(r_3) = 0.1 < maxL$, so the first three conditions are satisfied. Furthermore, we have $\frac{\sigma(r_3)}{\sigma(r_1) \times \sigma(r_2)} = \frac{0.1}{0.6 \times 0.7} = 0.238 < 1$, thus $\sigma(r_3) < \sigma(r_1) \times \sigma(r_2)$. So the fourth condition is also met and we have found a matched instance which reveals that items a and b suppress each other when occurring together with c.

Definition 2 (Twoway-Catalyst Pattern):
Let $R = \{r_1, r_2, \ldots, r_n\}, n > 2$, be a set of rules. Given a user specified threshold $minH$ and $maxL$, where $minH \geq maxL > 0$, a Twoway-Catalyst pattern is denoted as

$$\begin{aligned} CaPatt &= \{Par(r_p, Sib(r_i, r_j))| \\ &\quad \rho(r_i) \leq maxL, \\ &\quad \rho(r_j) \leq maxL, \\ &\quad \sigma(r_p) \geq minH, \\ &\quad \sigma(r_p) > \sigma(r_i) \times \sigma(r_j) \\ &\quad r_i, r_j, r_p \in R\} \end{aligned} \quad (23)$$

As shown in the above definition, the first three conditions require that $\rho(r_i)$ and $\rho(r_j)$ are lower than threshold $maxL$ and the support of the parent rule r_p is higher than threshold $minH$. The fourth condition requires that the itemsets of the two sibling rules r_i and r_j should be statistically positively correlated.

Twoway-Catalyst patterns are similar to competitor patterns except that it illustrates a positive relationship between the antecedent of rule r_i and r_j, where one facilitates the other when occurring together with their common consequent.

Definition 3 (Threeway-Catalyst Pattern):
Let $R = \{r_1, r_2, \ldots, r_n\}, n > 3$, be a set of rules. Given a user specified threshold $minH$ and $maxL$, where $minH \geq maxL > 0$, a Threeway-Catalyst pattern is denoted as

$$\begin{aligned} CaPatt &= \{Par(r_p, Sib(r_i, r_j, r_k))| \\ &\quad \rho(r_i) \leq maxL, \\ &\quad \rho(r_j) \leq maxL, \\ &\quad \rho(r_k) \leq maxL, \\ &\quad \sigma(r_p) \geq minH, \\ &\quad \sigma(r_p) > \sigma(r_i) \times \sigma(r_j) \times \sigma(r_k) \\ &\quad r_i, r_j, r_k, r_p \in R\} \end{aligned} \quad (24)$$

Threeway-Catalyst patterns illustrate the relationship among the antecedent of three sibling rules, which seldom occur individually, but more commonly occur together with their consequent.

These three types of patterns in rulesets widely exist in real world. Table 1 represents a descriptive list of the patterns in market basket data. However, such pattern may be exhibited within many domains. For example, in medicine, the exposure to different conditions may result in increased possibility of illness. In addition, patterns in rulesets may combine to form more complex patterns. The conjunction of ruleset patterns is beyond the scope of this paper.

5 The Horace approach

Horace adopts a tree-based approach which extends the concepts discussed in the presentation of FP-trees (Han & Pei 2000, Han et al. 2000). Since the frequent pattern trees contain all of the information represented by the (larger) rulesets generated from them, it is firstly possible and secondly more efficient to search for patterns in the trees than it is to find them from the rulesets. The overall context of Horace is shown in Figure 1.

There are three key parts to the Horace framework: the FP-tree, an RP library with an associated pattern language and a pattern search algorithm. The FP-tree is generated to concisely represent the pertinent dataset information. The Ruleset Pattern library contains a set of RP-trees (as discussed in Section 5.2), each of which represents an RP. The integration of a pattern library and its associated pattern language allows users to retrieve, modify and create new patterns in Horace based on their requirements and their

Table 1: Sample Ruleset Patterns

Pattern	Description	Example
Competitor	An itemset competes with another itemset	Ruleset : $\{cola\} \rightarrow \{chips\}$, $\{lemonade\} \rightarrow \{chips\}$, $\{cola, lemonade\} \rightarrow \{chips\}$ Description: Customers tend to buy chips and cola or chips and lemonade individually, but they seldom buy chips, cola and lemonade together.
Twoway-Catalyst	An itemset facilitates another itemset	Ruleset : $\{milk\} \rightarrow \{bread\}$, $\{butter\} \rightarrow \{bread\}$, $\{milk, butter\} \rightarrow \{bread\}$ Description: When customers buy bread, they tend to buy milk and butter together but not individually.
Threeway-Catalyst	Three or more itemsets frequently occur together but more rarely occur individually	Ruleset: $\{turkey\} \rightarrow \{ChristmasCards\}$ $\{crackers\} \rightarrow \{ChristmasCards\}$ $\{ham\} \rightarrow \{ChristmasCards\}$ $\{turkey, crackers, ham\} \rightarrow \{ChristmasCards\}$ Description: Turkey, Crackers and Ham are frequently bought together with Christmas Cards.

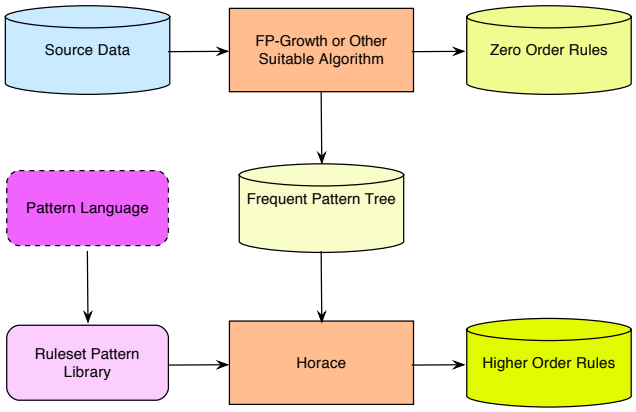


Figure 1: **Overall Context to Horace**

Algorithm 5.1 RP-tree Construction

```
1: Input: A ruleset pattern RP
2: Output: An RP-tree
3: Create root T
4: sort items in the parent rule itemset (RP.parent) in descending
   order of support
5: insert-tree(RP.parent, T, 0)
6: for each sibling rule r in RP do
7:     sort items in the itemset e of r according to their order in
       RP.parent
8:     insert-tree(e, T, 0)
9: end for
10: insert-tree(itemset e, treeNode node, int index)
11: if index < e.length then
12:     if node.hasChild() then
13:         if node.child.item − name ≠ e[index].item − name
            then
14:             node.addChild(e[index]),
15:             linkset.add(e[index])
16:         end if
17:     else
18:         node.addChild(e[index]),
19:         linkset.add(e[index])
20:     end if
21:     insert-tree(e,node.child,index++)
22: end if
```

datasets. At the core of Horace, there is a search algorithm to find all matches of a given RP held in an FP-tree.

5.1 FP-trees

FP-trees, proposed by Han et al. (2000), are a compact data structure that contains the complete set of information held in a database relevant to frequent pattern mining. The FP-tree stores a single item at each node, and includes a support count and additional links to facilitate processing. These links start from a header table and link together all nodes in the FP-tree which store the same item.

5.2 RP-trees

Horace represents patterns in rulesets using a novel RP-tree (Ruleset Pattern tree). An RP-tree has a set of prefix subtrees as the children of the root (denoted as T). Each node consists of three fields: the item, a node link and a count, where the node link points to the next node containing the same item in the RP-tree. The algorithm to construct an RP-tree is outlined below.

As indicated in the work of Han et al. (2000), items stored in a FP-tree are on their support descending order. Thus, the RP-tree construction process firstly sorts items in the parent rule itemset in support descending order to facilitate the searching process in the FP-tree. The parent rule itemset is then inserted into the tree followed by each child rule itemset. During the insertion process, when a new node is inserted, a node link is created pointing to the next node containing the same item in the RP-tree.

After construction, a language is required to describe the RP-tree. An RP-tree is defined as a collection of tuples $< node, parent, child, weight >$. If a branch represents the itemset of a parent rule or sibling rule, weight represents the support or relative support of the rule respectively. In addition, a label (denoted as $Desc$), which is an instantiated description of the pattern, can be imposed over the definition.

Given a database D and a user specified threshold $minH$ and $maxL$, Figure 2 shows the construction process of a competitor pattern containing three rules as follows:

$$\begin{aligned} r_i &: \{b\} \rightarrow \{a\} \\ r_j &: \{c\} \rightarrow \{a\} \\ r_p &: \{b, c\} \rightarrow \{a\} \end{aligned}$$

then the pattern RP might be:

$$\begin{aligned} RP \quad = \quad & \{Par(r_p, Sib(r_i, r_j))| \\ & \rho(r_i) \geq minH, \\ & \rho(r_j) \geq minH, \\ & \sigma(r_p) \leq maxL, \\ & \sigma(r_p) < \sigma(r_i) \times \sigma(r_j)\} \end{aligned} \tag{25}$$

Figure 2(a) creates the root T and inserts items in the parent rule (i.e.a, b,and c) based on the descending order of support. Weight, z, which is the support of the rule is noted on the arc between node b and c. Figure 2(b) shows the effect of inserting items of r_i (i.e., a and b) and its weight x. Figure 2(c) illustrates the effect of inserting items of r_j (i.e., a and c) and its weight y. A link is created between the two 'c' nodes.

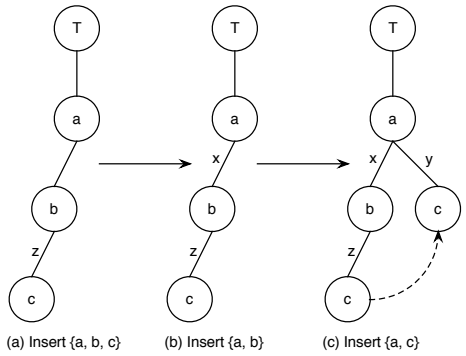


Figure 2: **Sample RP-tree and Description**

Thus the tuples which are constructed for all non-leaf RP-tree nodes.

$$\begin{aligned} & < T, , a, > \\ & < a, T, b, x > \\ & < a, T, c, y > \\ & < b, a, c, z > \\ & x \geq minH, y \geq minH, z \leq maxL \\ & \sigma(\{a, b, c\} < \sigma(\{a, b\}) \times \sigma(\{a, c\}) \\ & Desc : \text{“}CompetitorPattern\text{”} + b + \text{“,”} + c \end{aligned}$$

6 Searching the FP-tree

Given the structure of the two trees, RP-tree and FP-tree, the Horace algorithm searches the FP-tree for all matches of the RP-tree (shown in the algorithm below). Horace provides a mechanism for tree searching, which substitutes RP-tree nodes with the items from the FP-tree header table. The complexity of the process is $O\left(\frac{n!}{(n-k)!k!}\right)$, where n is the number of frequent items in an FP-tree and k is the number of distinct items in an RP-tree.

As the RP-tree's leftmost branch contains all items in the parent rule, Horace requires a single pass over this branch for search purposes. For each node in this parent rule branch, all other nodes representing the same item are accessible through the node links. The collection of itemsets that terminate with a specific item is efficiently obtained through this structure. During the process to calculate support, Horace visits the relevant FP-tree branches by following the link of the specific item in the header table.

Algorithm 6.1 Horace Algorithm

```
1: Input: FP-tree fp, RP-tree rp
2: Output: Matched instances of rp
3: mineHorace(rp.root.next)
4: mineHorace(RPtreeNode node)
5: if node != null then
6:     P = arrayOfPath(node)
7:     for each item in fp.header do
8:         C = getCountList(item,P)
9:         if node.prefix.length == rp.length then
10:            if !isValid(rp.condition, C) then
11:                return
12:            end if
13:        end if
14:        mineHorace (node.child)
15:    end for
16: end if
17: getCountList(FPtreeNode node, Array P)
18: countList = new int[P.size]
19: while node != null do
20:    if node.prefix.hasPath(P[i]) then
21:        countList[i] = countList[i] + node.count
22:    end if
23:    node = node:link
24: end while
25: return countList
26: arrayOfPath(RPtreeNode node)
27: path = new Array()
28: while node != null do
29:    path.add(node.prefix)
30:    node = node.link
31: end while
32: return path
```

If a visited branch contains the parent rule's itemset, the support of the parent rule increases by adding the support count of the visited node in the branch. If there is no parent rule's itemset in the branch and it contains a child rule's itemset, the relative support of the itemset increases by adding the support count of the visited node too. If there is no parent rule's itemset in the branch and the branch contains more than one child rule's itemsets, those itemsets are not independent of each other in that branch and the support count of the visited node will not be counted according to the definition of relative support.

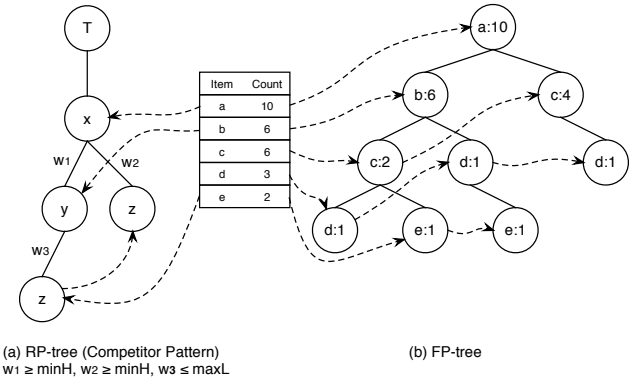


Figure 3: **Illustration of Horace data structures**

To illustrate, consider the example shown in Figure 3, which contains a competitor pattern tree (RP-tree) and an FP-tree. Given $minH = 0.4$, $maxL = 0.2$ and $|D| = 100$, the algorithm starts with node 'x' in the RP-tree. Each node in the RP-tree will be replaced by its corresponding value from the FP-tree header table.

Starting with item 'a', only one branch ends with 'a', and a support count of 10 is obtained from the header table. The process continues, moving to the next node in the traversal, 'y'. The node 'y' is replaced by all items in the header table, except 'a'.

Starting with item 'b', a branch is generated from the RP-tree that ends with 'b' and also contains 'a', which is {a, b}. Following the FP-tree link from 'b', all branches that contain {a, b} are found and the support is calculated, $\sigma(\{a,b\}) = 0.6$, which is checked against the threshold $minH$. Since $\sigma(a,b) = 0.6 > minH$, it is valid. The process advances to 'z', which is replaced with each item in the FP-tree header table except 'a' and 'b'.

Starting with 'c', generate a list of branches in the RP-tree that contains 'c' and have 'a' or 'b' in their parental path, specifically {a, b, c} and {a, c}. The support of {a, b, c} and the support of {a, c} are obtained, $\sigma((\{a,b,c\}) = 0.2$, $\sigma(\{a,c\}) = 0.6$. We can also calculate the relative support for {a, b} (0.4) and {a, c} (0.4). A condition check will be performed for the related three itemsets, {a, b, c}, {a, b} and {a, c}. Since $\sigma(\{a,b,c\}) = 0.2 < maxL$, $\rho(\{a,c\}) = 0.4 = minH$, $\rho(\{a,b\}) = 0.4 = minH$, the first three of the conditions for the *competitor* pattern are met. Furthermore, since $\frac{\sigma\{\{a,b,c\})}{\sigma(\{a,b\})\times\sigma(\{a,c\})} = \frac{0.2}{0.6\times0.6} = 0.56 < 1$ and $\sigma(\{a,b,c\}) < \sigma(\{a,b\}) \times \sigma(\{a,c\})$, all conditions of competitor pattern have been met. Therefore, a matched instance of the RP-tree has been found.

The algorithm then progresses to 'd'. Branches containing 'd' that have 'a' or 'b' in their parental path are {a, b, d} and {a, d}. The support of {a, b, d} and relative support of {a, d} are obtained, i.e., $\sigma(\{a,b,d\}) = 0.2, \rho(\{a,d\}) = 0.1$. Since $\rho\{(a,d)\} = 0.1 < minH$, it is invalid and thus is pruned. The process continues until all nodes in the leftmost branch of the pattern tree have been substituted.

7 Implementation of Horace Proof-of-Concept System

To demonstrate the concept, a prototype of Horace was implemented in Java and several experiments were conducted on both synthetic and real datasets. All tests are done on a 2.6 GHz PC with 1GB of main memory running Windows 7. This implementation is shown to be tractable and able to reveal patterns in ruleset of potential interest that would otherwise not be reported.

7.1 Synthetic Data

A synthetic data generator was built based on the work reported by Agrawal & Srikant (1994) to produce large quantities of transactional data. Table 2 presents the details of the generated synthetic data. Table 3 shows the parameters for data generation, along with their default values and the range of values on which experiments were conducted.

Table 2: Synthetic data

Data	$\|I\|$	$\|T\|$	$\|P\|$	TS	PS
Syn1	100	10,000	20	5	5
Syn2	200	50,000	150	10	10
Syn3	300	100,000	250	10	15

7.2 Real Data

Three datasets were used to test Horace. Their details are shown in Table 4. The Retail Data was supplied by an anonymous Belgian retail supermarket store (Brijs et al. 1999). The data were collected over three non-consecutive periods between 1999 and 2000. The two datasets BMS-WebView-1 and BMS-WebView-2 are taken from KDDCUP 2000 (Kohavi et al. 2000). They contain several months' worth of click stream data from two e-commerce web sites.

7.3 Tested Pattern

Figure 4 shows the patterns tested. Figure 4(a) and Figure 4(d) represent two Competitor patterns, with Figure 4(a) involving three rules: $\{b\} \rightarrow \{a\}$, $\{c\} \rightarrow \{a\}$, $\{b,c\} \rightarrow \{a\}$ and Figure 4(d) involves three rules: $\{m,n\} \rightarrow \{a\}$, $\{p,q\} \rightarrow \{a\}$, $\{m,n,p,q\} \rightarrow \{a\}$. Figure 4(b) represents a Twoway-Catalyst pattern, where items 'b' and 'c' facilitates each other resulting in a higher than one would expect support when they occur together with item 'a'. Figure 4(c) represents a Threeway-Catalyst pattern showing three items 'b', 'c' and 'd' which seldom occur individually with item 'a' but always occur together with item 'a'.

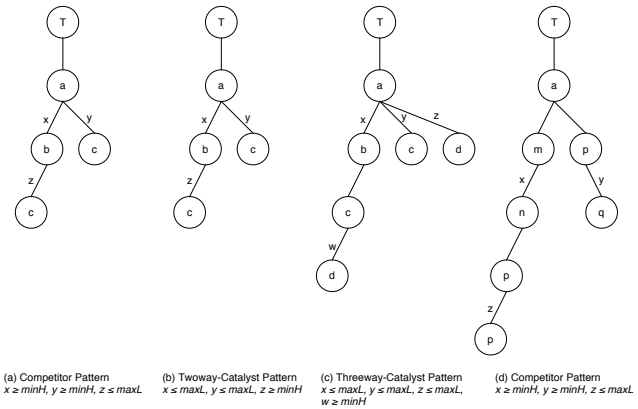


Figure 4: **Test Patterns**

7.4 Results and Evaluation

The experimental results demonstrate that Horace provides a sound and useful means of finding complex and random RP-tree patterns within an FP-tree. Test results, as shown in Table 5 , demonstrate that Horace is able to reveal patterns of potential interest. Patterns (a), (b) and (d) exist in both the real and synthetic datasets, while pattern (c) exists in two synthetic datasets.

Presented below are some examples discovered from Retail-Data (each item is denoted as a character c plus a number):

$$\begin{aligned} \{c39, c2925\} \quad & (\rho = 1.1\%), \\ \{c39, c1146\} \quad & (\rho = 1.1\%), \\ \{c39, c2925, c1146\} \quad & (\sigma = 0.009\%) \end{aligned} \tag{26}$$

Description: item c2925 competes with c1146 when they occur together with item c39.

$$\begin{aligned} \{c14945, c101, c236\} \quad & (\rho = 0.5\%), \\ \{c271, c270, c236\} \quad & (\rho = 0.8\%), \\ \{c14945, c101, c271, c270, c236\} \quad & (\sigma = 0.005\%) \end{aligned} \tag{27}$$

Description: itemset {c14945, c101} competes with itemset {c271,c270} when they occur together with item c236.

Table 3: Synthetic data parameters

Name	Description	Default Value	Range of Values
$\|I\|$	Number of Items	10	10-100
$\|T\|$	Number of Transactions	5,000	5,000-200,000
$\|P\|$	Number of Patterns	50	50-200
TS	Average Size of Transaction	5	5-10
PS	Average Size of Pattern	5	5-10

Table 4: Real datasets

Data	Retail-Data	BMS-WebView-1	BMS-WebView-2
NumberOfTrans	88,163	59,602	77,512
Distinct Items	16,470	497	3,340
Max TransSize	67	267	161
Average TransSize	15	2.5	5.0

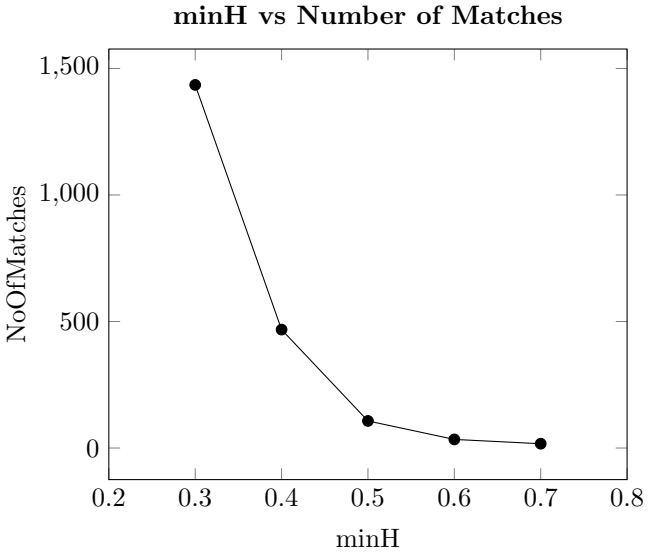


Figure 5: **Performance of Horace**
(minSup = 0.1, maxL =0.3, Data:Syn2)

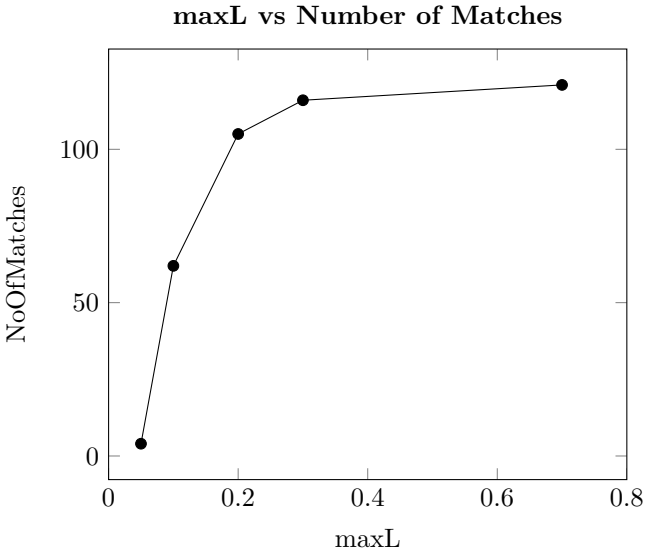


Figure 6: **Performance of Horace**
(minSup = 0.1, minH=0.7, Data:Syn2)

$$\{c39, c682\}(\rho = 24.4\%),$$
$$\{c48, c682\}(\rho = 14.7\%),$$
$$\{c39, c48, c682\}(\sigma = 33.1\%) \tag{28}$$

Description: item c39 facilitates item c48 when they occur together with item c682.

As shown in Figures 5 and 6 the number of matched instances is affected by the setting of $minH$ and $maxL$. The higher $minH$ or the lower $maxL$, the more itemsets with lower support are pruned out, and therefore, the fewer matched instances found. Similarly, the lower $minH$ or the higher $maxL$, the more itemsets with lower support are included, and therefore, more matches can be identified.

8 Discussion of Future Work

This paper outlines an approach to finding patterns in rulesets. It represents a ruleset pattern as a tree structure and searches an FP-tree for its matched instances. The experimental results demonstrate the capacity of this approach to find patterns of potential interest that cannot be identified by other data mining techniques.

The focus of the proof-of-concept implementation was not on performance but on proving that the design decisions were sound. A more efficient algorithm is planned to cope with more complex ruleset patterns and a full pattern language is being developed. The evaluation of the interestingness of a matched ruleset pattern instance is currently based on statistical dependency, which also warrants investigation.

References

Agrawal, R., Imielinski, T. & Swami, A. (1993), Mining association rules between sets of items in large databases, *in* P. Buneman & S. Jajodia, eds, 'ACM SIGMOD International Conference on the Management of Data', ACM Press, Washington DC, USA, pp. 207–216.

Agrawal, R. & Srikant, R. (1994), Fast algorithms for mining association rules, *in* J. Bocca, M. Jarke & C. Zaniolo, eds, '20th International Conference on Very Large Data Bases, VLDB'94', Morgan Kaufmann, Santiago, Chile, pp. 487–499.

Table 5: Test Results

Dataset	Minsup	FP-tree Info			Pattern (a)			Pattern (b)			Pattern (c)			Pattern (d)		
		Depth	Branches	Nodes	minH	maxL	Cnt	minH	maxL	Cnt	minH	maxL	Cnt	minH	maxL	Cnt
RetailData	0.01	12	12,142	31,037	1.0	0.01	6	1.0	1.0	1	1.0	0.5	0	0.5	0.5	48
BMS-WebView1	0.01	31	5,584	16,909	0.5	0.5	11	0.8	0.1	52	0.5	0.5	0	0.8	0.2	10
BMS-WebView2	0.005	28	14,044	48,571	0.5	0.5	5	0.5	0.5	0	0.5	0.5	0	0.8	0.2	8
Syn1	0.2	46	5,842	96,159	1.0	0.5	54	1.0	0.5	32	1.0	0.5	44	1.0	0.5	10
Syn2	0.05	23	17,426	127,463	1.0	1.0	144	0.2	0.2	132	1.0	1.0	3	0.8	0.2	2
Syn3	0.1	20	11,699	52,897	0.5	0.5	97	0.5	0.5	2	0.5	0.5	0	0.5	0.5	270

Ale, J. M. & Rossi, G. H. (2000), An approach to discovering temporal association rules, *in* J. Carroll, E. Damiani, H. Haddad & D. Oppenheim, eds, '2000 ACM Symposium on Applied Computing', Vol. 1, ACM, Como, Italy, pp. 294–300.

Brijs, T., Swinnen, G., Vanhoof, K. & Wets, G. (1999), The use of association rules for product assortment decisions: a case study, *in* S. Chaudhuri & D. Madigan, eds, 'Fifth International Conference on Knowledge Discovery and Data Mining', ACM Press, San Diego, USA, pp. 254–260.

Ceglar, A. & Roddick, J. F. (2006), 'Association mining', *ACM Computing Surveys* **38**(2).

Chan, K. C. C. & Au, W.-H. (1997), Mining fuzzy association rules, *in* '6th International Conference on Information and Knowledge Management', Las Vegas, Nevada.

Cheung, D., Han, J., Ng, V. & Wong, C. (1996), Maintenance of discovered association rules in large databases: an incremental updating technique, *in* S. Su, ed., '12th International Conference on Data Engineering (ICDE'96)', IEEE Computer Society, New Orleans, Louisiana, USA, pp. 106–114.

Cheung, D. W., Ng, V. T. & Tam, B. W. (1996), Maintenance of discovered knowledge : A case in multi-level association rules, *in* E. Simoudis, J. Han & U. Fayyad, eds, '2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)', AAAI Press, Menlo Park, CA, USA, Portland, Oregon, pp. 307–310.

Fortin, S. & Liu, L. (1996), An object-oriented approach to multi-level association rule mining, *in* '5th International Conference on Information and Knowledge Management, CIKM'96', ACM, New York, Rockville, Maryland, USA, pp. 65–72.

Geng, L. & Hamilton, H. J. (2006), 'Interestingness measures for data mining: A survey', *ACM Computing Surveys* **38**(3).

Gupta, G. K., Strehl, A. & Ghosh, J. (1999), Distance based clustering of association rules, *in* 'Intelligent Engineering Systems Through Artificial Neural Networks, ANNIE 1999', ASME, St. Louis, Missouri, USA, pp. 759–764.

Han, J. & Fu, Y. (1995), Discovery of multiple-level association rules from large databases, *in* U. Dayal, P. Gray & S. Nishio, eds, '21st International Conference on Very Large Data Bases, VLDB'95', Morgan Kaufmann, Zurich, Switzerland, pp. 420–431.

Han, J., Koperski, K. & Stefanovic, N. (1997), Geominer: A system prototype for spatial data mining, *in* J. Peckham, ed., 'ACM SIGMOD International Conference on the Management of Data, SIGMOD'97', ACM Press, Tucson, AZ, USA, pp. 553–556.

Han, J. & Pei, J. (2000), 'Mining frequent patterns by pattern growth: Methodology and implications', *SIGKDD Explorations* **2**(2), 14–20.

Han, J., Pei, J. & Yin, Y. (2000), Mining frequent patterns without candidate generation, *in* W. Chen, J. Naughton & P. Bernstein, eds, 'ACM SIGMOD International Conference on the Management of Data (SIGMOD 2000)', ACM Press, Dallas, TX, USA, pp. 1–12.

Kohavi, R., Brodley, C., Frasca, B., Mason, L. & Zheng, Z. (2000), 'Kdd-cup 2000 organizers' report: Peeling the onion', *SIGKDD Explorations* **2**(2), 86–93.

Koperski, K. & Han, J. (1995), Discovery of spatial association rules in geographic information databases, *in* '4th International Symposium on Large Spatial Databases', Maine, pp. 47–66.

Kuok, C., Fu, A. & Wong, M. H. (1998), 'Mining fuzzy association rules in databases', *ACM SIGMOD Record* **27**(1), 41–46.

Lent, B., Swami, A. & Widom, J. (1997), Clustering association rules, *in* A. Gray & P.-A. Larson, eds, '13th International Conference on Data Engineering', IEEE Computer Society Press, Birmingham, UK, pp. 220–231.

Li, Y., Ning, P., Wang, X. S. & Jajodia, S. (2003), 'Discovering calendar-based temporal association rules', *Data and Knowledge Engineering* **44**(2), 193–218.

Ong, K. H., Ong, K. L., Ng, W. K. & Lim, E. P. (2002), Crystalclear: Active visualization of association rules, *in* 'International Workshop on Active Mining (AM-2002) in Conjunction with the IEEE International Conference on Data Mining (ICDN'02)', IEEE Press, Maebashi City, Japan.

Pasquier, N., Bastide, Y., Taouil, R. & Lakhal, L. (1999), Discovering frequent closed itemsets for association rules, *in* '7th International Conference on Database Theory (ICDT99)', Springer, Jerusalem, Israel, pp. 398–416.

Perrizo, W. & Denton, A. (2003), Framework unifying association rule mining, clustering and classification, *in* 'International Conference on Computer Science, Software Engineering, Information Technology, e-Business, and Applications (CSITeA03)', Rio de Janeiro, Brazil.

Rainsford, C. & Roddick, J. F. (1999), Adding temporal semantics to association rules, *in* J. Zytkow & J. Rauch, eds, '3rd European Conference on Principles of Knowledge Discovery in Databases, PKDD'99', Vol. 1704 of *LNAI*, Springer, Prague, pp. 504–509.

Roddick, J. F., Spiliopoulou, M., Lister, D. & Ceglar, A. (2008), 'Higher order mining', *SIGKDD Explorations* **10**(1), 5–17.

Shen, L. & Shen, H. (1998), Mining flexible multiple-level association rules in all concept hierarchies, *in* G. Quirchmayr, E. Schweighofer & T. Bench-Capon, eds, '9th International Conference on Database and Expert Systems Applications, DEXA'98', Vol. 1460 of *LNCS*, Springer, Vienna, Austria, pp. 786–795.

Srikant, R. & Agrawal, R. (1995), Mining generalized association rules, *in* U. Dayal, P. Gray & S. Nishio, eds, '21st International Conference on Very Large Data Bases, VLDB'95', Morgan Kaufmann, Zurich, Switzerland, pp. 407–419.

Teng, C. M. (2002), Learning from dissociations, *in* Y. Kambayashi, W. Winiwarter & M. Arikawa, eds, '4th International Conference on Data Warehousing and Knowledge Discovery (DaWaK '02)', Vol. 2454 of *LNCS*, Springer, Aix-en-Provence, France, pp. 11–20.

Zaki, M. (2000), Generating non-redundant association rules, *in* '6th International Conference on Knowledge Discovery and Data Mining, (SIGKDD'00)', AAAI PressACM, Boston, MA, USA, pp. 34–43.

Zaki, M. & Ogihara, M. (1998), Theoretical foundations of association rules, *in* '3rd SIGMOD'98 Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD'98)', Seattle, WA, USA, pp. 85–93.

Data Cleaning and Matching of Institutions in Bibliographic Databases

Jeffrey Fisher[1] **Qing Wang**[1] **Paul Wong**[2] **Peter Christen**[1]

[1] Research School of Computer Science
[2] Office of Research Excellence, The Australian National University,
Canberra, ACT 0200
Email: Jeffrey.Fisher@anu.edu.au

Abstract

Bibliographic databases are very important for a variety of tasks for governments, academic institutions and businesses. These include assessing research output of institutions, performance evaluation of academics and compiling university rankings. However, incorrect or incomplete data in such databases can compromise any analysis and lead to poor decisions and financial loss. In this paper we detail our experience with an entity resolution project on Australian institution data using the SCOPUS bibliographic database. The goal of the project was to improve the entity resolution of institution data in SCOPUS so it could be used more effectively in other applications. We detail the methodology including a novel approach for extracting correct institution names from the values of one of the attributes. Along with the results from the project we present our insights into the specific characteristics and difficulties of the Australian institution data, and some techniques that were effective in addressing these. Finally, we present our conclusions and describe other situations where our experience and techniques could be applied.

Keywords: Data Matching, Bibliographic Databases, Deduplication, SCOPUS.

1 Introduction

Bibliographic databases are being used across an increasingly broad range of areas. From allocating research funding by governments, to quantifying connections between academics and institutions to determining academic promotions (Christen 2012). To support these applications, it is vital that bibliographic databases are correct, cleaned and well maintained. However, far too often, it is up to individual companies or researchers to enter their own work into these databases (Lee et al. 2007). Alternatively, many bibliographic databases are automatically created and updated which leads to a host of data integrity problems including multiple updates, missing entries, and differences in data quality and formats when drawing on different data sources (Lee et al. 2007). All these problems can compromise the quality of any analysis done on the databases, which can lead to poor decision making and the wasting of time and money.

In this paper we detail our experience and findings from a project attempting to improve the data quality of Australian institutions in the SCOPUS bibliographic database (Scopus 2009). We used a variety of established data cleaning and data matching techniques and refined them where necessary. We present an approach to extracting institution names from attribute values. We also capture and incorporate domain specific knowledge, and illustrate particular types of problems for data matching in bibliographic databases. While we developed our approach for a specific database, certain techniques and aspects of the domain knowledge could be generalised to other bibliographic databases, and potentially other application areas.

The structure of this paper is as follows: in Section 1 we provide some background on the applications of bibliographic databases and the project goals. In Section 2 we describe the main features of the SCOPUS bibliographic database that was used in this project. In Section 3 we examine data cleaning and we describe our technique for extracting institution names from the databaseand in Section 4 we discuss the two aspects of the data matching in the project, merging institution identifiers where they correspond to the same institution, and determining an institution identifier for records that do not have one. Finally, in Section 5 we present our conclusions, a discussion of other areas these techniques could prove useful, and some possibilities for extending this work.

1.1 Bibliographic Databases

Bibliographic databases such as SCOPUS (Scopus 2009) have a wide variety of applications for governments, academic institutions and businesses. Governments use them for policy development including assessing future areas of research need and allocating research funding. They also use them to evaluate research and program performance. For example, in Australia, the Commonwealth Government runs the Excellence in Research for Australia (ERA) program to assess the research performance of academic institutions. The ERA program relies on measures such as citation counts and article counts from the SCOPUS database (ERA 2012). The ERA program also determines the funding allocations for part of the Sustainable Research Excellence in Universities program (ERA 2012).

Academic institutions such as universities also use bibliographic databases for a wide variety of tasks. Analysis of collaboration data in bibliographic databases allows universities to develop strategic partnerships and assists in identifying research and

funding opportunities. Additionally, performance evaluation of academic personnel can also be based on research output. This can be captured through measures such as the h-index, which attempts to quantify research output and quality, and is calculated from citations counts in bibliographic databases (Hirsch 2005).

Bibliographic databases are also used for commercial applications. An example is the industry that has developed around ranking universities, and scales such as the Times Higher Education ranking (`http://www.timeshighereducation.co.uk`) rely on data from bibliographic databases such as Thomson Reuters Web of Science to create their rankings.

1.2 Project Goals

Given the variety of applications of bibliographic databases it is important the data quality within them is high. To that end, the goals of this project were twofold. Firstly, it was to improve the accuracy of institution identifiers within the SCOPUS database. Secondly, this project formed part of an ongoing program of work to allow the Office of Research Excellence at the Australian National University (ANU) to better understand the SCOPUS database, and build organisational capacity for analysing and applying the data. These two goals could have a variety of practical benefits. For example, in order to analyse collaborations between institutions it is important that the institutions themselves be correctly identified. Similarly, if institutions are incorrectly identified when assessing research performance, then this may result in an inequitable allocation of funding. Improving the quality of institution data would assist with these and other applications and the lessons learned from this project could assist with similar analysis in the future.

SCOPUS contains a variety of different institution types. Large academic institutions such as universities produce the majority of research articles in SCOPUS, but there are many smaller research labs, companies, government departments, and even private individuals who also conduct research. Since the larger institutions are more relevant in most analysis, we only considered institutions that had at least ten records in SCOPUS.

In some cases it is unclear exactly what constitutes an institution. For the purposes of this project, we generally use the highest level organisation as the institution. For example, each separate division of a university could potentially be considered an institution, however the goal of the project was to identify all such divisions with the university itself.

2 The SCOPUS Database

The SCOPUS database is a bibliographic database containing approximately 80 million author records, and approximately 40 million academic papers, conference proceedings, theses, and other works. The snapshot used in this project covers the period 1996 to 2011. The SCOPUS database is stored in XML format with a tree schema and with a paper or article as the root of each record. The schema is proprietry so cannot be provided here. For ease of analysis, portions of the data were extracted into relational database tables containing information about entities such as authors or papers, but this does not reflect the underlying structure of the SCOPUS database. In addition, SCOPUS uses an automated data collection process which sources data from many places and in a variety of formats. This can lead to variations in data and storage formats. For example, the attributes *city*, *state* and *postcode* were all completely blank for the Australian data, despite the fact that they are appropriate. This information is often present in a record, but is usually part of the *organization* attribute, along with other information.

Since we were primarily interested in the Australian portion of the data, the *country* attribute was used to separate the data and only records with a *country* value of "aus" (correspoding to Australia), were used in the project. This reduced the dataset to 1,611,172 records. A short summary of the attributes and characteristics for the Australian records is provided in Table 1 below. The completeness column describes the percentage of records that had a non-null value for the attribute.

Attribute	Unique Values	Completeness
afid	28,306	98.2%
dptid	52,879	73.7%
organization	238,460	99.3%
city group	28,889	95.3%
address	32,038	25.1%
city	0	0.0%
state	0	0.0%
postcode	0	0.0%

Table 1: Characteristics of the SCOPUS Database.

The overall data quality was reasonably good, and the presence of the *afid* attribute, which appeared to be an identifier, was promising. However the large number of missing values reduced the usefulness of many of the attributes. From a data matching perspective, the most useful attributes appeared to be *afid*, *country* and *organization*. We examined each of these in more detail.

2.1 Attribute *afid*

According to the SCOPUS documentation (Scopus 2009), *afid* is intended to be a unique identifier for institutions. However, there were many missing values so it was not a strict primary key in the relational database sense (Elmasri & Navathe 2011). In addition, while its description indicated that it was unique, there were institutions in the database that had multiple different *afid* values.

We found that a single institution having multiple *afid* values was more common in smaller institutions. Given that bibliographic databases are important for the funding of large research institutions, they are more likely to have internal personnel making sure that research papers and journals are present in SCOPUS and correctly attributed to their institutions. In the event they determine that papers are missing or incorrectly attributed, they can notify SCOPUS to get the problem rectified. Smaller institutions are probably less likely to take these steps.

The *afid* attribute was extremely important for the data matching process. However, since it was not a perfect key and there was not a unique institution name for each *afid*, extracting a single institution name from the records for each *afid* was a significant challenge.

2.2 Attribute *organization*

From a data matching perspective this attribute was the most useful in determining the correct name for each institution. Of the attributes containing information about institutions, *organization* is the most complete with over 99% of Australian records having a value in this attribute. In addition, candidates for the institution name were often included in this attribute. It was also the best candidate for string comparison and other data matching techniques. However, the values of the *organization* attribute contained many abbreviations and acronyms that needed to be dealt with prior to the data matching as will be discussed in Section 3.3.

2.3 Attribute *country*

As described above, we used the *country* attribute to limit the data to manageable quantities. In addition, collecting all records with the same value of *country* together meant that some of the problems of trying to match across languages were removed. However, even something as straightforward as country was still not perfect. For example, there were hundreds of records with a *country* value for Australia that pertained to institutions in Austria, such as the University of Innsbruck and the University of Salzburg.

2.4 Data Summary

In summary, the most important attributes for the project were *country*, which was used to limit the data, *afid*, which was a quasi-identifier for institutions, and *organization* that contained text information about the institutions. Of the selected attributes, *organization* needed the most pre-processing to be useful in the data matching process. This involved extracting the institution names in an automated fashion as well as dealing with abbreviations and acronyms. This data cleaning process is dealt with in the next section.

3 Data Cleaning

Data cleaning is an important aspect of almost all data matching exercises (Han et al. 2012). The main objective in the data cleaning step was to extract from the 1,611,172 records for Australian institutions a unique institution name for each of the 2,910 *afid* values with more than 10 records in SCOPUS. In addition, acronyms, abbreviations and stop words could make data matching more difficult. As part of the data cleaning process we generated a list of possible acronyms and abbreviations for Australia, along with their likely expansions. Each of these data cleaning tasks is discussed separately.

3.1 Institution Names

One of the biggest challenges for the project arose from the fact that although the attribute *afid* was intended to uniquely identify an institution, there was not a one-to-one relationship between *afid* and institution name. However, for *afids* with many individual records (10 or more) a large part of the variability amongst values of the *organization* attribute often came from capturing subdivisions of the institution. As a result, the institution name appeared to be the most frequently occurring substring. This formed the basis for our approach to extracting institution names.

3.2 Research Hypothesis

Building on some of the ideas presented by Ciszak (Ciszak 2008) and after experimentation, we determined the following hypothesis: the most frequently occurring comma-separated substring within the values of the *organization* attribute was the best candidate for the correct institution name. The ratio between the frequency of the most common substring and the second most common substring was an indicator of the likelihood that the name was correct.

To illustrate this we provide the following example which shows the name extraction process and ratio calculation for the ANU. The first step was to select the 63,389 records that had the *afid* value for the ANU. For each of these records we separated the values of the *organization* attribute into comma-separated substrings and these substrings were counted to create a frequency table. We present the four most frequent comma-separated substrings for the ANU in Table 2 below:

Substring	Frequency
Australian National University	29,311
Research School of Chemistry	5,483
Research School of Physical Sciences and Engineering	4,485
John Curtin School of Medical Research	4,447

Table 2: Substring Frequencies for the Australian National University.

The most frequently occurring substring was used for the institution name, which in this case was "Australian National University" as expected. The ratio was then calculated by the following formula:

$$Ratio = Highest\ Freq./2nd\ Highest\ Freq.$$

For the above example, this gives:

$$Ratio = 29,311/5,483 = 5.35$$

A ratio of 5.35 was relatively high and as a result we had good confidence that the name was correct. We provide some examples of extracted institution names in Table 3 below. We also provide a more detailed analysis of the relationship between name correctness and the ratio value in Section 3.5.

3.3 Acronyms and Abbreviations

In addition to extracting institution names, it was important to replace frequent acronyms and abbreviations by their expanded expressions. This improved the results of the institution name extraction, especially where the acronyms were common. In addition, during the subsequent data matching, we needed to match against common acronyms of large institutions when trying to deal with records that had no *afid* value, in case they only contained the acronym in their value for the *organization* attribute.

We used a look-up table that specified the most common acronyms along with their expanded forms. However, particularly amongst the abbreviations, there were many that had multiple possible expansions, for example "Med." could be "Medical" or "Medicine". Because of this, there were several common abbreviations that were left in their unexpanded form.

Extracted Name	Correct Name	Ratio	Notes
Australian Institute Marine Science	Australian Institute of Marine Science	18.05	None
Royal North Shore Hospital	Royal North Shore Hospital	11.98	None
Calvary Hospital	Calvary Hospital	7.00	Kogarah, N.S.W.
University Queensland	University of Queensland	6.05	None
Western Australian Institute Sport	Western Australian Institute of Sport	4.67	None
URS Australia Pty Limited	URS Australia Pty Limited	3.50	Contains Acronym
School Chemistry	Monash University	2.60	Incorrect Name
Ipswich Hospital	Ipswich Hospital	1.80	None
University Notre Dame	University of Notre Dame	1.35	None
School Psychiatry	Unknown	1.21	Incorrect Name
Innsbruck Medical University	Innsbruck Medical University	1.14	In Austria
Suite 3	Melbourne Heart Centre	1.00	Incorrect Name

Table 3: Examples of Extracted Institution Names and Ratio Values.

3.4 Stop Words

We also removed stop words such as "the", "of" and "for". Because these words contain little information, they are sometimes left out (Christen 2012) and removing them further standardised the institution names. As with abbreviations and acronyms, a look-up table was used to remove them from the values of the *organization* attribute.

3.5 Data Cleaning Results and Discussion

In this section, we present the overall results and analysis of the data cleaning. The main focus of the data cleaning was extracting a unique name for each *afid* value. We also examined whether the calculated ratio value influenced the likelihood that a name was correct.

To test the methodology, a random selection of 200 *afid* values were picked and we used our approach to extract an institution name. All of the *afid* values had at least 10 records in SCOPUS. To deal with acronyms and abbreviations, we used a manually created lookup table containing 73 acronyms and abbreviations along with their expansions. They were predominantly the most frequently occurring acronyms and abbreviations where there was little ambiguity as to what the correct expansion was. We also removed the most common stop words ("of", "and", "for", "the", "in", "at" and "on"). The results are displayed in Table 4 below:

Result	Number of *afid*s	Proportion
Correct Name	131	65.5%
Partially Correct Name	42	21.0%
Incorrect Name	27	13.5%
Total	200	100.0%

Table 4: Data Cleaning Results.

Names were judged to be correct if they could identify the institution in question. Names were judged partially correct if they contained abbreviations or acronyms that had not been expanded or were missing a word. In general, where the name was correct, but was for a smaller part of a large institution, we deemed it an incorrect result. For example one *afid* was assigned the name "Research School of Social Sciences" but this was deemed incorrect since it was a subdivision of the Australian National University. The only exceptions were if it appeared to be a separate research centre or similar, in which case we deemed it partially correct. There was a certain level of judgment involved, however these cases were fairly few in number. For an approach that was simple and easy to implement, the overall results were reasonably promising with a correct or partially correct name extracted for 86.5% of the *afid* values.

In order to test the hypothesis that the ratio between the two most common substrings was an indicator of how likely a name was to be correct, we conducted a more detailed analysis. The results of the 200 sampled *afid* values were broken into categories based on the calculated ratio value. Table 5 below shows how the ratio affects the quality of the names generated. We counted partially correct matches as correct for this analysis.

Ratio	% Correct	% Incorrect
Ratio ≥ 4.0	98.3%	1.7%
$2.0 \leq$ Ratio < 4.0	86.9%	13.1%
$1.5 \leq$ Ratio < 2.0	88.0%	12.0%
$1.2 \leq$ Ratio < 1.5	85.0%	15.0%
$1.1 \leq$ Ratio < 1.2	80.0%	20.0%
Ratio < 1.1	60.0%	40.0%
Total	86.5%	13.5%

Table 5: Effect of Ratio on Name Correctness.

As predicted by our hypothesis, a higher ratio was a general indicator of name correctness. However, there was a significant difference between ratio values above 4.0 and ratio values below 4.0. In applications where incorrect matches would be a significant problem, then excluding everything with a ratio below 1.1 or 1.2, would reduce the incorrect names without massively lowering the coverage. For this project we retained the names extracted for all 2,910 *afid* values that had more that 10 records in SCOPUS.

We present some possible ways of improving this methodology in Section 5, along with other situations where this technique could be applied.

4 Data Matching

Data matching is the task of identifying, matching, and merging records that correspond to the same entities from one or more databases (Christen 2012). For this project, there were two main parts to the data matching process: determining which different *afid* values corresponded to the same institution and could be merged, and assigning an *afid* value to records that did not have one.

The data matching was performed using different string comparison techniques on the 2,910 institution names that were extracted during the data cleaning. For each step of the data matching, we set a minimum similarity threshold. Comparisons that returned a similarity score above this threshold were *positive matches*. Comparisons that returned a similarity score below this threshold were *non-matches*.

The positive matches fell into two categories, *true positives* and *false positives*. Matches were *true positives* if the two values matched actually corresponded to the same real world institution. Matches were *false positives* if the two values corresponded to different real world institutions. In order to determine whether matches were true positives or false positives, we conducted a manual evaluation. Non-matches could also be divided into *true negatives* and *false negatives*. However for non-matches the vast majority were true negatives and so we were unable to manually review a sufficient number of non-matches to accurately estimate the number of true negatives and false negatives.

Since we could not determine the actual number of true negatives and false negatives we were unable to calculate recall and accuracy. As a result, we used precision to evaluate the data matching results. Precision is calculated as follows (Han et al. 2012):

$$Precision = true\ positives / positive\ matches$$

True positives and *positive matches* are as defined above.

Transitive closure was also a potential problem. Transitive closure refers to the situation where if three records, "a", "b" and "c" are compared to each other pair-wise and "a" matches to "b" and "a" matches to "c" then "b" should also match to "c" (Christen 2012). However, in practice this is not guaranteed and for this project it was an issue we had to resolve. We dealt with this slightly differently for each part of the data matching so it is discussed in the next sections.

A number of different string comparison techniques were used in the data matching. Each technique takes two strings as input and returns a similarity value between 0 and 1. A result of 0 indicates the strings are completely different (what constitutes completely different varies depending on the technique). Higher values between 0 and 1 indicate more similar strings. If the two strings are identical the result will be 1. However, for some techniques different strings may also give a result of 1.0 (Christen 2012). A brief description of the techniques that were used in this project is provided below (Christen 2012).

Exact: exact matching returns either 0 or 1, with 0 indicating different strings and 1 indicating the strings are identical.

Q-gram: q-gram string matching splits the two input strings into substrings of length q using a sliding window approach, and then measures the proportion of substrings that are common to both of the original strings.

Jaro: Jaro comparison uses a sliding window approach and measures the number of characters the two strings have in common in this window and also takes into account the number of transpositions.

Longest common substring (LCS): LCS comparison iteratively removes the longest common substring from each of the two strings down to a minimum length and then computes a similarity measure based on the proportion of the strings that have been removed.

Bag distance: bag distance counts the number of characters the two strings have in common by converting them each into a multiset and then subtracting one from the other.

The data matching code was written in Python 3.2 and used the Febrl library (Christen 2009) for the string comparison techniques. The code was run on an I7 2600, 3.4Ghz machine with 16 Gigabytes of memory running the Windows 7 operating system. The majority of techniques had running times of 15 minutes or less when calculating similarities and clustering institutions. However, the LCS comparisons took longer, with running times of up to two hours. Since we only ran each technique a small number of times, this was not a significant issue, but if they needed to be run repeatedly or with larger data sets, then alternative languages or libraries could be investigated, along with a possible parellelisation of the algorithm.

While there were similarities between the two different data matching tasks, matching between *afid* values and determining an *afid* value for records that did not have one, they each had unique characteristics so are treated separately.

4.1 Matching Between *afid* Values

The purpose of data matching between different *afid* values was to determine where they corresponded to the same institution so they could be merged together.

The approach used an agglomerative hierarchical clustering technique (Han et al. 2012, Naumann & Herschel 2010). Initially, each *afid* was assigned to its own cluster, and each cluster was also given the name we had extracted for the *afid* as a second attribute. The data matching compared clusters using the name attribute and merged clusters where the similarity score between the names was above the assigned threshold.

To deal with transitive closure, we conducted pair-wise matching between clusters and recorded all successful matches. All clusters where there was a successful match between the names were merged. In some cases two clusters were merged even though the similarity score between their names was below the threshold, for example when they both successfully matched with a third cluster. The evaluation examined all pair-wise matches from merged clusters, even where individual matches were below the required similarity threshold.

The data matching was conducted iteratively. After each step of the data matching, clusters were merged where they had been matched successfully and a new comparison technique was tried, generally with a lower similarity threshold. The initial techniques were exact matching with a threshold of 1.0 and q-gram matching with a threshold of 0.9. Several techniques were tested for the third step. The results of the data matching are described in Table 6 below.

4.1.1 Evaluation

We provide a brief description of the results of each iterative matching step, along with specific examples

Comparison Technique	Similarity Threshold	Other Parameters	Clusters Matched	Clusters Formed	Precision
Exact matching - step 1	1.00	None	566	230	85.9%
Q-gram matching - step 2	0.90	q = 2	75	35	87.0%
Jaro - step 3	0.80	None	1,332	165	< 50%
Jaro - step 3	0.90	None	169	63	< 50%
LCS - step 3	0.80	Shortest length = 3	465	160	< 50%
LCS - step 3	0.90	Shortest length = 3	69	34	72.2%

Table 6: Data Matching Results (part 1). Note that some comparison techniques yielded precision results that were clearly less than 50% and were not fully evaluated.

where they are relevant. We started with 2,910 clusters, corresponding to the 2,910 *afid* values that we extracted a name for in the data cleaning. Exact matching merged clusters if they had exactly the same name. This matching technique merged 566 clusters down to 230 new ones representing a reduction of 336 clusters, which was 11.5% of the initial 2,910.

To evaluate precision, a random selection of 100 new clusters was reviewed. Because in some cases three or more clusters were merged into a single new one, there were more than 100 matches to evaluate. Of the 220 pairwise matches, 189 or 85.9% were correct. Of the 100 clusters sampled, 88 were completely correct, i.e. every cluster merged was part of the same institution, one was partially correct where two of the clusters were actually the same institution, and the third was different, and 11 were incorrect with none of the matched clusters referring to the same institution. Of the 12 clusters that were incorrect or partially incorrect, four of them had the correct name for the institutions, but it was a common name, e.g. "Calvary Hospital", while in the eight other cases, at least one of the institution names was incorrect.

The second technique was q-gram matching with a similarity score of 0.9 and a q-value of 2. This largely resolved institutions that were present multiple times, but with minor variations in their names. This technique merged 75 clusters down to 35, which was a reduction of 40, or 1.4% of the initial 2,910. Since there were fewer than 100 new clusters they were all evaluated.

Of the 46 pair-wise matches that were generated in this step, 40 of them were correct which is a precision of 87.0%. Of the 35 values formed, 29, or 82.9% were completely correct and 6 were completely incorrect. It is worth noting that this step combined several different divisions of the "Commonwealth Scientific and Industrial Research Organisation" (CSIRO) into a single cluster, and these were treated as correct matches. This occurred because the expansion of the acronym "CSIRO" is so long that it heavily skews the similarity scores when conducting string comparisons. We discuss this further in our conclusion.

Of the techniques tested in the third step, only LCS with a similarity threshold of 0.9 had a precision that was reasonable (greater than 70%) and even for these matches, the increase in incorrect matches would not be justified in many applications. In addition, some comparison measures gave results where the precision was clearly less than 50% were not further evaluated. The results from these techniques indicated that we might be reaching the limits of what could be achieved with string comparisons.

To try and assess the number of true matches that remained, a matching round was explored using q-gram matching and a low similarity score of 0.75 and q = 2.

The overall precision with this approach was extremely low and a few clusters that contained many institutions with similar names dominated the results. There were some true positives in the matches. It was difficult to gauge exactly how many more matches could be obtained with perfect string comparison techniques, but it is probably in the vicinity of 150 to 200. This suggests we had discovered approximately two thirds of the true matches. However, this is not counting any matches between cases where afids have completely different names but correspond to the same institution. These are unlikely to be picked up through string comparison techniques and we provide some suggestions to deal with these in our section on future work.

4.2 Records Without an *afid* Value

Out of the 1,611,172 records for Australian institutions, 29,184 or 1.8% had no value for *afid*. In some cases, this could be correct, since individuals who are not associated with an institution can perform research. However in other cases, these records had institution information present, usually in the *organization* attribute, and as a result it appeared that the blank value for *afid* was actually a data quality issue. We again used string comparison techniques to try and determine the correct value of *afid* for these records.

Of the 29,184 records for Australia that had no *afid* value, 10,376 also had no information in the *organization* attribute. These were excluded from the process since they had nothing to match against. This left 18,808 records on which to perform the data matching.

The same pre-processing steps were applied to the values of the *organization* attribute that were used when generating the institution names, such as expansion of acronyms and abbreviations and splitting the string into comma separated tokens. Both the tokens and the institution names were also converted to lower case to improve the quality of the matches.

Once the pre-processing was complete we tested different string comparison techniques and matched the tokens from the *organization* attribute for the records with no *afid*, against the institution names extracted in the data cleaning step. This was an iterative process and after each comparison technique we removed records with no *afid* that had been matched successfully from the data before trying the next comparison technique.

As when matching between *afid* values, the process began with exact matching with a threshold of 1.0, then q-gram matching with a threshold of 0.9 and then we experimented with a number of different techniques for the third step. The results of the data matching are described in Table 7 above.

Comparison Technique	Similarity Threshold	Other Parameters	Unique Records Matched	Precision
Exact matching - step 1	1.00	None	5,822	96.0%
Q-gram matching - step 2	0.90	q = 2	1,815	96.0%
Jaro - step 3	0.80	None	5,865	< 50%
Jaro - step 3	0.90	None	887	< 50%
LCS - step 3	0.80	Shortest length = 3	2,513	57.0%
LCS - step 3	0.85	Shortest length = 3	1,114	60.0%
LCS - step 3	0.90	Shortest length = 2	183	54.0%
Bag Distance - step 3	0.80	None	6,087	< 50%
Bag Distance - step 3	0.90	None	372	< 50%

Table 7: Data Matching Results (part 2). Note that some comparison techniques yielded precision results that were clearly less than 50% and were not fully evaluated.

4.2.1 Evaluation

We provide a brief summary of the results of each matching technique, along with some examples.

Exact matching checked whether one of the tokens in the organization attribute was exactly the same string as the name for an institution. This matched 5,822 records with no *afid* to an institution. This represented 31.0% of records without an *afid*, which was a higher proportion than expected. Our initial expectation was that records without an *afid* would have generally poor overall data quality.

However, transitive closure was a problem with 252 of the records matched receiving an exact match to two or more different institutions. In these cases an organization value had at least two comma-separated substrings and they had each matched exactly to two different institutions. For example "Department of Physics, University of Sydney" matched to both an institution called "Department of Physics" and an institution called "University of Sydney". The institution named "Department of Physics" likely has an incorrect name and the true match should be with the "University of Sydney". From the analysis it appeared that there was a fairly strong link between the confidence in the institution names in the data cleaning phase, and the likelihood that they were correctly matched. As a result, the ratio value calculated during the data cleaning stage was used as a tiebreaker when resolving transitive closure in these cases. An evaluation of 100 randomly selected matches gave a precision of 96.0%. For all 4 records that were incorrectly matched, the institution name was probably incorrect. Of the successful matches, a few large institutions that were frequently missing an *afid* were responsible for a large proportion of the total matches.

The second technique was q-gram matching with q = 2 and a similarity threshold of 0.9. This matched another 1,815 records. Transitive closure affected another 233 records and was dealt with as for exact matching. This step primarily matched records with minor name variations or typographical errors. An analysis of 100 random matches gave a precision of 96.0%. As with exact matching, a small number of institutions contributed the majority of the positive matches.

Unfortunately, for the third step of the matching, the results were not promising with none of the comparison techniques tried giving a good level of precision. As with exact matching and q-gram matching, we performed an evaluation on a random sample of 100 matched records. However, in many cases the matching quality was too poor to warrant a detailed analysis since it was clearly less than 50% precision. Of the techniques tested, only the LCS comparison with a minimum substring length of three gave a precision of 60%, and even this is generally too low to be useful.

Since the first two techniques had only matched approximately 40% of the 18,808 records, we performed a more detailed analysis of the results to determine why the match quality was so poor and detected three main causes for the poor precision.

Institutions where the extracted names were only partially correct or were incorrect had a significant effect on the results. In particular, a small number of institutions with a name that was similar to subdivisions of other institutions had a disproportionate impact on the number of incorrect matches. For example, one *afid* was assigned the incorrect name "Department of Medicine". This resulted in many records that contained substrings such as "Department of Renal Medicine" or "Department of Emergency Medicine" generating high enough similarity scores to achieve a false positive match.

Another problem was certain types of institutions with very similar names. For example, within Australia, many State Governments, as well as the Commonwealth Government, have a "Department of Primary Industries" and the string comparison techniques were not effective at distinguishing between them. This resulted in many records being assigned to the incorrect institution.

Finally, within the records that did not have an *afid*, a significant proportion did not have the institution name present. An evaluation of 100 records that were not matched by either the exact match or the q-gram matching found that in 40% of cases, the institution name did not appear to be present anywhere in the record. In another 13% of cases, the institution name was heavily abbreviated or shortened enough to make string matching difficult.

4.3 Data Matching Summary

Overall, the data matching led to mixed results. The process for matching between different *afid* values to determine whether they corresponded to the same institution was reasonably successful. We achieved precision of 85% or higher for the first two comparison techniques, and estimate very roughly that between them they accounted for around two thirds of the positive matches that could be found. This could be stretched a bit further using the LCS technique if a few more incorrect matches were acceptable for the end use case.

However, the matching to assign an *afid* value to records that did not have one was less successful. While the matches from the first two techniques were very good with a precision over 95%, the total coverage was only 40.6% of the records without an *afid* and after this no technique produced good results. There was a significant difference between the easy matches, and the more difficult ones. Once the records for the large institutions that were frequently missing an *afid* had been resolved, it was quite difficult to deal with the remainder.

5 Conclusions and Future Work

Overall, the project results were reasonably positive, and largely achieved the project goals, but there was still room for improvement. The data cleaning phase where we extracted institution names went well. For institutions with 10 or more records, we extracted a correct or partially correct institution name for 86.5% of *afids*. This alone was a very useful result from the project, since identifying the correct name for institutions in SCOPUS can be challenging and is often a limiting factor when using the data.

The merging of different *afids* was also reasonably successful with the exact matching and the first q-gram matching both having a precision of over 85% and between them reducing the number of afid values by 13%. The longest common substring comparison with a similarity threshold of 0.9 achieved precision over 70%, however in practice, the increase in incorrect matches may not justify the overall number of additional matches gained. An examination of the output with a low similarity threshold suggested that for the institutions where the names were correct approximately two thirds of the true matches had been detected.

Dealing with the records that did not have an *afid* was less successful. The initial steps were good with a few large institutions that were frequently missing an *afid* and were easy to resolve resulting in a precision over 95%. Between them they accounted for around 40% of the records. However, all subsequent techniques had very poor match quality.

We identified three common causes for the incorrect matches, including institutions with very similar or the same names, a small number of incorrect names extracted during the data cleaning phase resulting in a disproportionate number of incorrect matches, and many of the records not containing the institution name.

We detail in the section on future work some ways these issues can be addressed. Once this has been done to a satisfactory degree, the output from this project can be incorporated into the SCOPUS database and used to improve future analysis.

With respect to the second goal of improving organisational capacity with respect to SCOPUS, the project was also valuable. Two projects currently underway involve assessing the links between Australian institutions and those in Indonesia and India. The experience and knowledge gained from this project has been valuable for this analysis, particularly regarding the specifics of the SCOPUS database and the challenges present in the Australian institution data.

5.1 Applications and Domain Knowledge

There were many characteristics of the project that were unique to the SCOPUS database and which might not be applicable elsewhere. However, aspects of the domain knowledge could be useful in other data matching on Australian institutions or worldwide. In addition, the approach we used for name extraction and the ratio concept could be applied in other areas.

One of the biggest problems for the project was a result of the word "department" pertaining to both subdivisions of larger institutions, particularly universities, and also to institutions such as government departments. While extracting the institution names, there were a number of small institutions that incorrectly received names such as "Department of Medicine" and "Department of Physics". In most of these cases, the institution name was not actually present in any of the records, so it was impossible to tell what the institution actually was. However, when these names were used in the data matching, they caused significant problems, particularly when matching against records without an *afid* value, where they frequently caused false positive matches.

A few rules could be quite effective at resolving this problem. For example, given a country, it might be worthwhile to create a lookup table of the main government departments, and exclude any institution name that contains the word "department" which is not in that table. A small number of rules to deal with cases such as these could significantly improve the results.

In addition, an improved methodology for dealing with acronyms could also be worthwhile. As mentioned in the analysis, all the divisions of CSIRO were combined by the early string comparison techniques since the expanded form of CSIRO is so long that it dominates the matching. For CSIRO this was not a problem since they are all part of the same institution. However, similar situations occurred to a lesser degree with other long expansions such as CRC for Cooperative Research Centre, or NSW for New South Wales. In practice, it could be worthwhile to change the data cleaning approach to detect the expanded forms of acronyms, perhaps allowing slight variations, and then reduce them to their acronyms for data matching purposes, rather than the other way round. This would prevent these terms causing too many incorrect matches.

While we have not tested it elsewhere, there is no a priori reason why the frequency based approach that we used to extract institution names couldn't be applicable in other areas. In particular, any application where the domain is relatively small in relation to the number of records would be a good candidate for this approach. Examples could include suburb names for a country, or potentially company names or product names. Alternatively, the domain could be restricted to the larger examples, as we did in this project, in order for the technique to be used. For a simple and easy to implement technique, it was surprisingly effective.

The ratio concept could also be used in these situations as an indicator of confidence in the correctness of the result. A high ratio value indicates that there is only a single good candidate for the correct value, whereas a ratio value close to 1 indicates that there are two or more candidates for the correct value and it may be difficult to pick between them. This technique could be extended by creating a probability distribution from the results, rather than using the ratio value, which only considers the two most frequent values. Doing this could better capture the variability, especially if there are three or more candidates for the correct result. However, care would need to be taken in these situations to not overemphasise the impact of the low probability results. For example, in this project, very few institutions with a ratio value above 4.0 had an incorrect name ex-

tracted. However, for the Australian National University, the correct name only made up 23.7% of the comma-separated substrings. This situation was also common in other large institutions so using a probability distribution could risk more incorrect results rather than less.

Finally, small variations to the methodology would also be worthwhile in many practical applications. If coverage is less important for the analysis being undertaken, then for the SCOPUS data, accuracy could be increased to over 90% with a reduction in the coverage of approximately 12%. In practice this is probably a worthwhile tradeoff, since even a small number of incorrect names from the data cleaning step significantly increased the number of incorrect matches in the data matching. Similarly, modifying the technique to also incorporate the number of records could improve the result, since generally the large institutions were more likely to be correct.

5.2 Future Work

There are a number of ways this work could be extended in the future. The use of more sophisticated data matching techniques such as incorporating TF-IDF (Term Frequency - Inverse Document Frequency) (Christen 2012) could improve the quality of the matching, particularly for determining an institution for records without an *afid*. When dealing with institutions that had long names such as "Department of Natural Resources and Mines", where a small difference in the name is actually important from a data matching perspective, a TF-IDF approach could be quite effective. However, even these techniques would not assist in cases where the institution name is simply not present in the record.

Based on our evaluation of the results in the data matching phase, a few different situations were responsible for a large proportion of the incorrect matches, both when data matching between *afid* values and when trying to determine an *afid* value for records that did not have one. The creation of a small set of domain specific rules could significantly improve the quality of the institution name extraction, and the subsequent data matching.

Finally, a collective data matching approach (Christen 2012) that attempted to do data matching on articles, authors and institutions simultaneously might be very successful though it would also be complex and computationally intensive. The data could be treated as a network capturing links between articles, individuals and institutions with the weights of the links measuring the frequency of the connections. This type of approach could potentially handle missing values in the data, and would also be very good at dealing with situations where a few records had incorrect values.

References

Christen, P. (2009), 'Development and user experiences of an open source data cleaning, deduplication and record linkage system', *SIGKDD Explorations* **11**(1), 39–48.

Christen, P. (2012), *Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection*, Springer.

Ciszak, L. (2008), Application of clustering and association methods in data cleaning, *in* 'International Multiconference on Computer Science and Information Technology', IEEE, pp. 97–103.

Elmasri, R. & Navathe, S. B. N. (2011), *Database systems: models, languages, design, and application programming*, Pearson.

ERA (2012), 'Excellence in Research for Australia 2012 National Report'. Australian Research Council.

Han, J., Kamber, M. & Pei, J. (2012), *Data mining: concepts and techniques*, 3 edn, Waltham, MA: Morgan Kaufmann.

Hirsch, J. (2005), 'An index to quantify an individual's scientific research output', *Proceedings of the National Academy of Sciences of the United States of America* **102**(46), 16569–16572.

Lee, D., Kang, J., Mitra, P., Giles, C. L. & On, B.-W. (2007), 'Are your citations clean?', *Commununications of the ACM* **50**, 33–38.

Naumann, F. & Herschel, M. (2010), *An introduction to duplicate detection*, Vol. 3 of *Synthesis Lectures on Data Management*, Morgan and Claypool Publishers.

Scopus (2009), *Scopus Custom Data Documentation*, Elsevier, Amsterdam.

Smalheiser, N. R. & Torvik, V. I. (2009), 'Author name disambiguation', *Annual review of information science and technology* **43**(1), 1–43.

A Novel Framework Using Two Layers of Missing Value Imputation

Md. Geaur Rahman **Md Zahidul Islam**

Center for Research in Complex Systems (CRiCS), School of Computing and Mathematics
Charles Sturt University, Bathurst, NSW 2795, Australia.
Emails: {grahman, zislam}@csu.edu.au

Abstract

In this study we present a novel framework that uses two layers/steps of imputation namely the Early-Imputation step and the Advanced-Imputation step. In the early imputation step we first impute the missing values (both numerical and categorical) using existing techniques. The main goal of this step is to carry out an initial imputation and thereby refine the records having missing values so that they can be used in the second layer of imputation through an existing technique called DMI. The original DMI ignores the records having missing values. Therefore, we argue that if a data set has a huge number of missing values then the imputation accuracy of DMI may suffer significantly since it ignores a huge number of records. In this study we present four versions of the framework and compare them with three existing techniques on two natural data sets that are publicly available. We use four evaluation criteria and two statistical significance analyses. Our experimental results indicate a clear superiority of the proposed framework over the existing techniques.

Keywords: Data pre-processing; data cleansing; missing value imputation; EM algorithm; Decision Trees

1 Introduction

The existence of missing values in data sets is a common problem. Due to various reasons including human errors and misunderstanding, equipment malfunctioning, faulty data transmission, propagation and measurements, collected data often have missing or incorrect values (Rahman & Islam 2011, Farhangfar et al. 2008). If the data are collected through a survey then often we can have missing values just because of the existence of some survey questions that a user may not feel comfortable to answer. For example, even if the identity of a participant is protected still s/he may not feel comfortable to answer the questions that are related to the sensitive disease (such as HIV positive) or financial condition (Young et al. 2011). Various studies show that the amount of missing values can be approximately 5% or more unless an organization takes extreme care during data collection (Zhu et al. 2004, Maletic & Marcus 2000).

We consider a data set D_F as a two dimensional table where rows represents records $R = \{R_1, R_2, \ldots R_N\}$ and columns represent attributes $A = \{A_1, A_2, \ldots A_m\}$. The attributes can be either numerical (like 4 and 5.54) or categorical (like Canberra and Bathurst). Categorical values do not have any natural ordering in them. A numerical attribute has its domain $A_j = [low, up]$ where low is the lower limit and up is the upper limit. A categorical attribute $A_j = \{a_1, a_2, \ldots a_k\}$ has a domain with k (i.e. $|A_j| = k$) different values. The size of a data set $|D_F|$ or $|R|$ is N, which is the number of records. We consider that R_{ij} is the jth attribute value of the ith record. By "missing values" we mean that some of the R_{ij} values are missing/absent for various reasons. A missing value is denoted as $R_{ij} =?$. If a record $R_i \in R$ contains one or more missing values then we consider that $r_m \subset R_i$ $(m < M)$ is a $1 \times m$ vector having m number of attributes with missing values and $r_a \subset R_i$ $(a = M - m)$ is a $1 \times a$ vector having a number of attributes with available values.

The data sets, collected by the organizations, are typically used for various data mining processes. However, the performance of a data mining technique can significantly be disturbed due to the existence of missing or incorrect values in the data sets (Khoshgoftaar & Van Hulse 2005). Moreover, the presence of missing values in data sets can cause an inaccurate and non-sensible decision which may make the whole process of data collection and analysis useless for the users (Han & Kamber 2000).

Therefore, it is crucial to have an effective data preprocessing framework for dealing with missing values. One important data preprocessing task is the imputation of missing values. A number of techniques have been proposed for imputing missing values (Aydilek & Arslan 2013, Rahman & Islam 2011, Cheng et al. 2012, Schneider 2001, Zhu et al. 2011).

For dealing with missing values an early method just deletes the records having missing value/s (Derjani Bayeh & Smith 1999). However, the usability of the data sets for various statistical analyses can generally be reduced if the records, having missing values, are deleted from a small sized data set. Moreover, the results of the analysis can be misleading due to the use of a data set having insufficient number of records (Osborne & Overbay 2008).

Another early method uses the mean of all available values of an attribute for imputing missing values (Schafer & Graham 2002). However, it is shown that the mean imputation approach can often produce more misleading results (from data mining and statistical analysis) than the simple record deletion approach (Osborne & Overbay 2008).

For imputing missing value/s of a record an advanced technique called k-Nearest Neighbour Imputation (kNNI) (Batista & Monard 2003) first finds the (user-defined) k-most similar records (of the record having missing value/s). If the missing value belongs to a categorical attribute, the technique imputes the missing value by using the most frequent value, of the same attribute, within the k-Nearest Neighbour (k-NN) records. Otherwise for imputing a numerical value the technique makes use of the attribute mean value for the k-NN records. kNNI is a simple technique, and it performs better than

the normal mean/mode imputation technique which calculates mean/mode from a whole data set, instead of the horizontal segment having k-NN records (Liu et al. 2010). However, for a large data set the technique can be found expensive since it finds the nearest neighbours of each record having missing value/s (Batista & Monard 2003, Wu et al. 2008).

Instead of using k-NN records, a more advanced approach called EMI (Schneider 2001, Junninen et al. 2004) considers a whole data set for imputing numerical missing value/s. The technique utilizes the mean values of all numerical attributes, and the correlations between attributes having missing values and attributes having available values of a record in order to impute the numerical missing value/s of the record. However, EMI does not work for a numerical attribute having the same value in all records. Moreover, it is not useful for imputing a record where all numerical attribute values are missing.

Another recent technique called DMI (Rahman & Islam 2011) identifies that the correlations of attributes within a leaf of a decision tree (i.e. a horizontal segment of a data set) are higher than the correlations of attributes within a whole data set. Therefore, the technique first finds a set of horizontal segments, where the records in each segment are considered to be similar to each other, from a data set by using an existing decision tree algorithm such as $C4.5$ (Quinlan 1996), and then applies an EMI algorithm within each segment for imputing numerical missing value/s. It is shown that the imputation accuracy and the improvement of data quality through DMI are higher than through EMI (Rahman & Islam 2013*a*, 2011).

However, the imputation accuracy of DMI declines for a data set having higher missing ratios (see Figure 2). This is perhaps due to the approach of considering only the complete records (i.e. the records having no missing values at all) while building a decision tree in order to find horizontal segments for the application of EMI. It does not take the records having any missing values into consideration for building the tree and therefore for the application of EMI. If a data set has a huge number of missing values then DMI ignores many records resulting in having only a small number of records for building the tree and applying the EMI.

We argue that the imputation accuracy of DMI, on a data set having a huge number of records with missing values, can be improved if it considers the records having missing values during the creation of decision trees. Therefore, we propose a framework which imputes missing values of a data set by the improved use of the records having missing values. The framework consists of two imputation steps namely "Early-Imputation" and "Advanced-Imputation". In the Early-Imputation step we use an existing algorithm in order to initially impute the missing values, and in the Advanced-Imputation step we use DMI for final imputation in order to get better imputation accuracy.

In this study we use four difference versions of the proposed framework, namely "EDI", "ESI", "LDI" and "LSI" for imputing both numerical and categorical missing values. They use two layers of imputation: an early-imputation and an advanced-imputation. We also evaluate the performances of our proposed techniques by comparing them with three high quality existing techniques namely DMI, EMI and IBLLS in terms of four evaluation criteria namely co-efficient of determination (R^2), Index of agreement (d_2), root mean squared error ($RMSE$) and mean absolute error (MAE) on two real data sets namely Autompg and Yeast that are publicly available in the UCI machine learning repository (Frank & Asuncion 2010). For simulating missing values we use four missing patterns namely Simple, Medium, Complex and Blended, four missing ratios (1%, 3%, 5% and 10%), and two missing models namely Overall and Uniformly Distributed (UD). The initial experimental results indicate that our proposed technique EDI performs significantly better (based on 95% confidence interval analysis and statistical t-test analysis) than the existing techniques.

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

2 Background Study

For imputing missing values a number of techniques have been proposed recently (Aydilek & Arslan 2013, Olivetti de França et al. 2013, Dorri et al. 2012, Zhang et al. 2011, Zhu et al. 2011, Liew et al. 2011, Liu et al. 2010, Twala & Phorah 2010, Farhangfar et al. 2007, Cai et al. 2006, Kim et al. 2005, Li et al. 2004, Rahman & Islam 2013*b*). Three existing techniques namely "Expectation-Maximisation based Imputation (EMI)" (Junninen et al. 2004), "Iterative Bicluster based Local Least Square based Imputation (IBLLS)" (Cheng et al. 2012), and "Decision Tree and EMI based Imputation (DMI)" (Rahman & Islam 2011) are used in the experimentation of this study to compare them with our proposed techniques and thereby evaluate the performance of the proposed techniques. We also briefly discuss the EMI, IBLLS and DMI techniques here.

2.1 EMI

EMI (Schneider 2001, Junninen et al. 2004) uses correlations of the attributes having missing values and the attributes having available values of a whole data set for imputing numerical missing values of R_i. Let $Q = \sum_{aa}^{-1} \sum_{am}$ be a matrix in which $\sum_{aa}$ is the covariance matrix of available attribute values and $\sum_{am}$ is the covariance matrix of available and missing values. Also let μ_m and μ_a be the mean vectors of missing values and available values, respectively. Based on the mean vectors and the correlations, the technique then imputes the missing value (r_m) by using the following linear regression model (Schneider 2001).

$$r_m = \mu_m + (r_a - \mu_a)Q + e \quad (1)$$

where $e = [\mu_0 + H.Z^T]^T$ is a residual error in which μ_0 is a mean vector having zero value/s, H is a cholesky decomposition of the correlation matrix Q and Z is a vector having Gaussian random values (Muralidhar et al. 1999).

Similarly, EMI imputes all other records, of D_F, having missing values. Once the records, having missing values, are imputed the technique re-calculates the mean vectors (μ_m and μ_a) and the correlation matrix (Q). Using the re-calculated μ_m, μ_a and Q, EMI re-imputes the missing values of D_F. EMI repeats this process of imputation until the change of μ_m, μ_a and Q of two consecutive iterations is under a user-defined threshold.

2.2 IBLLS

Unlike EMI, IBLLS (Cheng et al. 2012) first finds k-nearest neighbour (k-NN) records of R_i from D_F for imputing numerical missing values of R_i. The value of k is determined automatically through applying a heuristic approach (Kim et al. 2005). Now let $A_{k\times a}$ be the matrix that contains values from the k-NN records for the attributes having available values (r_a) and $B_{k\times m}$ is the matrix that contains values from k-NN records for the attributes having missing values (r_m). Using A and B, the

method calculates the correlation matrix $Q_{m\times a}$ $(= B^T A)$ for the attributes having missing values and the attributes with available values.

Note that IBLLS imputes the missing values of R_i one by one. For imputing the jth missing value r_m^j of r_m, the technique finds a set of k-NN records of R_i from D_F by considering the correlation matrix Q with a weighted Euclidean distance measure (Cheng et al. 2012). It then partitions the k-NN records vertically by considering only the attributes having high correlation with r_m^j. IBLLS then finds the r_a^j, A^j and B^j for r_m^j from the data segment (Bi-Cluster) which is partitioned both horizontally and vertically. Finally, IBLLS imputes r_m^j by using the following regression model (Cheng et al. 2012).

$$r_m^j = r_a^j C^j \tag{2}$$

where C^j is the matrix that contains the regression coefficients that are obtained by minimising the following Least Square equation (Kim et al. 2005).

$$\underset{C^j}{\operatorname{argmin}} \left\| A^j C^j - B^j \right\|_2 \tag{3}$$

The solution of Equation (3) can be obtained as follows.

$$\hat{C^j} = \left(A^{j\dagger}\right)^T B^j \tag{4}$$

where $A^{j\dagger}$ is a pseudo inverse of A^j. Thus, the missing value r_m^j can be imputed as

$$r_m^j = r_a^j \left(A^{j\dagger}\right)^T B^j \tag{5}$$

Similarly, IBLLS imputes all other missing values (if any) of R_i, and all other records having missing values. The process of imputation is an iterative approach. For each iteration (from iteration 2), it calculates the Normalised Root Mean Squared Error ($NRMSE$) (Cheng et al. 2012) by comparing the imputed values of the current iteration with imputed values of the previous iteration. Once the $NRMSE$ value is under a user-defined threshold the technique stops the process.

2.3 DMI

For imputing both numerical and categorical missing values DMI (Rahman & Islam 2011) uses a decision tree algorithm and an Expectation-Maximization algorithm (EMI) (Schneider 2001). Since EMI uses the correlations of attribute values of a data set, and generally the correlations of attribute values within a leaf are higher than the correlations of attribute values within the whole data set (Rahman & Islam 2011), the imputation accuracy of DMI is expected to be better through applying EMI for the records within a leaf rather than for the records within the whole data set.

DMI first divides D_F into two sub data sets namely D_C and D_I. The data set D_C contains records without any missing values whereas D_I contains records with missing values. If an attribute $A_j \in A$ has missing values, the technique then builds a decision tree T_j from D_C through applying a decision tree algorithm such as $C4.5$ algorithm (Quinlan 1996) by considering the attribute A_j as the class attribute. For a numerical attribute A_j DMI first generalizes the values of the attribute into N_C categories, where N_C is the squared root of the domain size of A_j. Note that the output of each tree is a set of logic rules where each logic rule represents a leaf. For each logic rule L_{kj}, DMI generates a sub data set d_{kj} by assigning the records of D_C that satisfy the logic rule L_{kj}, where L_{kj} is the logic rule representing the kth leaf of the jth tree. DMI assigns each record $R_i \in D_I$ into a sub data set d_{kj} corresponding to the logic rule L_{kj} where the record R_i falls in L_{kj}.

Once the records are assigned to the sub data sets, DMI imputes the missing values in the sub data sets one by one. If a missing value R_{ij} is numerical then for imputing R_{ij} DMI uses the EMI algorithm (Junninen et al. 2004, Schneider 2001) within the sub data sets where R_{ij} belongs to. If R_{ij} is categorical then it considers the majority class value of the sub data set as the imputed value. A majority class value is the class value having the highest frequency, and a class value is the value of the attribute that is considered as the class attribute for building the tree (Quinlan 1993).

3 A Novel Imputation Framework

We present a novel imputation framework in four different versions namely "EDI", "ESI", "LDI" and "LSI". The framework uses two main steps/layers: an early-imputation step and an advanced-imputation step. In the early-imputation step we impute the missing values by using an imputation technique such as EMI (Junninen et al. 2004, Schneider 2001) or IBLLS (Cheng et al. 2012). In advanced-imputation step, we then apply DMI (Rahman & Islam 2011) on the early-imputed data set. Before we discuss the framework in details we first introduce the basic concepts.

3.1 Basic Concept

DMI divides a given data set D_F, having missing values, into two sub data sets namely D_C (having only records without missing values) and D_I (having only records with missing values). It uses the $C4.5$ algorithm on D_C in order to build the decision trees (DTs). DMI then applies the EMI technique on the records of each leaf of a tree. If the number of records within a leaf is big then the imputation accuracy is typically higher than when the number of records for the leaf is small. If a data set has high number of missing values then DMI creates a data set D_C with a small number of records resulting in a small number of records in the leaves of the tree built from D_C. When EMI is applied on these small number of records, it typically produces a poor quality imputation. It was reported that DMI performs better in a data set having low number of missing values than a data set having high number of missing values (Rahman & Islam 2011). We understand that one reason of the low accuracy of DMI is the existence of large number of records with missing values in D_F.

In many cases, the size of D_C can be very small compared to D_F. For example, Table 1 shows the number of records in D_F, D_I, and D_C for the Autompg and Yeast data sets (Frank & Asuncion 2010) in terms of the 10% missing ratio and "Blended" missing pattern. Note that the data sets are publicly available in the UCI machine learning repository (Frank & Asuncion 2010). The details about the simulation of missing values (i.e. the 10% missing ratio and "Blended" missing pattern) are discussed in Section 4.2. For the Autompg data set, only 79 out of 392 records are used in D_C (see Table 1).

Since the DTs are built on the small sized D_C, the knowledge extracted by the DTs may not be as useful as it could be if the DTs were built on the whole data set D_F. Therefore, DMI often fails to perform well for a data set having a large number of missing values. In order to explain this better, we consider here an example/toy data set

Table 1: The number of records in D_F, D_I, and D_C for the Autompg and Yeast data sets in terms of 10% missing ratio and "Blended" missing pattern.

Data set	Total Records in D_F	Number of records in D_I (having missing values)	Number of records in D_C (without missing values) used to build Decision Trees
Autompg	392	313	79
Yeast	1484	674	810

having 15 records and 4 attributes as shown in Table 2a. We then build a DT from the toy data set D_F by considering the attribute "Pos." as a class attribute (see Figure 1(a)) where a leaf of the DT displays the information on the number of records belonging to a value of the class attribute. In the figure a leaf is represented by a rectangle and a node is represented by a circle. We then artificially create some missing values in D_F (see Table 2b). The data set D_F is then divided into D_C (Table 2c) and D_I (Table 2d).

Note that D_I only contains the records having missing values. Even if the records (such as "R4", "R6", "R10", "R12" and "R14") have only a single missing value, they are taken out of D_C and placed in D_I. Thus, we often end up having a small number of records in D_C. Therefore in this example, the leaves of the DT (see Figure 1(b)) built on D_C do not provide any information on the the value "L" of the class attribute "Pos.". However, the DT (see Figure 1(a)) built on the full data set D_F provides information on the value "L". Now if we classify the records "R4", "R6", "R10", "R12" and "R14" of Table 2a by the DT (see Figure 1(a)) obtained from D_F then we get the correct class value "L", whereas the records are misclassified as "Ap" if we use the DT (see Figure 1(b)) that is obtained from D_C. We get a higher classification accuracy by the first DT than the second DT and therefore we expect a better imputation accuracy when we use the first DT than the second DT. That is, the removal of many useful records from D_C just because they have one/two missing values may not be a good idea.

Table 2: A toy data set D_F

Rec.	Age	Edu.	Salary	Pos.
R1	27	MS	85	L
R2	45	PhD	115	AP
R3	42	PhD	145	P
R4	25	MS	85	L
R5	50	PhD	146	P
R6	28	MS	85	L
R7	38	PhD	120	AP
R8	43	PhD	147	P
R9	44	PhD	146	P
R10	25	MS	86	L
R11	42	PhD	142	P
R12	26	MS	84	L
R13	42	PhD	143	P
R14	25	MS	86	L
R15	43	PhD	143	P

(a) A toy data set D_F (Original)

Rec.	Age	Edu.	Salary	Pos.
R1	27	?	?	?
R2	45	PhD	115	AP
R3	42	PhD	145	P
R4	?	MS	85	L
R5	50	PhD	146	P
R6	28	MS	85	?
R7	38	PhD	120	AP
R8	43	PhD	147	P
R9	44	PhD	146	P
R10	25	MS	?	L
R11	42	PhD	142	P
R12	26	?	84	L
R13	42	PhD	143	P
R14	?	MS	86	L
R15	43	PhD	143	P

(b) D_F with missing values

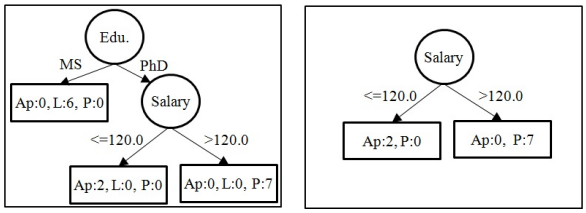
Rec.	Age	Edu.	Salary	Pos.
R2	45	PhD	115	AP
R3	42	PhD	145	P
R5	50	PhD	146	P
R7	38	PhD	120	AP
R8	43	PhD	147	P
R9	44	PhD	146	P
R11	42	PhD	142	P
R13	42	PhD	143	P
R15	43	PhD	143	P

(c) Data set D_C

Rec.	Age	Edu.	Salary	Pos.
R1	27	?	?	?
R4	?	MS	85	L
R6	28	MS	85	?
R10	25	MS	?	L
R12	26	?	84	L
R14	?	MS	86	L

(d) Data set D_I

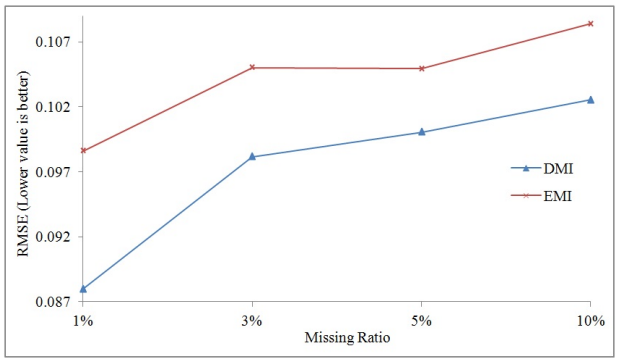
We also analyse the impact of the number of missing values in a data set, in terms of imputation accuracy ($RMSE$). Four missing ratios namely 1%, 3%, 5%, and 10% are used. Two publicly available data sets are used as shown in Figure 2. The missing values are imputed by



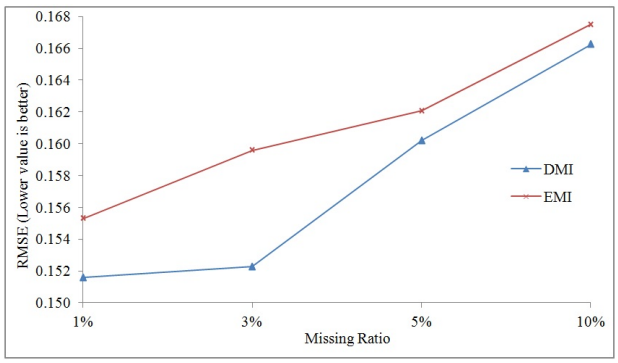
(a) Decision tree from D_F (Original) (b) Decision tree from D_C

Figure 1: Decision trees built by using D_F and D_C.

DMI (Rahman & Islam 2011) and EMI (Schneider 2001). Here, x% missing ratios means x% of the total attribute values (i.e. N records $\times$ M attributes) of a data set are missing. Figure 2 shows that DMI outperforms EMI in terms of $RMSE$ (the lower the better) on both data sets namely Autompg (see Figure 2(a)) and Housing (see Figure 2(b)). However, the imputation accuracies of both DMI and EMI drop (on both data sets) with the increase of the missing ratios. DMI performs significantly better than EMI for a data set having small missing ratio (see Figure 2(a) and Figure 2(b)).



(a) Autompg data set



(b) Housing

Figure 2: Performance comparison on the Autompg and Housing data sets in terms of $RMSE$ for different missing ratios.

We argue that if a data set has a large number of records with missing values, the imputation accuracy of DMI can be improved by taking the records of D_I (in a refined form) into consideration instead of totally ignoring them. Therefore, we propose a novel framework that imputes missing values of D_F by first refining the records of D_I and then using them in D_C. The framework therefore is a combination of two imputation steps/layers namely "Early-Imputation" and "Advanced-Imputation". We now discuss the steps in details as follows.

3.2 Early-Imputation

The main goal of this step is to refine the records (that go to D_I) that have one or more missing values, so that the records can be considered in D_C in order to increase the

imputation accuracy. The records are refined by performing an early imputation for them.

In this step, we first initialize a missing matrix Y which is then used in the Advanced-Imputation step for indicating whether a value is originally missing or available. Each element $y_{ij} \in Y$ ($(1 \leq i \leq N)$ and $(1 \leq j \leq |A|)$) contains either 0 or 1, which is calculated using Equation 6.

$$y_{ij} = \begin{cases} 1 & \text{if } R_{ij} \in D_F \text{ is missing} \\ 0 & \text{if } R_{ij} \in D_F \text{ is available} \end{cases} \quad (6)$$

The novel framework uses existing algorithms for imputing the missing values of D_F in this step. It first imputes the numerical missing values in D_F by using high quality techniques such as EMI (Junninen et al. 2004, Schneider 2001) and IBLLS (Cheng et al. 2012). It then imputes the categorical missing values of an attribute A_j of a record R_i. The framework first find k nearest neighbour (k-NN) records of R_i where the numerical missing values are already imputed. The mode value of the attribute A_j within the k-NN records is then considered as the imputed value. Based on the literature, the default value of k (for the k-NN) is set to 10 (Batista & Monard 2003, Bø et al. 2004, Troyanskaya et al. 2001).

3.3 Advanced-Imputation

The framework then applies DMI (Rahman & Islam 2011) on the early-imputed data set for the further improvement of the imputation quality for both the numerical and categorical missing values. It uses the matrix Y for identifying the missing values.

DMI builds a set of DTs $T = \{T_1, T_2, \ldots T_M\}$ where each tree T_i considers an attribute A_i as the class attribute. In this study we consider the following two options namely SDMI (Single DMI) and NDMI (Numerous DMI). In SDMI, we build a single DT (instead of a set of trees T) by considering the natural class attribute of D_F as the class attribute. Typically, every data set has a natural class attribute. For example, the natural class attribute of a patient data set can be "Diagnosis".

In NDMI we build a DT for each attribute of D_F as it is done in DMI. For numerical attribute we first generalizes the values of the attribute into N_C categories where N_C is the squared root of the domain size of the attribute. Therefore, for M attributes of D_F we have M decision trees in NDMI.

Following DMI, for both SDMI and NDMI we generate a sub data set for each logic rule of the DTs by assigning the records, of D_F, which satisfy the conditions of the logic rule. The numerical missing values of each sub data set are then imputed by using the EMI algorithm. Let, a numerical attribute A_j has a missing value for the record R_i, i.e. R_{ij} is missing. For SDMI, we identify the leaf where the record R_i falls in. EMI is then applied on all records representing the leaf and thus R_{ij} is imputed. For NDMI, we first find the leaf (of the tree T_j) where the record R_i falls in. EMI is then applied on all records belonging to the leaf for imputing R_{ij}. If the exact leaf of a record R_i cannot be determined due to the existence of the missing values then we use the union of all possible leaves.

For the categorical imputation by SDMI, a missing value R_{ij} is imputed by the mode value of the attribute A_j within the records of the leaf where the record R_i falls in. On the other hand, in NDMI the missing value R_{ij} is imputed by the majority class value of the leaf (of the tree T_j) where the record R_i falls in. Note that if a record R_i has multiple categorical missing values then in NDMI multiple trees are used, one tree for one attribute.

3.4 Proposed Framework

The proposed framework uses two steps of imputation, the Early-Imputation step and the Advanced-Imputation step. In this study we use one of the two existing high quality imputation techniques namely EMI (Junninen et al. 2004, Schneider 2001) and IBLLS (Cheng et al. 2012) in the Early-imputation step. For the Advanced-Imputation step we use either SDMI or NDMI.

Therefore, in this study we compare four versions of the framework namely EDI, ESI, LDI and LSI. EDI is the combination of EMI and NDMI, ESI is the combination of EMI and SDMI, LDI is the combination of IBLLS and NDMI, and LSI is the combination of IBLLS and SDMI. We now compare the performances of the techniques in the following section.

4 Experimental Results and Discussion

We implement our novel framework in four different versions namely EDI, ESI, LDI, and LSI, and three other high quality existing techniques namely DMI (Rahman & Islam 2011), EMI (Junninen et al. 2004, Schneider 2001) and IBLLS (Cheng et al. 2012). It was shown in the literature that the imputation accuracies of the existing techniques are better than many other techniques including Bayesian principal component analysis (BPCA) (Oba et al. 2003), LLSI (Kim et al. 2005), and ILLSI (Cai et al. 2006).

4.1 Data Set

We apply the techniques on two real data sets, namely the Yeast data set and the Autompg data set that are available from UCI Machine Learning Repository (Frank & Asuncion 2010). A brief description of the data sets is presented in Table 3.

Table 3: Data sets at a glance.

Data set	Records	Num. attr.	Cat. attr.	Missing	Pure Rec.
Yeast	1484	8	1	No	1484
Autompg	398	5	3	Yes	392

The Yeast data set has 1484 records, 8 numerical and 1 categorical attributes. There are no records having natural missing values in the data set. So, we use all 1484 records as a pure data set in our experiment. On the other hand, the Autompg data set has 398 records, 5 numerical and 3 categorical attributes. There are a number of records having missing values. We first remove all records having missing values. Therefore, we get a pure data set having 392 records without any missing values. In our experiments we use the pure data sets. Note that for the experimentation purpose we artificially create missing values in the pure data sets, the actual value of which is known to us.

4.2 Simulation of Missing Values

For experimentation, we artificially create missing values in the pure data set. We then impute the missing values by different techniques. Since we know the actual values of the artificially created missing value, we can evaluate the performances of the techniques by comparing the actual and the imputed values.

Generally the performances of an imputation technique depends on both the amount and the type/pattern of missing values (Junninen et al. 2004, Rubin 1976, Schneider 2001). Therefore, in this experiment we use various patterns of missing values such as simple, medium, complex and blended as discussed below.

A simple pattern permits a record to have at most one missing value, whereas a medium pattern permits a record to have minimum 2 attributes with missing values and maximum 50% of the attributes with missing values. Like wise, a complex pattern permits a record to have minimum 50% and maximum 80% attributes with missing values. A blended pattern allows a mixture of records from all three other patterns. We consider that a blended pattern simulates a natural scenario where we may expect a combination of all three missing patterns. In a blended pattern we have 25%, 50% and 25% records having missing values in the simple pattern, medium pattern and complex pattern, respectively (Junninen et al. 2004, Rahman & Islam 2011, Rahman et al. 2012).

For each missing pattern, we use four missing ratios: 1%, 3%, 5% and 10% where x% missing ratios means x% of the total attribute values (i.e. N records $\times$ M attributes) of a data set are missing. Note that for 10% missing ratios and simple pattern, the expected total number of records to have missing values may exceed the total records in some data sets. Therefore, in the simple missing pattern we consider 6% missing ratios (rather than 10% missing ratios) for all data sets.

In addition, two types of missing models namely Overall and Uniformly Distributed (UD) are considered. In the overall model, the attributes may not have equal number of missing values, and in the worst case scenario a single attribute can have all missing values. However, in the UD model the missing values are distributed equally in each attribute.

Based on the missing ratios, missing models, and missing patterns, we have a total of 32 missing combinations (id $1, 2, \cdots, 32$). For each combination, we generate 10 data sets with missing values. For example, for the combination having "1%" missing values , "overall" missing model , and "simple" missing pattern (id 1, see Table 4) we generate 10 data sets with missing values. Therefore, we generate all together 320 data sets for each natural data set namely Yeast and Autompg.

4.3 Evaluation Criteria

We evaluate the imputation accuracies (/performances) of the proposed and existing techniques in terms of four well known evaluation criteria namely co-efficient of determination (R^2), Index of agreement (d_2), root mean squared error ($RMSE$) and mean absolute error (MAE).

We now define the evaluation criteria briefly. Let L be the number of artificially created missing values, O_i ($1 \leq i \leq L$) be the actual value of the ith artificially created missing value, and P_i be the imputed value of the ith missing value. Also let $\overline{O}$ and $\overline{P}$ be the averages of the actual values $O_i; \forall i \in L$ and the imputed values $P_i; \forall i \in L$, respectively. Let σ_O and σ_P be the standard deviation of the actual values and the imputed values, respectively.

The coefficient of determination (R^2) (Junninen et al. 2004) describes the imputation accuracy based on the degree of correlation between actual and imputed values. The output of R^2 is a value between 0 and 1, where 1 indicates a perfect imputation.

$$R^2 = \left[\frac{1}{L} \frac{\sum_{i=1}^{L} \left[(P_i - \overline{P})(O_i - \overline{O}) \right]}{\sigma_P \sigma_O} \right]^2 \tag{7}$$

The index of agreement (d_2) (Willmott 1982) evaluates the degree of agreement between actual and imputed values. The output of d_2 is also a value between 0 and 1. Similar to R^2, a higher value of d_2 indicates a better fit. It is calculated as follows:

$$d = 1 - \left[\frac{\sum_{i=1}^{L} (P_i - O_i)^g}{\sum_{i=1}^{L} (|P_i - \overline{O}| + |O_i - \overline{O}|)^g} \right] \tag{8}$$

where the value of g can be either 1 or 2. We use a value 2 for the index g (i.e. d_2)throughout this experiment.

The root mean squared error ($RMSE$) (Junninen et al. 2004) measures the average difference between the actual and the imputed values. Its value ranges from 0 to ∞, where a lower value indicates a better imputation.

$$RMSE = \left(\frac{1}{L} \sum_{i=1}^{L} [P_i - O_i]^2 \right)^{\frac{1}{2}} \tag{9}$$

Finally, the mean absolute error (MAE) (Junninen et al. 2004) determines the similarity between the actual and imputed values. Similar to $RMSE$, its value ranges from 0 to ∞, where a lower value indicates a better matching.

$$MAE = \frac{1}{L} \sum_{i=1}^{L} |P_i - O_i| \tag{10}$$

4.4 Experimental Result Analysis on the Autompg and Yeast Data Sets

We present the performance of EDI, ESI, LDI, LSI, DMI, EMI, and IBLLS based on R^2, d_2, $RMSE$, and MAE for 32 missing combinations on the Autompg data set in Table 4. The table shows the average values of performance indicators on 10 data sets with missing values for each missing combination. For example, we have 10 data sets with missing values for the combination ($id = 1$) of "1%" missing ratio, "Overall" missing model and "Simple" missing pattern. The average of R^2 for the data sets having $id = 1$ is 0.908 for EDI as reported in Table 4. Bold values in the table indicate the best results among the seven techniques. Our proposed techniques (EDI, ESI, LDI and LSI) perform better than the existing techniques namely DMI, EMI and IBLLS in terms of all evaluation criteria. Moreover, the last row of the table, we present a score of each technique for each evaluation criteria, where a score "S" indicates that a technique performs the best among all the techniques in "S" (out of 32) number of missing combinations. The table shows that EDI outperforms all other techniques, the technique scores 31 (out of 32) for all evaluation criteria.

Similarly, Table 5 demonstrates the performance of the techniques in terms of all evaluation criteria for 32 missing combinations on the Yeast data set. The last row of the table indicates for all evaluation criteria EDI outperforms other techniques. EDI scores 32 (out of 32) for all evaluation criteria except MAE where EDI scores 29 and DMI scores 3.

4.5 Statistical Significance Analysis for All Data Sets

We present several statistical significance analysis on the Autompg and Yeast data sets. Since EDI (among the techniques we proposed in this paper) outperforms three other existing techniques, we present the statistical significance analysis of EDI, DMI, EMI and IBLLS as follows.

Figure 3 demonstrates 95% confidence interval analysis of EDI with DMI, EMI and IBLLS in terms of R^2 (Figure 3(a)), d_2 (Figure 3(b)), $RMSE$ (Figure 3(c)), and MAE (Figure 3(d)) for all 32 missing combinations on the Autompg data set. It is clear from the figures that EDI performs better (i.e. better average value and no overlap

Table 4: Performance of EDI, ESI, LDI, LSI, DMI, EMI, and IBLLS based on R^2, d_2, $RMSE$, and MAE for 32 missing combinations on Autompg data set

Missing combination		Id	R^2 (Higher value is better)							d_2 (Higher value is better)							$RMSE$ (Lower value is better)							MAE Lower value is better						
			EDI	ESI	LDI	LSI	DMI	EMI	IBLLS	EDI	ESI	LDI	LSI	DMI	EMI	IBLLS	EDI	ESI	LDI	LSI	DMI	EMI	IBLLS	EDI	ESI	LDI	LSI	DMI	EMI	IBLLS
1%	Overall	Simple 1	**0.908**	0.881	0.893	0.891	0.889	0.847	0.850	**0.972**	0.964	0.966	0.963	0.963	0.950	0.942	**0.071**	0.078	0.075	0.078	0.078	0.090	0.087	**0.053**	0.057	0.056	0.059	0.059	0.069	0.068
		Medium 2	**0.895**	0.878	0.873	0.864	0.891	0.841	0.828	**0.967**	0.963	0.960	0.958	0.966	0.953	0.935	**0.075**	0.079	0.081	0.084	0.076	0.088	0.092	**0.056**	0.059	0.062	0.062	0.067	0.069	0.074
		Complex 3	**0.814**	0.795	0.751	0.804	0.803	0.787	0.681	**0.928**	0.915	0.902	0.921	0.919	0.914	0.876	**0.095**	0.099	0.115	0.101	0.101	0.104	0.137	**0.070**	0.073	0.081	0.075	0.075	0.079	0.101
		Blended 4	**0.823**	0.808	0.793	0.809	0.814	0.796	0.676	**0.941**	0.936	0.934	0.937	0.940	0.932	0.875	**0.097**	0.102	0.104	0.100	0.099	0.103	0.138	**0.067**	0.069	0.069	0.070	0.069	0.074	0.096
	UD	Simple 5	**0.913**	0.901	0.912	0.909	0.908	0.872	0.855	**0.975**	0.970	0.974	0.973	0.973	0.960	0.921	**0.069**	0.073	0.069	0.071	0.071	0.084	0.096	**0.052**	0.055	0.052	0.055	0.054	0.063	0.076
		Medium 6	**0.894**	0.878	0.891	0.867	0.873	0.824	0.809	**0.968**	0.960	0.967	0.958	0.962	0.922	0.895	**0.081**	0.091	0.084	0.093	0.090	0.119	0.133	**0.061**	0.068	0.061	0.067	0.067	0.093	0.100
		Complex 7	**0.806**	0.770	0.806	0.751	0.774	0.756	0.651	**0.939**	0.925	0.938	0.917	0.926	0.918	0.847	**0.094**	0.105	0.094	0.108	0.104	0.109	0.146	**0.070**	0.078	0.071	0.082	0.080	0.087	0.112
		Blended 8	**0.892**	0.806	0.870	0.818	0.853	0.819	0.767	**0.962**	0.933	0.955	0.937	0.947	0.935	0.891	**0.073**	0.095	0.076	0.091	0.084	0.091	0.113	**0.054**	0.066	0.054	0.065	0.062	0.068	0.091
3%	Overall	Simple 9	**0.899**	0.864	0.897	0.887	0.861	0.850	0.839	**0.971**	0.961	0.971	0.968	0.961	0.956	0.939	**0.073**	0.084	0.073	0.076	0.084	0.087	0.095	**0.053**	0.060	0.053	0.056	0.061	0.067	0.070
		Medium 10	**0.865**	0.864	0.845	0.843	0.847	0.834	0.790	**0.961**	0.961	0.955	0.955	0.954	0.950	0.938	**0.083**	0.085	0.091	0.091	0.091	0.094	0.106	**0.060**	0.061	0.064	0.065	0.066	0.071	0.072
		Complex 11	**0.775**	0.769	0.717	0.743	0.750	0.731	0.598	**0.931**	0.929	0.913	0.919	0.919	0.906	0.866	**0.108**	0.109	0.122	0.116	0.115	0.127	0.150	**0.078**	0.081	0.082	0.085	0.087	0.099	0.103
		Blended 12	**0.846**	0.833	0.825	0.837	0.841	0.834	0.800	**0.959**	0.951	0.945	0.950	0.950	0.947	0.938	**0.088**	0.095	0.098	0.096	0.095	0.097	0.107	**0.063**	0.066	0.066	0.069	0.070	0.073	0.072
	UD	Simple 13	**0.883**	0.857	0.877	0.878	0.862	0.833	0.830	**0.965**	0.959	0.964	0.964	0.959	0.949	0.950	**0.082**	0.089	0.083	0.082	0.088	0.097	0.098	**0.058**	0.063	0.060	0.060	0.063	0.072	0.070
		Medium 14	**0.843**	0.825	0.842	0.831	0.832	0.800	0.796	**0.955**	0.950	0.954	0.950	0.949	0.922	0.930	**0.087**	0.092	0.087	0.091	0.092	0.107	0.105	**0.063**	0.066	0.064	0.066	0.068	0.084	0.078
		Complex 15	**0.764**	0.707	0.716	0.734	0.724	0.717	0.548	**0.925**	0.903	0.902	0.912	0.901	0.894	0.818	**0.116**	0.128	0.128	0.124	0.128	0.130	0.166	**0.082**	0.092	0.089	0.091	0.099	0.101	0.113
		Blended 16	**0.845**	0.833	0.837	0.824	0.827	0.804	0.789	**0.954**	0.952	0.953	0.949	0.950	0.941	0.918	**0.092**	0.093	0.092	0.095	0.094	0.101	0.107	**0.064**	0.066	0.065	0.068	0.069	0.075	0.077
5%	Overall	Simple 17	**0.889**	0.880	0.886	0.876	0.874	0.842	0.856	**0.968**	0.966	0.968	0.964	0.963	0.951	0.958	**0.079**	0.082	0.079	0.083	0.084	0.096	0.090	**0.056**	0.058	0.056	0.060	0.061	0.072	0.065
		Medium 18	**0.837**	0.813	0.799	0.815	0.805	0.789	0.743	**0.950**	0.946	0.943	0.946	0.942	0.936	0.924	**0.089**	0.094	0.099	0.095	0.097	0.101	0.111	**0.065**	0.065	0.067	0.068	0.070	0.075	0.075
		Complex 19	**0.772**	0.720	0.731	0.724	0.688	0.688	0.643	**0.927**	0.916	0.919	0.916	0.893	0.892	0.884	**0.109**	0.121	0.118	0.119	0.129	0.130	0.141	**0.080**	0.086	0.082	0.087	0.099	0.100	0.103
		Blended 20	**0.843**	0.835	0.821	0.843	0.841	0.819	0.763	**0.955**	0.952	0.948	0.954	0.953	0.945	0.931	**0.086**	0.090	0.096	0.089	0.089	0.096	0.109	**0.060**	0.064	0.065	0.064	0.065	0.072	0.074
	UD	Simple 21	**0.883**	0.871	0.876	0.874	0.860	0.852	0.842	**0.968**	0.964	0.966	0.965	0.960	0.955	0.955	**0.078**	0.082	0.080	0.080	0.085	0.089	0.091	**0.056**	0.058	0.057	0.059	0.061	0.067	0.065
		Medium 22	**0.850**	0.843	0.848	0.838	0.829	0.804	0.798	**0.958**	0.955	0.957	0.953	0.949	0.939	0.921	**0.088**	0.089	0.089	0.092	0.095	0.101	0.107	**0.062**	0.064	0.063	0.066	0.071	0.077	0.080
		Complex 23	**0.771**	0.749	0.737	0.737	0.721	0.713	0.608	**0.932**	0.925	0.921	0.919	0.903	0.899	0.870	**0.105**	0.110	0.113	0.112	0.120	0.122	0.143	**0.077**	0.078	0.080	0.083	0.094	0.097	0.101
		Blended 24	**0.824**	0.813	0.808	0.813	0.803	0.791	0.726	**0.949**	0.945	0.944	0.944	0.940	0.935	0.912	**0.094**	0.098	0.100	0.098	0.102	0.105	0.123	**0.065**	0.069	0.068	0.071	0.075	0.079	0.085
10%	Overall	Simple 25	**0.873**	0.855	0.870	0.859	0.860	0.827	0.815	**0.965**	0.959	0.964	0.960	0.961	0.948	0.947	**0.081**	0.087	0.081	0.085	0.085	0.096	0.098	**0.056**	0.060	0.057	0.062	0.060	0.072	0.069
		Medium 26	**0.835**	0.831	0.825	0.833	0.824	0.806	0.770	**0.953**	0.952	0.951	0.952	0.948	0.942	0.933	**0.094**	0.095	0.096	0.094	0.097	0.102	0.111	**0.067**	0.068	0.067	0.069	0.072	0.077	0.076
		Complex 27	0.750	**0.762**	0.743	0.743	0.730	0.712	0.620	0.925	**0.931**	0.924	0.921	0.906	0.899	0.861	0.114	**0.112**	0.116	0.117	0.123	0.128	0.154	0.081	**0.079**	0.080	0.085	0.093	0.098	0.116
		Blended 28	**0.830**	0.819	0.806	0.799	0.790	0.776	0.708	**0.952**	0.948	0.945	0.939	0.929	0.922	0.913	**0.094**	0.097	0.099	0.102	0.108	0.113	0.123	**0.065**	0.067	0.069	0.073	0.081	0.087	0.084
	UD	Simple 29	**0.884**	0.864	0.878	0.860	0.865	0.843	0.831	**0.968**	0.962	0.966	0.960	0.961	0.954	0.952	**0.078**	0.086	0.081	0.087	0.086	0.093	0.095	**0.055**	0.061	0.059	0.064	0.063	0.070	0.068
		Medium 30	**0.832**	0.829	0.809	0.826	0.818	0.803	0.726	**0.952**	0.952	0.946	0.950	0.946	0.939	0.917	**0.093**	0.095	0.101	0.096	0.099	0.104	0.121	**0.067**	0.068	0.070	0.069	0.074	0.079	0.084
		Complex 31	**0.757**	0.756	0.689	0.735	0.720	0.713	0.538	**0.928**	0.927	0.903	0.916	0.903	0.899	0.843	**0.111**	0.112	0.126	0.117	0.123	0.125	0.161	**0.080**	0.082	0.086	0.087	0.096	0.099	0.112
		Blended 32	**0.827**	0.806	0.824	0.800	0.806	0.790	0.705	**0.951**	0.944	0.949	0.941	0.940	0.933	0.898	**0.094**	0.100	0.095	0.101	0.101	0.106	0.133	**0.067**	0.070	0.068	0.072	0.075	0.080	0.098
Score (Out of 32)			**31**	1	0	0	0	0	0	**31**	1	0	0	0	0	0	**31**	1	0	0	0	0	0	**31**	1	0	0	0	0	0

Table 5: Performance of EDI, ESI, LDI, LSI, DMI, EMI, and IBLLS based on R^2, d_2, $RMSE$, and MAE for 32 missing combinations on Yeast data set

Missing combination		Id	R^2 (Higher value is better)							d_2 (Higher value is better)							$RMSE$ (Lower value is better)							MAE Lower value is better						
			EDI	ESI	LDI	LSI	DMI	EMI	IBLLS	EDI	ESI	LDI	LSI	DMI	EMI	IBLLS	EDI	ESI	LDI	LSI	DMI	EMI	IBLLS	EDI	ESI	LDI	LSI	DMI	EMI	IBLLS
1%	Overall	Simple 1	**0.860**	0.523	0.743	0.661	0.833	0.770	0.803	**0.959**	0.848	0.911	0.868	0.953	0.924	0.925	**0.090**	0.179	0.121	0.144	0.098	0.108	0.103	**0.062**	0.094	0.080	0.097	0.063	0.072	0.075
		Medium 2	**0.805**	0.514	0.607	0.551	0.757	0.742	0.731	**0.944**	0.848	0.841	0.800	0.930	0.907	0.914	**0.105**	0.179	0.161	0.184	0.119	0.127	0.125	**0.067**	0.087	0.106	0.127	0.070	0.077	0.083
		Complex 3	**0.776**	0.636	0.756	0.669	0.746	0.744	0.732	**0.932**	0.890	0.926	0.878	0.925	0.904	0.918	**0.116**	0.151	0.121	0.149	0.123	0.126	0.127	**0.071**	0.078	0.072	0.092	0.072	0.077	0.081
		Blended 4	**0.820**	0.588	0.725	0.667	0.809	0.765	0.795	**0.948**	0.874	0.902	0.880	0.945	0.919	0.937	**0.100**	0.159	0.125	0.140	0.103	0.111	0.108	**0.065**	0.083	0.081	0.087	0.066	0.073	0.077
	UD	Simple 5	**0.838**	0.494	0.689	0.611	0.787	0.759	0.761	**0.952**	0.841	0.888	0.853	0.938	0.912	0.924	**0.100**	0.185	0.140	0.163	0.111	0.118	0.115	**0.067**	0.094	0.089	0.105	0.067	0.075	0.079
		Medium 6	**0.782**	0.473	0.656	0.588	0.746	0.741	0.729	**0.936**	0.826	0.886	0.851	0.922	0.901	0.914	**0.109**	0.189	0.147	0.169	0.123	0.129	0.126	**0.068**	0.099	0.092	0.109	0.072	0.076	0.083
		Complex 7	**0.788**	0.598	0.762	0.685	0.754	0.748	0.711	**0.938**	0.870	0.930	0.871	0.928	0.907	0.913	**0.110**	0.159	0.117	0.149	0.120	0.126	0.130	**0.070**	0.084	0.072	0.098	0.071	0.078	0.082
		Blended 8	**0.791**	0.566	0.622	0.579	0.744	0.736	0.713	**0.939**	0.859	0.864	0.832	0.924	0.902	0.909	**0.111**	0.170	0.160	0.177	0.126	0.133	0.133	**0.068**	0.090	0.102	0.118	0.072	0.078	0.086
3%	Overall	Simple 9	**0.800**	0.614	0.657	0.655	0.793	0.758	0.766	**0.941**	0.872	0.878	0.876	0.940	0.914	0.924	**0.108**	0.157	0.146	0.148	0.111	0.115	0.119	**0.068**	0.091	0.096	0.102	0.069	0.074	0.086
		Medium 10	**0.800**	0.514	0.664	0.586	0.794	0.754	0.771	**0.941**	0.848	0.880	0.836	0.941	0.913	0.926	**0.109**	0.181	0.149	0.174	0.111	0.117	0.118	0.068	0.094	0.093	0.113	**0.067**	0.074	0.085
		Complex 11	**0.764**	0.506	0.746	0.663	0.760	0.740	0.701	**0.931**	0.825	0.920	0.878	0.930	0.904	0.903	**0.118**	0.185	0.123	0.147	0.119	0.125	0.133	**0.069**	0.109	0.079	0.096	0.069	0.076	0.093
		Blended 12	**0.802**	0.534	0.592	0.632	0.797	0.757	0.737	**0.942**	0.846	0.835	0.851	0.942	0.912	0.906	**0.108**	0.175	0.162	0.153	0.109	0.116	0.130	**0.068**	0.101	0.108	0.103	0.068	0.075	0.094
	UD	Simple 13	**0.790**	0.431	0.697	0.633	0.782	0.752	0.767	**0.939**	0.794	0.895	0.835	0.938	0.911	0.925	**0.110**	0.208	0.135	0.170	0.112	0.117	0.117	**0.064**	0.118	0.089	0.119	0.067	0.073	0.083
		Medium 14	**0.814**	0.424	0.563	0.483	0.810	0.762	0.777	**0.945**	0.800	0.818	0.762	0.945	0.915	0.933	**0.104**	0.201	0.171	0.203	0.105	0.110	0.111	**0.064**	0.112	0.114	0.141	0.067	0.073	0.082
		Complex 15	**0.805**	0.516	0.708	0.707	0.801	0.755	0.743	**0.943**	0.832	0.893	0.882	0.943	0.914	0.918	**0.107**	0.185	0.133	0.141	0.107	0.113	0.124	**0.065**	0.106	0.088	0.096	0.068	0.074	0.089
		Blended 16	**0.821**	0.555	0.384	0.308	0.785	0.752	0.754	**0.947**	0.861	0.732	0.686	0.938	0.912	0.923	**0.103**	0.168	0.223	0.250	0.113	0.118	0.121	**0.067**	0.093	0.147	0.168	0.068	0.075	0.086
5%	Overall	Simple 17	**0.793**	0.549	0.670	0.594	0.773	0.753	0.658	**0.939**	0.855	0.889	0.844	0.933	0.911	0.879	**0.111**	0.172	0.140	0.166	0.117	0.118	0.144	**0.071**	0.105	0.093	0.114	0.072	0.074	0.102
		Medium 18	**0.797**	0.491	0.243	0.249	0.786	0.757	0.765	**0.941**	0.806	0.635	0.644	0.938	0.913	0.921	**0.109**	0.207	0.259	0.260	0.112	0.117	0.120	**0.067**	0.124	0.176	0.178	0.069	0.073	0.087
		Complex 19	**0.787**	0.550	0.699	0.626	0.777	0.749	0.693	**0.938**	0.850	0.904	0.856	0.935	0.908	0.895	**0.111**	0.169	0.134	0.162	0.114	0.120	0.138	**0.070**	0.101	0.089	0.114	0.071	0.078	0.101
		Blended 20	**0.786**	0.507	0.335	0.256	0.779	0.748	0.698	**0.937**	0.824	0.714	0.666	0.937	0.909	0.900	**0.110**	0.181	0.228	0.254	0.112	0.117	0.133	**0.068**	0.108	0.146	0.169	0.068	0.074	0.091
	UD	Simple 21	**0.788**	0.467	0.469	0.402	0.778	0.754	0.707	**0.937**	0.810	0.781	0.726	0.935	0.912	0.900	**0.112**	0.195	0.198	0.229	0.115	0.117	0.134	**0.070**	0.118	0.131	0.159	0.071	0.074	0.096
		Medium 22	**0.791**	0.380	0.547	0.491	0.785	0.752	0.711	**0.939**	0.765	0.825	0.783	0.939	0.911	0.902	**0.110**	0.216	0.171	0.193	0.111	0.117	0.131	0.069	0.124	0.111	0.131	**0.068**	0.075	0.092
		Complex 23	**0.781**	0.565	0.721	0.644	0.775	0.745	0.703	**0.936**	0.857	0.911	0.857	0.934	0.907	0.901	**0.113**	0.165	0.129	0.159	0.114	0.120	0.132	**0.070**	0.094	0.086	0.109	0.071	0.076	0.094
		Blended 24	**0.789**	0.614	0.522	0.484	0.776	0.751	0.677	**0.935**	0.874	0.813	0.778	0.933	0.911	0.884	**0.110**	0.155	0.177	0.194	0.114	0.117	0.140	**0.070**	0.097	0.119	0.136	0.071	0.074	0.101
10%	Overall	Simple 25	**0.807**	0.650	0.477	0.448	0.795	0.761	0.733	**0.944**	0.888	0.767	0.748	0.942	0.913	0.909	**0.106**	0.149	0.194	0.209	0.110	0.113	0.129	**0.067**	0.092	0.128	0.139	0.068	0.073	0.096
		Medium 26	**0.814**	0.598	0.154	0.154	0.800	0.765	0.731	**0.945**	0.863	0.596	0.585	0.942	0.916	0.901	**0.104**	0.160	0.285	0.290	0.108	0.112	0.130	0.071	0.101	0.193	0.201	**0.068**	0.072	0.097
		Complex 27	**0.771**	0.388	0.687	0.611	0.763	0.738	0.612	**0.933**	0.769	0.890	0.839	0.931	0.903	0.851	**0.115**	0.212	0.136	0.165	0.118	0.123	0.155	**0.071**	0.131	0.089	0.111	0.072	0.077	0.113
		Blended 28	**0.786**	0.637	0.297	0.227	0.765	0.746	0.688	**0.936**	0.883	0.680	0.635	0.932	0.907	0.887	**0.113**	0.149	0.252	0.279	0.119	0.121	0.141	**0.071**	0.091	0.173	0.195	0.071	0.076	0.103
	UD	Simple 29	**0.800**	0.580	0.562	0.490	0.768	0.752	0.718	**0.941**	0.856	0.825	0.766	0.934	0.910	0.904	**0.108**	0.162	0.172	0.202	0.110	0.117	0.131	**0.069**	0.102	0.115	0.141	0.070	0.073	0.095
		Medium 30	**0.797**	0.506	0.353	0.283	0.789	0.758	0.736	**0.942**	0.821	0.727	0.686	0.940	0.911	0.904	**0.109**	0.181	0.228	0.250	0.111	0.116	0.129	**0.068**	0.111	0.157	0.175	0.069	0.075	0.097
		Complex 31	**0.783**	0.536	0.606	0.525	0.778	0.748	0.637	**0.938**	0.843	0.863	0.808	0.936	0.907	0.863	**0.112**	0.174	0.160	0.190	0.114	0.120	0.150	**0.070**	0.107	0.100	0.123	0.070	0.076	0.109
		Blended 32	**0.795**	0.566	0.447	0.416	0.779	0.745	0.589	**0.937**	0.855	0.770	0.724	0.936	0.908	0.847	**0.111**	0.165	0.207	0.230	0.115	0.120	0.158	**0.068**	0.101	0.135	0.155	0.069	0.076	0.112
Score (Out of 32)			**32**	0	0	0	0	0	0	**32**	0	0	0	0	0	0	**32**	0	0	0	0	0	0	**29**	0	0	0	3	0	0

of confidence intervals) than other techniques for most of the missing combinations except for the 13 cases (out of 32×4=128 cases) in Figure 3 marked by the circles. Even for those 13 cases the average values of of EDI are better than the average values of the other techniques in terms of R^2, d_2, $RMSE$, and MAE.

The 95% confidence interval analysis of EDI with existing techniques on the Yeast data set is presented in Figure 4. The figure also indicates that EDI performs better than the existing techniques for all missing combinations

except for those marked by the circles.

We can see from the figures that IBLLS in general performs worse for a high missing ratios, whereas EDI maintains almost the same performance even for a high missing ratios.

In Figure 5 we present a statistical significance analysis using t-test for all 32 missing combinations of all data sets. The figure demonstrates a considerably better performance of EDI over other techniques at $p = 0.05$ based on all evaluation criteria for the Autompg and Yeast data sets. The t-values are higher than the t(ref) values. We get the values of t (ref) using Student's t distribution table (*Distribution table: Students t [online available: http://www.statsoft.com/textbook/distribution-tables/]* 2013).

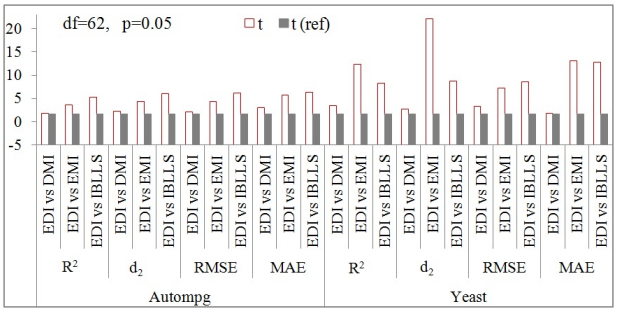


Figure 5: t-test analysis on Autompg, and Yeast data sets

4.6 Aggregated Performance Analysis for All Data Sets

We now present aggregated performances of all techniques in terms of missing ratios, missing models, and missing patterns in Figure 6 for the Autompg data set. The figures demonstrate that EDI performs better (i.e. higher average imputation accuracy) than other techniques for all missing ratios, for all missing models, and for all missing patterns in terms of R^2 (Figure 6(a)), d_2 (Figure 6(b)), $RMSE$ (Figure 6(c)), and MAE (Figure 6(d)).

Similarly for the Yeast data set we present the aggregated imputation accuracies in Figure 7. The figures demonstrate that EDI performs better (i.e. higher average imputation accuracy) than other techniques for most of the missing ratios, for all missing models, and for all missing patterns in terms of all evaluation criteria except MAE (Figure 7(d)) where DMI also achieves the same accuracy for 5% and 10% missing ratios, and simple and medium missing patterns.

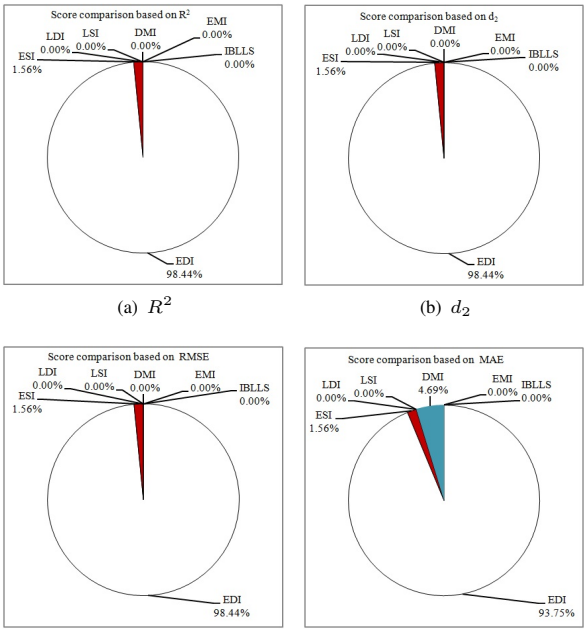
We also present overall performances (i.e. the average value of the accuracies for the 320 data sets) based on R^2, d_2, $RMSE$, and MAE for the Autompg and the Yeast data set in Table 6. For the data sets the overall imputation accuracy of EDI is higher than the overall imputation accuracy of other techniques. For the Autompg data set the overall imputation accuracies of EDI, ESI, LDI and LSI, in terms of R^2, d_2, $RMSE$, and MAE, are higher than of DMI,EMI and IBLLS. Similarly we get better imputation accuracy for EDI on Yeast data in terms of R^2, d_2, $RMSE$, and MAE.

Figure 8 presents the percentage of the combinations (out of the total 64 combinations for the two data sets) where the techniques perform the best. For example, EDI performs the best in 98.44% combinations in terms of R^2 (Figure 8(a)), d_2 (Figure 8(b)) and $RMSE$ (Figure 8(c)), and in 93.75% combinations in terms of MAE (Figure 8(d)).

Based on the experimental results of this study it appears that the imputation accuracy improves significantly by the use of the Early-imputation step in DMI. Out of the four versions of the framework, EDI outperforms the others in terms of all evaluation criteria for all data sets. The initial experimental results indicate that the use of EMI in the Early-imputation step and NDMI (i.e. DMI with the M number of decision trees) in the Advanced-imputation step improves the imputation accuracy significantly. EDI achieves higher imputation accuracies than the accuracies of DMI and the two other existing techniques.

Table 6: Overall average performance on Autompg and Yeast data sets

Data set	Evaluation Criteria	EDI	ESI	LDI	LSI	DMI	EMI	IBLLS
Autompg	R^2	**0.841**	0.822	0.822	0.821	0.818	0.797	0.744
	d_2	**0.952**	0.946	0.946	0.945	0.942	0.932	0.909
	$RMSE$	**0.090**	0.095	0.095	0.096	0.097	0.104	0.118
	MAE	**0.065**	0.068	0.067	0.070	0.073	0.080	0.085
Yeast	R^2	**0.797**	0.533	0.574	0.518	0.780	0.752	0.720
	d_2	**0.940**	0.842	0.831	0.793	0.936	0.910	0.905
	$RMSE$	**0.109**	0.176	0.169	0.190	0.113	0.119	0.129
	MAE	**0.068**	0.101	0.111	0.129	0.069	0.075	0.091



(a) R^2 (b) d_2 (c) $RMSE$ (d) MAE

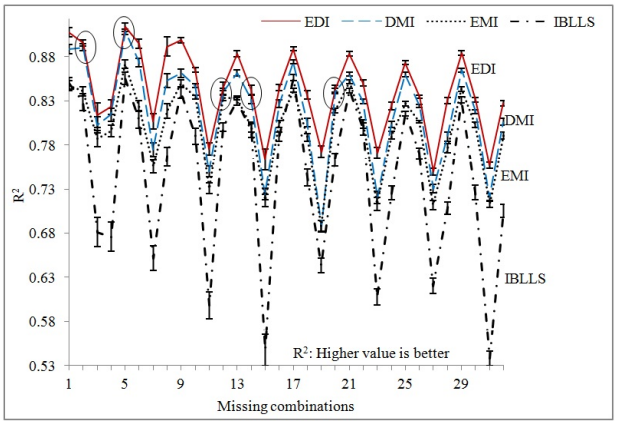
Figure 8: Percentage of combinations for all data sets, where a technique achieves the best result.

4.7 Experimentation on the Imputation of the Categorical Missing Values

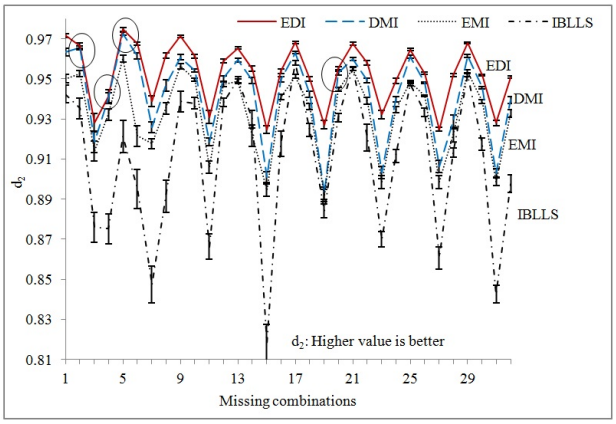
Unlike EMI and IBLLS, the proposed techniques (i.e. EDI,ESI, LDI, LSI) can impute categorical missing values in addition to numerical missing values. Therefore, we now compare the performances of the techniques with only DMI, for the imputation of categorical values. Fig. 9 shows that EDI achieves lower $RMSE$ and MAE values than DMI for all data sets. For each data set $RMSE$ and MAE values are computed using all 32 combinations. Note that for $RMSE$ and MAE a lower value indicates a better imputation.

4.8 Execution Time Complexity Analysis

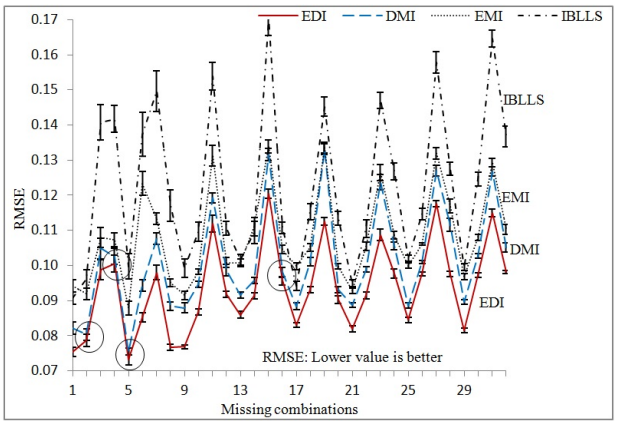
We now present the average execution time (in milliseconds) for 320 data sets (32 combinations $\times$ 10 data sets per combination) with missing values for each real data set in Table 7. We carry out the experiments on a machine



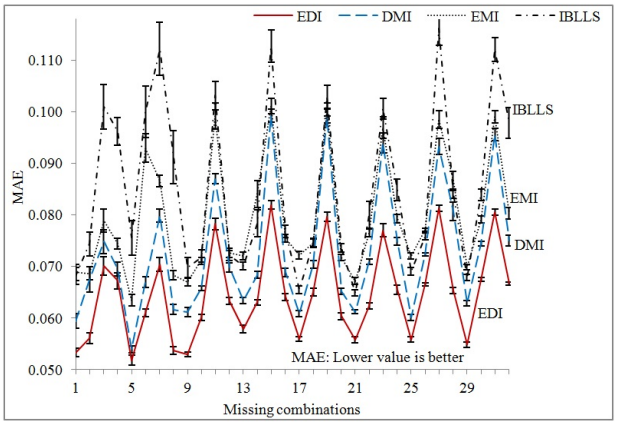
(a) R^2



(b) d_2

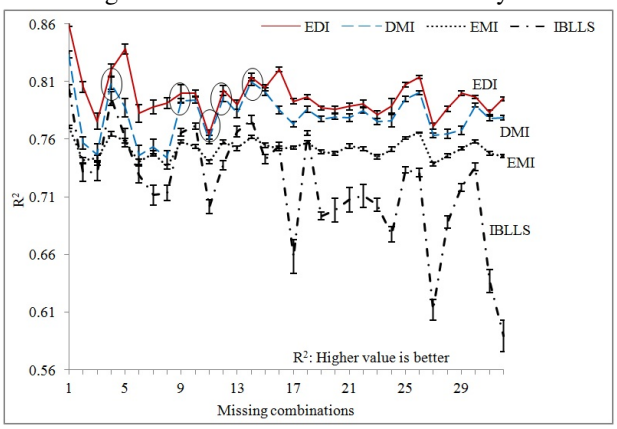


(c) $RMSE$

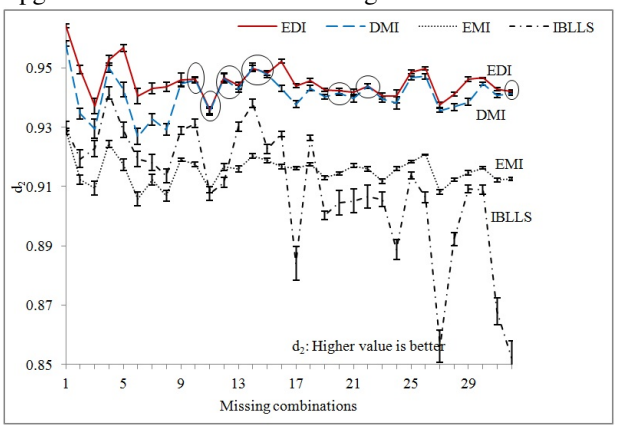


(d) MAE

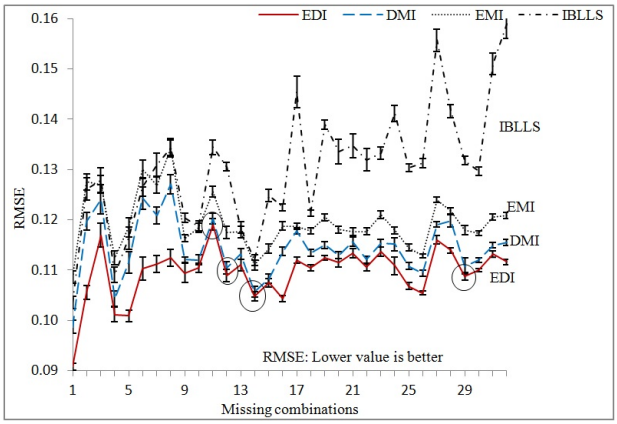
Figure 3: 95% confidence interval analysis on Autompg data set in terms of 32 missing combinations.



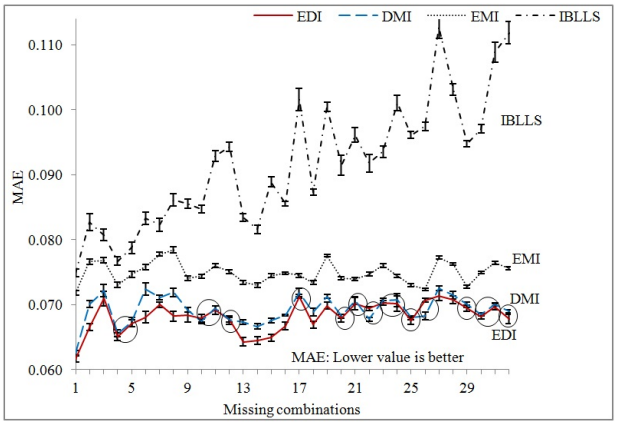
(a) R^2



(b) d_2

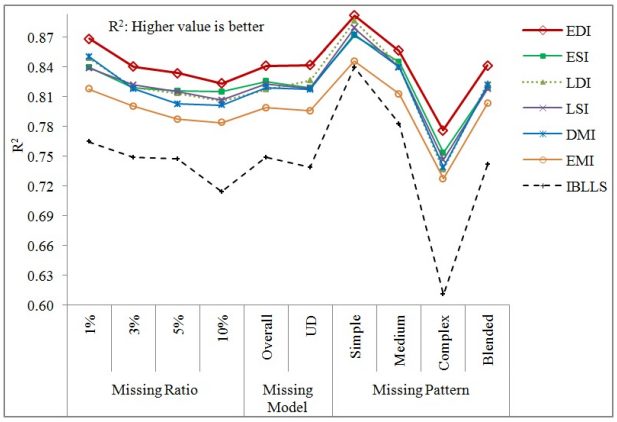


(c) $RMSE$

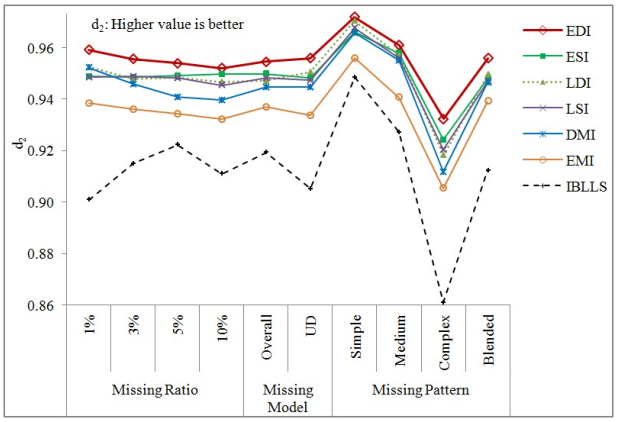


(d) MAE

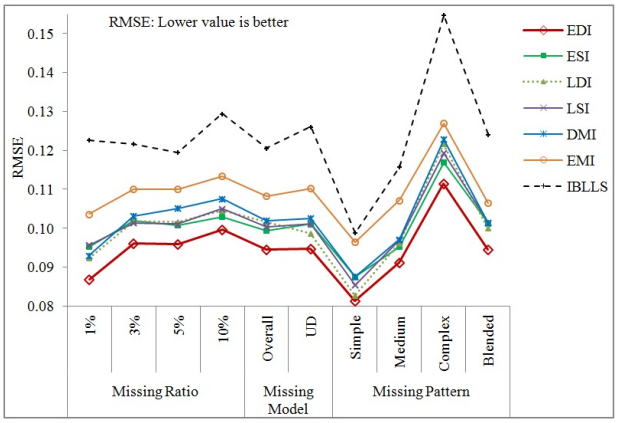
Figure 4: 95% confidence interval analysis on Yeast data set in terms of 32 missing combinations.



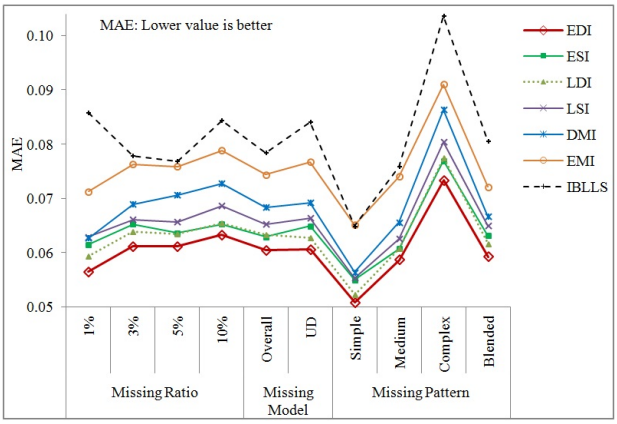
(a) R^2



(b) d_2

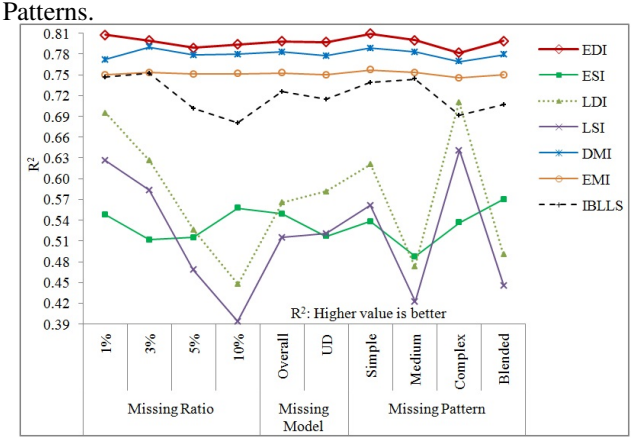


(c) $RMSE$

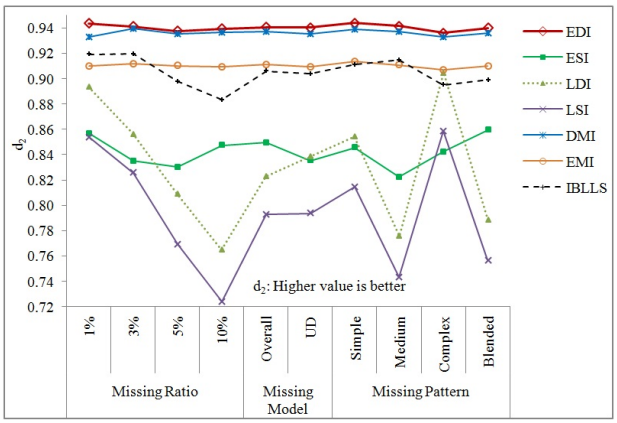


(d) MAE

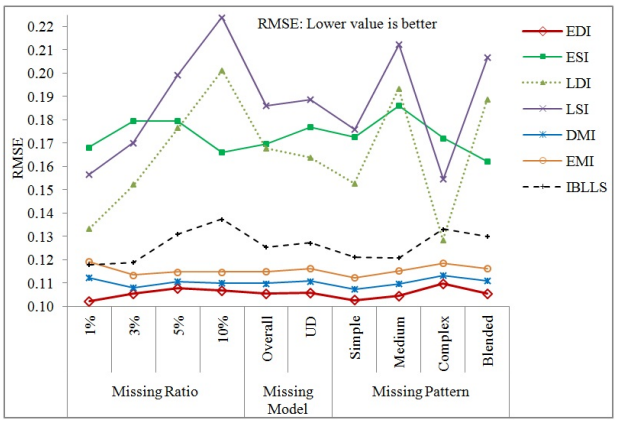
Figure 6: Aggregated performance on Autompg data set in terms of Missing Ratios, Missing Models, and Missing Patterns.



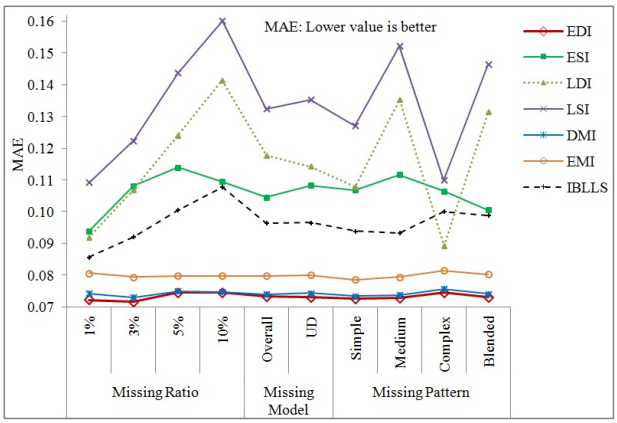
(a) R^2



(b) d_2



(c) $RMSE$



(d) MAE

Figure 7: Aggregated performance on Yeast data set in terms of Missing Ratios, Missing Models, and Missing Patterns.

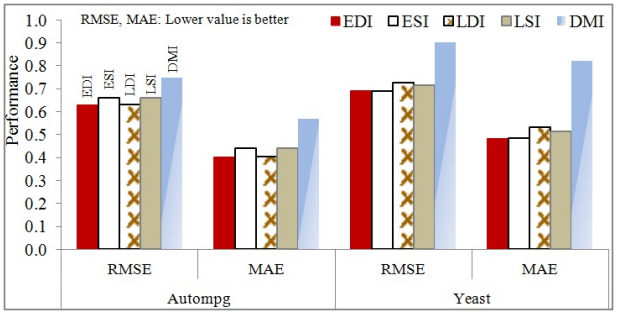


Figure 9: Performance comparison for categorical imputation on two data sets

having configuration 4 × 8 core Intel E7-8837 Xeon processors, 256 GB RAM. EDI takes less time than IBLLS, whereas it takes slightly more time than EMI to pay the cost of a significantly better quality imputation.

Table 7: Average execution time (in milliseconds) of different techniques on the two data sets.

Data set	EDI	ESI	LDI	LSI	DMI	EMI	IBLLS
Autompg	4,191	782	12,264	8,959	2,215	18	8,861
Yeast	6,913	4,164	182,670	193,147	3,024	92	173,209

5 Conclusion

In this paper we present a novel imputation framework that uses two layers of imputation, an Early-imputation and an Advanced-imputation step. We argue that an early imputation before the actual one should improve the imputation accuracy significantly. Especially for an existing technique called DMI the two layered approach of imputation should improve the accuracy significantly. We point out that if a big number of records have missing values then DMI may suffer from low accuracy. In this study we experimented four versions of the proposed framework on two data sets and four evaluation criteria. The experimental results show that the version called EDI (which is the combination of EMI and NDMI) gives the best results. The superiority of EDI over ESI, and LDI over LSI supports our belief on the supremacy of NDMI over SDMI. EDI performs better than ESI in 63 out of 64 combinations, and LDI outperforms LSI in 50 out of 64 combinations for the two data sets (see Table 4 and Table 5). Additionally, the superiority of EDI and the other three versions of the proposed framework over the three existing techniques justifies our argument in favour of the two layered approach. EDI outperforms all three existing techniques in 63 out of 64 total combinations for the two data sets. Our future research plans include the further development of the framework in order to reduce its time complexity.

References

Aydilek, I. B. & Arslan, A. (2013), 'A hybrid method for imputation of missing values using optimized fuzzy c-means with support vector regression and a genetic algorithm', *Information Sciences* **233**, 25 – 35.

Batista, G. & Monard, M. (2003), 'An analysis of four missing data treatment methods for supervised learning', *Applied Artificial Intelligence* **17**(5-6), 519–533.

Bø, T. H., Dysvik, B. & Jonassen, I. (2004), 'Lsimpute: accurate estimation of missing values in microarray data with least squares methods', *Nucleic acids research* **32**(3), e34–e34.

Cai, Z., Heydari, M. & Lin, G. (2006), 'Iterated local least squares microarray missing value imputation', *Journal of Bioinformatics and Computational Biology* **4**(5), 935–958.

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.

Derjani Bayeh, A. & Smith, M. J. (1999), 'Effect of physical ergonomics on vdt workers' health: a longitudinal intervention field study in a service organization', *International Journal of Human-Computer Interaction* **11**(2), 109–135.

Distribution table: Students t [online available: http://www.statsoft.com/textbook/distribution-tables/] (2013). Accessed July 7, 2013.
URL: *http://www.statsoft.com/textbook/distribution-tables/*

Dorri, F., Azmi, P. & Dorri, F. (2012), 'Missing value imputation in dna microarrays based on conjugate gradient method', *Computers in Biology and Medicine* **42**, 222–227.

Farhangfar, A., Kurgan, L. A. & Pedrycz, W. (2007), 'A novel framework for imputation of missing values in databases', *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* **37**(5), 692–709.

Farhangfar, A., Kurgan, L. & Dy, J. (2008), 'Impact of imputation of missing values on classification error for discrete data', *Pattern Recognition* **41**(12), 3692–3705.

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*

Han, J. & Kamber, M. (2000), 'Data mining: Concepts and techniques', *The Morgan Kaufmann Series in data management systems* **2**.

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.

Khoshgoftaar, T. & Van Hulse, J. (2005), Empirical case studies in attribute noise detection, *in* 'Information Reuse and Integration, Conf, 2005. IRI-2005 IEEE International Conference on.', IEEE, pp. 211–216.

Kim, H., Golub, G. & Park, H. (2005), 'Missing value estimation for dna microarray gene expression data: local least squares imputation', *Bioinformatics* **21**(2), 187–198.

Li, D., Deogun, J., Spaulding, W. & Shuart, B. (2004), Towards missing data imputation: A study of fuzzy k-means clustering method, *in* 'Rough Sets and Current Trends in Computing', Springer, pp. 573–579.

Liew, A. W.-C., Law, N.-F. & Yan, H. (2011), 'Missing value imputation for gene expression data: computational techniques to recover missing data from available information', *Briefings in bioinformatics* **12**(5), 498–513.

Liu, C., Dai, D. & Yan, H. (2010), 'The theoretic framework of local weighted approximation for microarray missing value estimation', *Pattern Recognition* **43**(8), 2993–3002.

Maletic, J. & Marcus, A. (2000), Data cleansing: Beyond integrity analysis, *in* 'Proceedings of the Conference on Information Quality', Citeseer, pp. 200–209.

Muralidhar, K., Parsa, R. & Sarathy, R. (1999), 'A general additive data perturbation method for database security', *Management Science* pp. 1399–1415.

Oba, S., Sato, M., Takemasa, I., Monden, M., Matsubara, K. & Ishii, S. (2003), 'A bayesian missing value estimation method for gene expression profile data', *Bioinformatics* **19**(16), 2088–2096.

Olivetti de França, F., Palermo Coelho, G. & Von Zuben, F. J. (2013), 'Predicting missing values with biclustering: A coherence-based approach', *Pattern Recognition* **46**(5), 1255–1266.

Osborne, J. & Overbay, A. (2008), 'Best practices in data cleaning', *Best Practices in Quantitative Methods* pp. 205–213.

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

Quinlan, J. R. (1996), 'Improved use of continuous attributes in C4.5', *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*), Data quality improvement by imputation of missing values, *in* 'International Conference on Computer Science and Information Technology (CSIT-2013)', Yogyakarta, Indonesia.

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* .
URL: *http://dx.doi.org/10.1016/j.knosys.2013.08.023*

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, Brisbane, Australia, pp. 1 –10.

Rubin, D. (1976), 'Inference and missing data', *Biometrika* **63**(3), 581–592.

Schafer, J. L. & Graham, J. W. (2002), 'Missing data: our view of the state of the art.', *Psychological methods* **7**(2), 147.

Schneider, T. (2001), 'Analysis of incomplete climate data: Estimation of mean values and covariance matrices and imputation of missing values', *Journal of Climate* **14**(5), 853–871.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D. & Altman, R. B. (2001), 'Missing value estimation methods for dna microarrays', *Bioinformatics* **17**(6), 520–525.

Twala, B. & Phorah, M. (2010), 'Predicting incomplete gene microarray data with the use of supervised learning algorithms', *Pattern Recognition Letters* **31**(13), 2061–2069.

Willmott, C. (1982), 'Some comments on the evaluation of model performance.', *Bulletin of the American Meteorological Society* **63**, 1309–1369.

Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Philip, S. Y. et al. (2008), 'Top 10 algorithms in data mining', *Knowledge and Information Systems* **14**(1), 1–37.

Young, W., Weckman, G. & Holland, W. (2011), 'A survey of methodologies for the treatment of missing values within datasets: limitations and benefits', *Theoretical Issues in Ergonomics Science* **12**(1), 15–43.

Zhang, S., Jin, Z. & Zhu, X. (2011), 'Missing data imputation by utilizing information within incomplete instances', *Journal of Systems and Software* **84**(3), 452–459.

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.

Zhu, X., Zhang, S., Jin, Z., Zhang, Z. & Xu, Z. (2011), 'Missing value estimation for mixed-attribute data sets', *Knowledge and Data Engineering, IEEE Transactions on* **23**(1), 110–121.

Towards a Feature Rich Model for Predicting Spam Emails containing Malicious Attachments and URLs

Khoi-Nguyen Tran*, Mamoun Alazab#, Roderic Broadhurst#
*Research School of Computer Science, #Regulatory Institutions Network
The Australian National University, ACT 0200, Australia
{khoi-nguyen.tran, mamoun.alazab, roderic.broadhurst}@.anu.edu.au

Abstract

Malicious content in spam emails is increasing in the form of attachments and URLs. Malicious attachments and URLs attempt to deliver software that can compromise the security of a computer. These malicious attachments also try to disguise their content to avoid virus scanners used by most email services to screen for such risks. Malicious URLs add another layer of disguise, where the email content tries to entice the recipient to click on a URL that links to a malicious Web site or downloads a malicious attachment. In this paper, based on two real world data sets we present our preliminary research on predicting the kind of spam email most likely to contain these highly dangerous spam emails. We propose a rich set of features for the content of emails to capture regularities in emails containing malicious content. We show these features can predict malicious attachments within an area under the precious recall curve (AUC-PR) up to 95.2%, and up to 68.1% for URLs. Our work can help reduce reliance on virus scanners and URL blacklists, which often do not update as quickly as the malicious content it attempts to identify. Such methods could reduce the many different resources now needed to identify malicious content.

Keywords: Email, Spam, Malicious, Attachment, URL, Machine Learning.

1 Introduction

Email spam, unsolicited bulk email (Blanzieri & Bryl, 2008), accounted for an average of 66.5% of all emails sent in the first quarter of 2013, and of these 3.3% contained malicious attachments[1]. Estimates show that approximately 183 billion emails (i.e. 6 billion emails with malicious attachments) were sent every day in the first quarter of 2013[2]. Malicious attachments and embedded URLs (Universal Resource Locators – also known as Web links) are attempts to infect the computer of a recipient with malware (malicious software) such as viruses, trojans, and keyloggers. Malicious attachments in an email are attempts at direct delivery of malware, whereas malicious URLs are indirect. These spam emails with malicious content (attachments or URLs) try to entice the recipient into opening an attachment or to click on a URL. Such spam emails have subject and content text that entices or alarms the recipient into acting on the disguised malicious content.

To find this type of dangerous spam emails, scanning the attachments of emails and URLs with virus scanners or against blacklists often reveals their scope and the nature of the malicious content. However, scanning emails require external resources that are often computationally expensive and difficult to maintain (Ma, Saul, Savage, & Voelker, 2009). This method of identifying spam and other spam filtering methods aim to be highly responsive to changes in spamming techniques, but are often not sufficiently flexible to handle variations in spam emails (Blanzieri & Bryl, 2008).

The task of identifying malicious content (attachments or URLs) in spam emails has been subject to limited research. Our specific definition of malicious software includes only malware and so is different from research on classifying phishing emails by analysing URLs in their content. This research should help identify one of the most harmful types of spam emails received.

In this initial work, we proposed several potential novel features for predicting malicious attachments and URLs in spam emails. We hypothesised that spam emails with malicious attachments or URLs can best be predicted only from the text content in the email subject and body. Our work also differs from related work as it is self-contained (did not require external resources such as blacklists and the like) and did not add risks of exposure to malicious content by attempts to analyse or scan dubious attachments, or by tracking URLs. We use two real world data sets obtained from two different sources. The first data set is from the Habul plugin for the Thunderbird mail client, and the second data set, Botnet, is collected from honeypots around the world to study the characteristics of email spam botnets.

We extract many features from the metadata and text content of these real world spam emails. These proposed features are: self-contained (no need to scan emails using external resources such as virus scanners and blacklists); robust (high adaptability to changes in spamming techniques); and time efficient (process many emails per second). We apply a Random Forest classifier to these select features to show their effectiveness in distinguishing risky spam emails (i.e. those with

[1] Kaspersky Lab Securelist article: "Spam in Q1 2013." (8 May 2013) http://www.securelist.com/en/analysis/204792291/Spam_in_Q1_2013

[2] Radicati Group Reports – Executive Summary: "Email Statistics Report, 2013-2017." (22 April 2013) http://www.radicati.com/wp/wp-content/uploads/2013/04/Email-Statistics-Report-2013-2017-Executive-Summary.pdf

malicious attachments) from those without malicious attachments. However, these select features are insufficient to comprehensively classify the spam emails into at risk or not (i.e. that is spam or without malicious URLs). We discuss the success and failure of our features in identifying malware associated with spam and the potential research directions that arise from this initial work.

Our contributions in this initial work are (1) developing new or novel features that do not require external resources for the task of classifying malicious spam emails, (2) evaluating these features on two real-world data sets, and (3) demonstrating how well malicious attachments can be predicted from only the content of the email itself with high classification scores. Our work aims to reduce the need to scan emails for malicious content, saving time and resources.

The paper is organised as follows. Section 2 summarises related work. Section 3 explains the structure and content of malicious spam emails, and Section 4 provides details our real world data sets. Section 5 presents our proposed 'novel' features that help to capture malicious intent in these emails. Section 6 details our evaluation methodology and Section 7 summarises our results. We discuss our results in Section 8 and conclude our findings in this initial work in Section 9.

2 Related Work

We summarise related work in respect to four aspects of our work, highlighting text and machine learning based approaches. We look at spam filtering and specifically related work on classifying malicious attachments and URLs. From the related research on the detection of vandalism on Wikipedia, we borrow some of the features helpful in that context and adapt them to our problem.

2.1 Email Spam Filtering

Spam filtering is a well-developed field with many different techniques applied to many types of spam (Blanzieri & Bryl, 2008). A survey of machine learning based approaches to spam filtering by Blanzieri & Bryl (2008) covered the ambigious definitions of spam, summarised a variety of spam detection methods and their applicability to different parts of an email, and summarised the various data sets used in this research. Their survey showed a variety of machine learning approaches that relied on certain features extracted from the email header, body, and the whole email message.

In summary, email spam filtering is a mature research field with many filtering techniques available such as rule based, information retrieval based, machine learning based, graph based, and hybrid techniques. However, identifying emails with malicious content remains a problem worthy of further investigation.

2.2 Classification of Malicious Attachments

Emails containing malicious attachments are potentially one the most dangerous types of emails as the associated malware has the potential to do significant damage to computers and to spread rapidly. The user's email usage behaviour can also change depending on the malware's capability for spreading infection. By engineering features that capture behavioural properties of email use and the content of emails, the outgoing email behaviour of users can predict when malware has compromised a computer (Martin, Nelson, Sewani, Chen, & Joseph, 2005). Applying feature reduction techniques can further improve the classification accuracy of malware propagated in outgoing mail (Masud, Khan, & Thuraisingham, 2007). These approaches aim to identify new malware by observing behaviour after infection.

For preventative solutions that do not need to scan attachments, analysing properties of the software executables can reveal malicious intent (Wang, Yu, Champion, Fu, & Xuan, 2007). Our work also aims to be preventative, but without adding the risk of infection by analysing software executables which may escape.

2.3 Classification of Malicious URLs

Research on classifying URLs for malicious intent extend beyond spam emails, because of the common nature of URLs in many Web documents and communications. Blacklisting is a highly efficient method of preventing access to malicious URLs, but it relies on discovering which URLs are malicious beforehand (Ma, Saul, Savage, & Voelker, 2011). Furthermore, blacklisting services cannot keep up with high volume spamming botnets that operate from frequently changing URLs and IP addresses (Ramachandran, Dagon, & Feamster, 2006).

To be fully effective and adaptive to new malicious URLs, creating relevant URL features or variables based on text and hosting properties for classifiers has been shown to be successful as classifiers (Ma, Saul, Savage, & Voelker, 2009; Le, Markopoulou, & Faloutsos, 2011). However, these features require many external resources such as IP blacklists, domain registration details, DNS records, and reliable geographical location of IP addresses. Although these features can be applied in the real-time classification of URLs, there are trade-offs in accuracy and processing time (Ma, Saul, Savage, & Voelker, 2009).

Other methods for the detection of malicious URLs require access to the Web pages of URLs and then performing further analysis. Parts of Web pages can be obfuscated to hide malicious intent, such as malicious Javascript code (Likarish, Jung, & Jo, 2009). However, developing many different sets of features or variables over both structure and content of the page provided a comprehensive analysis of the likelihood that a Web page may be malicious (Canali, Cova, Vigna, & Kruegel, 2011).

In this work, we do not consider all the possible features for classifying URLs as discussed by other researchers. Our focus is on using email content alone to predict if a URL is malicious. In future work, we intend to perform further analysis of these promising features used in related work and apply and where possible improve them in our identification of risky emails.

2.4 Wikipedia Vandalism Detection

In this initial work, we borrow some text features from the related field of vandalism detection on Wikipedia. The problem of vandalism detection (i.e. a malicious edit or change in the content) and detecting emails with malicious content are related and may share similar characteristics. In both cases, the text within a Wikipedia

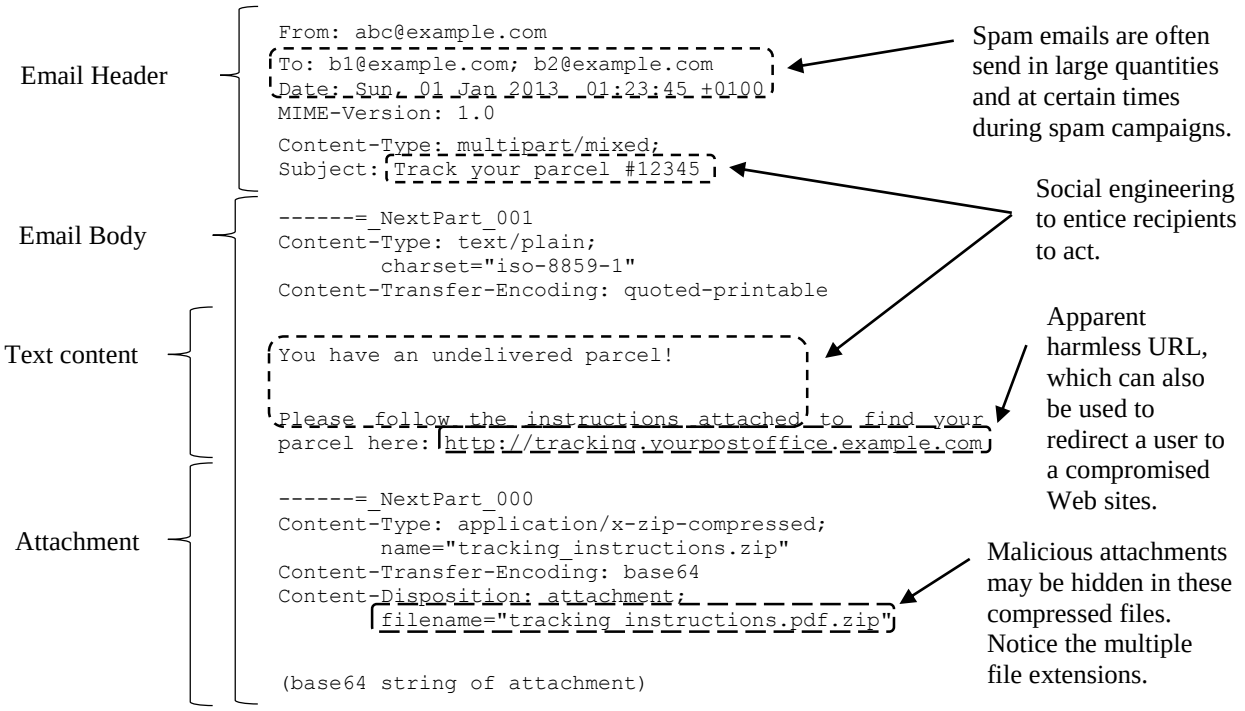


Figure 1: An example (fake) spam email with a potential malicious attachment and URL.

article and text in an email may contain content that distinguishes it from a normal article or normal (spam) email, respectively. For example, abnormal use of vulgar words or excessive use of uppercase words may hint at malicious intent. Our initial work provided for a comparison of the classification models across these two research areas, and also helped to address the problem of insufficient training samples for the testing of classification models.

The PAN Workshops in 2010 and 2011 held competitions for vandalism detection in Wikipedia, where they released a data set containing manually classified cases of vandalism. In Section 7, we describe our selected text features from the winners of the competitions in 2010 (Velasco, 2010) and 2011 (West & Lee, 2011). These text features aim to show text regularities within spam emails.

3 Malicious Spam Emails

Spam emails vary from annoying, but harmless, advertising to dangerous scams, fraudulent activity, and other cybercrime. Spam emails with malware or URLs that direct users to malware are common methods used by cybercriminals to find new victims. For example, spammers may want to expand their botnets or cybercriminals may use them to propagate their computer viruses so as to harvest passwords, credit cards, bank accounts, and other sensitive personal information. Our work aims to provide a preventative method to help reduce the propagation of malware using spam emails. Before presenting our results, we briefly describe our raw data of malicious spam emails and how cybercriminals disseminate spam emails.

Email formats are well-known, but less familiar are the raw email data that we use to construct our features. We present an example of a (fake) spam email with potential malicious content in Figure 1, stripped of irrelevant metadata. The figure shows an email in raw text format with annotations showing important parts of the email that can be used for the construction of features/variables. We have the email header that contains delivery instructions for mail servers, and the email body that can have many sections for text, attachments, and other types of attachable data. Emails are identified as spam in two ways: a user determines if an email is spam, and emails collected and identified as sourced from known spamming networks. Both scenarios for determining spam are captured in our two real world data sets.

Our example in Figure 1 shows the typical structure of a malicious spam email. The subject or text content of malicious spam emails often contains social engineering methods to manipulate recipients into first reading and then act on the email. In this case, we have the premise of a fake undelivered parcel that requires the recipient to download a compressed file (purposefully misleading with multiple file extensions). This compressed file serves the purpose of hiding malware executables from virus scanners operated/applied by mail servers. The URL in this example acts as a secondary method of delivering malicious content. Similar to attachments, malicious URLs can disguise a malicious Web site (e.g. example.com) by adding subdomains representing a known and safe Web site (e.g. tracking.yourpostoffice). Our example also shows a possible spam template, where attachments or URLs may have different names, but the same malicious intent.

Spam templates are often used in spam campaigns, where many emails are sent in a short period of time often with minor lexical variations to their content (Stone-Gross, Holz, Stringhini, & Vigna, 2011). In our example in Figure 1, variations can occur in the tracking number, attachment name, and URL. These variations are attempts to prevent basic routine spam detection methods applied by mail servers. Other obfuscation methods

include manipulation of email headers to include legitimate email addresses that also help avoid spam filtering and thus allowing more spam emails to be sent undetected.

The emergence and proliferation of botnets have allowed large quantities of spam emails to be sent in a coordinated way, and amplify cybercrime activities (Broadhurst, Grabosky, Alazab, et al., 2013). Botnets are networks of compromised computers controlled by a 'botmaster' who often rents the botnet to spammers and others that intent to use them to deliver malware. Botnets are the backbone of spam delivery, and estimates suggest that approximately 85% of the world's spam email were sent by botnets every day (John, Moshchuk, Gribble, & Krishnamurthy, 2009). The widespread use of botnets show how spammers understand and manipulate the networks of compromised computers and servers around the world to ensure high volumes of spam are delivered to large numbers of Internet users.

Overall, the use of spam emails is an important vector to propagate malware, and the forms of social engineering used in spam emails have grown more sophisticated, improving the ability to deceive many users into malware self-infection.

4 Email Spam Data Sets

We use two real world data sets from two different spam collection sources. The first comes from the Habul Plugin for Thunderbird (an offline mail client) that uses an adaptive filter to learn from a user's labelling of emails as spam or normal email. Table 1 summarizes the statistics for the Habul data set (Habul DS), which are compiled monthly. The second data set is compiled from a global system of spam traps designed to monitor information about spam and other malicious activities. The second data set we labelled the Botnet data set (Botnet DS), which were also compiled monthly. Table 2 summarizes the descriptive statistics for our larger Botnet DS. We received both data sets in anonymised form, so no identifiable email addresses or IPs are available for analysis.

For each email, we extract attachments and URLs and upload to VirusTotal[3], a free online virus checker that offers support for academic researchers, to scan for viruses and suspicious content. VirusTotal uses over 40 different virus scanners, and we consider an attachment or URL to be malicious if at least one scanner shows a positive result. For this initial study, we only focus on emails with attachments or that contains URLs to predict/identify emails with malicious content.

The Habul DS is much smaller than the Botnet DS, but has the advantage that these emails have been manually labelled as spam by recipients. This means the spam in the Habul DS has been viewed, but the Botnet DS contains spam that circulated all over the world, but without the certainty that the emails have reached their intended targets.

Both of the data sets show some similarities: nearly half of spam emails contain at least one URL, but only a small percentage were identified as malicious. In contrast,

Habul		with Attachments		with URLs	
Month	Emails	Total	Mal.	Total	Mal.
Jan	67	7	3	25	3
Feb	104	10	2	33	6
Mar	75	5	0	28	4
Apr	65	4	2	26	2
May	83	4	0	38	5
Jun	94	1	0	41	5
Jul	72	2	1	26	11
Aug	85	0	0	46	10
Sep	363	11	7	140	4
Oct	73	1	1	11	3
Nov	193	4	0	89	13
Dec	95	6	3	31	12
Total	1,369	55	19	534	78

Table 1: Habul Data Set Statistics

Botnet		with Attachments		with URLs	
Month	Emails	Total	Mal.	Total	Mal.
Jan	31,991	139	27	12,480	4
Feb	49,085	528	66	14,748	4
Mar	45,413	540	52	19,895	23
Apr	33,311	328	175	12,339	0
May	28,415	753	592	13,645	3
Jun	11,587	102	56	8,052	80
Jul	16,251	425	196	5,615	92
Aug	21,970	291	113	16,970	707
Sep	27,819	282	12	17,924	442
Oct	13,426	899	524	4,949	2
Nov	17,145	1,107	882	7,877	49
Dec	20,696	621	313	7,992	241
Total	317,109	6,015	3,008	142,486	1,647

Table 2: Botnet Data Set Statistics

many more emails that include an attachment were malicious. For each data set, there were peaks of spam that either contained malicious content or not, and which suggested different types of spam (mass propagation) campaigns. These campaigns usually shared similarities in the content of their emails, and this alone may indicate the risk of malicious content.

5 Feature Engineering

In this initial work, we explore a comprehensive set of features that help characterise email content. We borrow some of these features, as noted, from the related field of vandalism detection on Wikipedia. The aim of vandalism detection is to identify malicious modifications to articles. In particular, we borrow some text features from the winners of vandalism competitions held at the PAN Workshops in 2010 and 2011 (Velasco, 2010; West & Lee, 2011). As far as we are aware, none of the features described below have been used to predict malicious content in emails. We describe the novelty of these 'features', which we use as risk variables, in the context of their applications in related areas of research.

5.1 Feature Description

Table 3 shows our features and a summary description. Features with prefix H are email header features; prefix S are subject features; prefix P are payload features (or content of email); prefix A are features of attachments;

[3] https://www.virustotal.com/en/

and prefix U are features of URLs. We describe these features in detail below and how these groups of features are related.

5.1.1 Header Features

Features **H01** to **H04** are simple variables that capture the time when emails were sent. The times of emails have been normalised to Greenwich Median Time (GMT) to account for emails that are sent from different servers at different times. Emails propagated via a spam campaign are often sent at the same time en masse.

Features H05 and H06 are counts of the email addresses of the sender and intended recipients. Since these features have been anonymised, we only count the number of addresses. Further analysis of these email addresses is warranted, especially if addresses are at least partially de-confidentialised, because it is likely that more detailed features will help identify particular spam campaigns.

5.1.2 Text Features

These features are applied to the subject (prefix S) and payload (prefix P) of emails. Although we apply these features identically on different data, they require different interpretation for subject and payload data. For text in the subject and payload, we extract a list of words and then count the number of appearances of each word.

Feature S01 (P01) is a simple count of the number of characters in the text of the subject or payload.

Features S02 to S04 (P02 to P04) are a count of special or potentially relevant words in malicious emails. We obtained lists of these words from the English Wiktionary[4] and applied them to both data sets. This word mapping produced 27 unique pronoun words, 1064 unique vulgar words, and 5,980 unique slang words. The presence of these word form features were strong indicators of a spam email and also of possible malicious content especially when the 'payload' attempted to persuade users to download files or follow a URL. These words features were borrowed directly from the PAN Workshops (Velasco, 2010; West & Lee, 2011), but we used different sources to identify these words.

Features S05 to S12 (P05 to P12) are also borrowed from the PAN Workshops (Velasco, 2010; West & Lee, 2011). These features are self descriptive and look for patterns in the words used in the subject and payload of emails. We expect these features to distinguish genuine emails from spam campaigns because these campaigns often use email text templates (Kreibich, et al., 2009).

Features **S13** to **S23** (**P13** to **P23**) are our set of new features. These features look closely at the distribution of character types in the form of ratios. We select out the maximum and minimum of each features applied to each word to highlight any unique oddities in the words used in the email subject and payload. Our definitions of two of the less self-descriptive features are as follows:

- Character diversity was a concept also borrowed from Velasco (2010) and interpereted here as a measure of the number of different characters in a word compared to the word length: $length^{\frac{1}{unique\ characters}}$

[4] http://wiktionary.org

Feature	Description
H01-DAY	Day of week when email was sent.
H02-HOUR	Hour of day when email was sent.
H03-MIN	Minute of hour when email was sent.
H04-SEC	Second of minute when email was sent.
H05-FROM	Number of "from" email addresses, known as email senders.
H06-TO	Number of "to" email addresses, known as email recipients.
S01-LEN	Number of characters.
S02-PW	Number of pronoun words.
S03-VW	Number of vulgar words.
S04-SW	Number of slang words.
S05-CW	Number of capitalised words.
S06-UW	Number of words in all uppercase.
S07-DW	Number of words that are digits.
S08-LW	Number of words containing only letters.
S09-LNW	Number of words containing letters and numbers.
S10-SL	Number of words that are single letters.
S11-SD	Number of words that are single digits.
S12-SC	Number of words that are single characters.
S13-UL	Max ratio of uppercase letters to lowercase letters of each word.
S14-UA	Max of ratio of uppercase letters to all characters of each word.
S15-DA	Max of ratio of digit characters to all characters of each word.
S16-NAA	Max of ratio of non-alphanumeric characters to all characters of each word.
S17-CD	Min of character diversity of each word.
S18-LRC	Max of the longest repeating character.
S19-LZW	Min of the compression ratio for the lzw compressor.
S20-ZLIB	Min of the compression ratio for the zlib compressor.
S21-BZ2	Min of the compression ratio for the bz2 compressor.
S22-CL	Max of the character lengths of words.
S23-SCL	Sum of all the character lengths of words.
P01 to P12, **P13** to **P23**	Same as features S01 to S23, but for the email payload (content).
A01-UFILES	Number of unique attachment files in an email.
A02-NFILES	Number of all attachment files in an email.
A03-UCONT	Number of unique content types of attachment files in an email.
A04-NCONT	Number of all content types of attachment files in an email.
U01-UURLS	The number of unique URLs in an email.
U02-NURLS	The number of all URLs in an email.

Table 3: Email Features. Features in bold text are novel features not seen in other research areas.

- Compression ratio was defined as: $\frac{uncompressed\ size}{compressed\ size}$

In the subject of spam emails, these emphasise unique words much stronger than features S02 to S12, because of the relatively shorter length of text to the payload.

Features **S18** to **S21** are variants of the same concept of identifying particular words with repetitive characters. We use these features to account for simple misspellings of words by repeating characters. These are the most computationally intensive features, with feature **S19** on average taking 4 milliseconds (ms) per email, and features **S18**, **S20**, and **S21** on average taking less than 1ms. All other features on average took between 0.0050ms and 0.0100ms per email. Note that these are the time taken to generate a single feature and does not include parallelisation and batch pre-processing of the required data.

Type	Attachments				URLs			
Data Set	Habul		Botnet		Habul		Botnet	
Month	Feature	Score	Feature	Score	Feature	Score	Feature	Score
Nov	S05-CW	0.1115	**S21-BZ2**	0.1066	**U02-NURLS**	0.0875	**H01-DAY**	0.0628
	S23-SCL	0.0812	**S20-ZLIB**	0.0860	**U01-UURLS**	0.0719	P01-LEN	0.0562
	S09-LNW	0.0741	**S17-CD**	0.0722	P09-LNW	0.0530	**P23-SCL**	0.0536
	S15-DA	0.0665	**S19-LZW**	0.0581	**P21-BZ2**	0.0508	**H03-MIN**	0.0531
	H02-HOUR	0.0628	**S22-CL**	0.0451	P08-LW	0.0406	**H02-HOUR**	0.0476

Table 4: Top 5 features determined by Random Forest classifier. Scores are the information entropy of features.

5.1.3 Attachment Features

These features (prefix A) are specific to spam emails with attachments. We do not use URL features with these attachment features. Our initial investigation looks only at simple, but novel, features of how attachments appear in emails. In particular, we count the number of files and the declared content types (such as image or zip files). For spam emails with attachments, malicious attachments may appear as the only attachment in emails, or may attempt to hide among many different types of attachments. In future work, we aim to generate more features from filenames or other attributes of attachments and so hope to avoid the need to scan for malicious content.

5.1.4 URL Features

These features (prefix U) are specific to spam emails with URLs. We do not use these features in conjunction with the attachment features, but they are novel to our classification task. In future work, we intend to apply more complex text analysis specifically for URLs in order to extract features that may distinguish URLs that are designed to direct users to websites (with and without malicious content). For example, this may occur when a number of URLs share a common domain names or common access pages.

5.2 Feature Ranking

With many varieties of potential variables or features, we find such features important to our classification task and thus we compare them across the two data sets. We compare them by using the Random Forest classifier that produced a ranking of these features based on their entropy scores (Pedregosa, et al., 2011). Please see Section 7 below for a description of our classifier and classification results.

The entropy score measures the additional information gained when splitting a decision tree (in the forest) is further split on that feature. The aim is to have the most homogenous decision branches after a split, and which improves classification results. For example, for emails with attachments in the Botnet data set, we gain more than twice as much information by splitting on feature S21 (0.1066) than when we split on the feature S22 (0.0451). To account for the overall randomness of the Random Forest classifier, we present the average scores of 10 training iterations in Table 4 for the data split of the month of November in both data sets (details below in Section 7). We bold features that are our novel contributions.

From Table 4, we see the majority of the best performing features are our proposed features for this classification task. In particular, for the larger Botnet DS with a larger number of emails, we find our selected features perform consistently well. The variety of features shows that no single feature dominates among the top 5 scores across data sets, and attachments and URLs. This result further emphasised the need for a feature rich model to capture variations in different types of spam emails containing malicious content.

For the Habul data set, predicting malicious attachments and URLs from email content shows different but also important features. For attachments, we find features S05, S23, S09, and S15, suggested emails with capitalised words containing letters and digits in the subject line. This apparent formality in the subject line attempts to mimic legitimacy in order to gain the trust of recipients to open the email and download the attachments. The presence of feature H02 also suggest these malicious spam email may be originating from a spam campaign. For URLs, we find URL and payload features are relevant when U02 and U01 appear together perhaps indicative that a few unique URLs are at risk. This suggests malicious spam emails contain fewer URLs with associated content designed to persuade recipients to click on those URLs.

For the Botnet data set, we find the subject of the email to be the strongest predictor of the presence of malicious attachments, whereas when the email was sent was a good predictor of malicious URLs. For attachments, we found the email subjects with low compressibility of words for all three compression algorithms (S21, S20, and S19), combined with many different characters (S17), and long words (S22) were also useful predictors. This suggested subject lines with seemingly random characters, which may trigger the recipient's curiosity to download the (malicious or risky) attachments associated with the email. For URLs, the

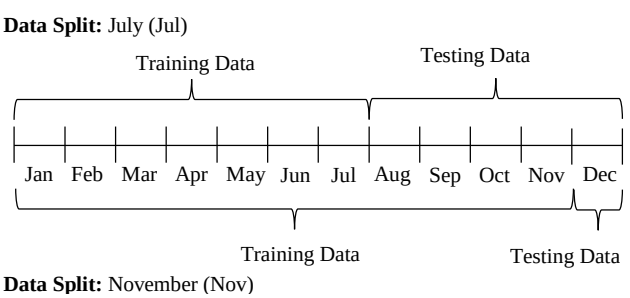


Figure 2: Illustration of splitting data into training and testing sets.

time features are highly predictive along with the length of the email's content. Again this indicates spam campaigns with email templates that offer 'strange'/unconventional subject text may induce the curiosity of recipients to download the associated attachments.

We found similarities in the features of both two data sets that were identified as predictive of the presence of malicious attachments and URLs. Emails with attachments indicate their likely malicious intent by their subject line. For those emails with URLs, the frequency or number of URLs, the text, and the time when the emails were sent were predictive of the risk of malicious intent.

6 Evaluation Methodology

As our data sets are already partitioned into months, we combine the data sets into months, then learn on the earlier months and test our classifier on the later months. Figure 2 illustrates our data splitting process into training and testing data sets for the months of July and November. For example, for the months of July, we train on all spam emails with malicious content from January to July, and then test the model on spam emails with attachments or URLs from August to December. This shows the effects of different training sample sizes on classification quality, and the adaptability of the classifiers used.

We combine the feature sets differently for classification of attachments and URLs. For attachments, we choose features with the prefixes of H, S, P, and A. For URLs, we choose with prefixes of H, S, P, and U.

We use three classifiers to evaluate our features: Naïve Bayes (NB), Random Forest (RF), and Support Vector Machine (SVM); and we use the evaluation metrics from the Scikit-learn toolkit (Pedregosa, et al., 2011). The NB and SVM classifiers are commonly used in spam classification, whereas the RF classifier is not commonly used (Blanzieri & Bryl, 2008). We performed a standard grid search with 10-fold cross validation to determine the best parameters for each classifier.

We measure the performance of the classifier using the average precision score, also known as the area under the precision-recall curve (AUC-PR), and the accuracy (ACC). The AUC-PR score gives the probability that our classifiers correctly identify a randomly selected email that also contains with malicious content is correctly labelled by our classifier. The ACC scores give the percentage of spam emails that are correctly classified as containing malicious content or not. These measures are defined from four different scenarios from spam emails with attachments or URLs: true positive (TP), emails correctly classified as containing malicious attachments or URLs; true negative (TN), emails correctly classified as non-malicious; false positive (FP), emails incorrectly classified as malicious; and false negative (FN), emails incorrectly classified as non-malicious. From these definitions, we have the positive precision value (precision) as $PPV = \frac{TP}{TP+FP}$, and the true positive rate (recall) as $TPR = \frac{TP}{TP+FN}$. By plotting PPV against TPR with instances of positive and negative values, we obtain a precision-recall (PR) curve, and calculate its area. We calculate the accuracy as: $ACC = \frac{TP+TN}{TP+FN+FP+TN}$.

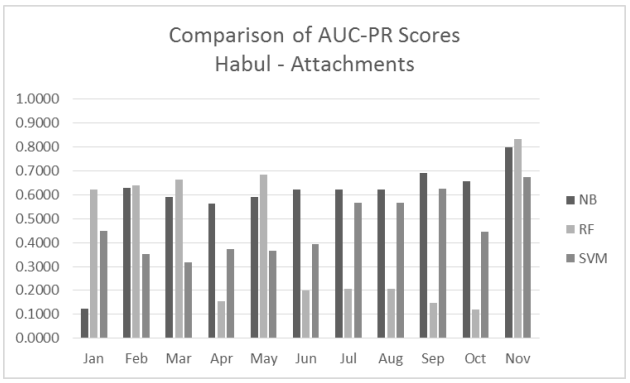
As we are the first (as far as we are aware) to use these methods to predict malicious content in emails. There are no comparable baseline measures available for comparison. In future work, we plan to expand our set of URL features and compare these to related work on the prediction of phishing URLs in emails. For now, we present our classification results and discuss our findings.

7 Classification Results

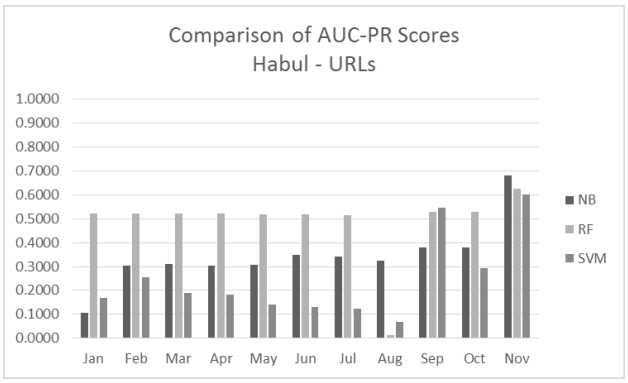
We compare the classification results for the three classifiers in Figure 4 for AUC-PR scores and in Figure 3 for ACC scores. In Figure 5, we compare our classification results for the SVM classifier. We compare the data splits in each figure for two different data sets and three different classifiers. Our figures also show the effect of the accumulation each month of the spam data on predicting malicious emails in the subsequent months.

For emails with attachments, predicting whether attachments are malicious was successful on the Botnet data set, reaching a peak AUC-PR score of 0.9261 (Figure 4 (a) and (c)). The low AUC-PR score for the training set split in January was expected as we have insufficient data to observe whether attachments are malicious in the subsequent months (February to December). The classifier shows very poor performance on the Habul data set for many data splits (Figure 4 (a)). The reason is clear from Table 1, where we see again very few emails with attachments for the classifier to learn from. In some months corresponding with the other data splits (e.g. August), we do not have any or few emails with malicious attachments to learn from. The low AUC-PR (Figure 4 (a) and (c)) and high ACC scores (Figure 3 (a) and (c)) show the high risk of false negatives as many emails *with* malicious content are not classified correctly. However, for the data split of November, where we have more training data when compared to the testing data, the three classifiers perform well with AUC-PR scores for both data sets above 0.8 (Figure 4 (c)). The classifier performs well for the Botnet data set for attachments as we have many training samples for each month as seen in Table 2.

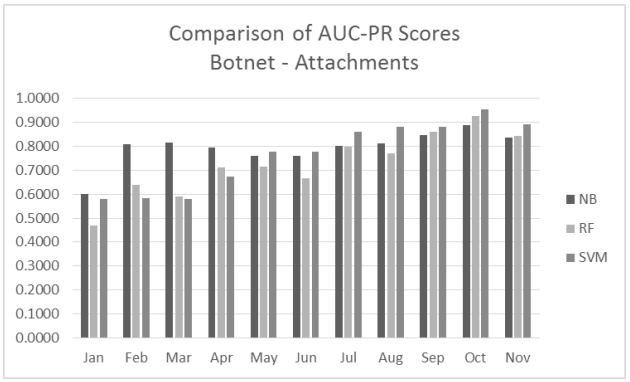
For emails with URLs, all three classifiers show poor performance with AUC-PR scores (Figure 4 (b) and (d)) around or below 0.5. This means for an email with malicious URLs, the classifiers NB and SVM will label them correctly less than 50% of the time, worse than a random guess. However, we have very high accuracy scores for the classifiers RF and SVM in both data sets (Figure 3 (b) and (d)) for most data splits. The low AUC-PR scores and high ACC scores show the classifiers cannot distinguish emails with malicious URLs from emails with no malicious URLs. The reason for the poor performance of the classifier is the overwhelming number of emails with no malicious URLs. Our proposed features are insufficient to distinguish malicious URLs as they are generally uncommon or underrepresented in the data set, as seen in Tables 1 and 2. This means our method cannot determine malicious URLs only from the text of emails with URLs.



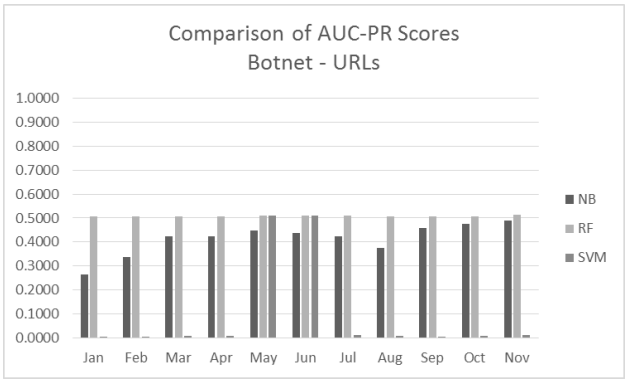
(a)



(b)

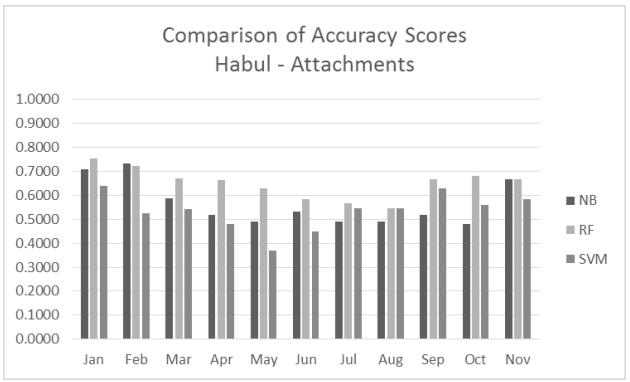


(c)

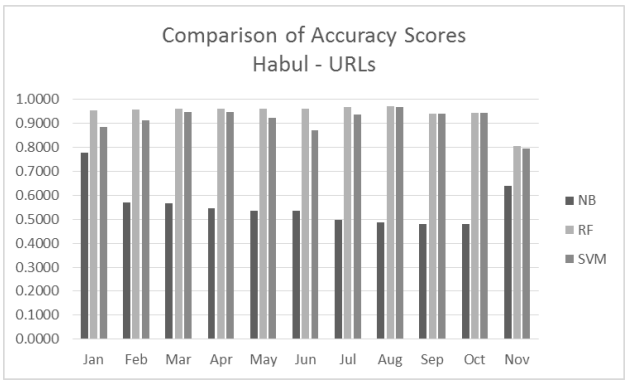


(d)

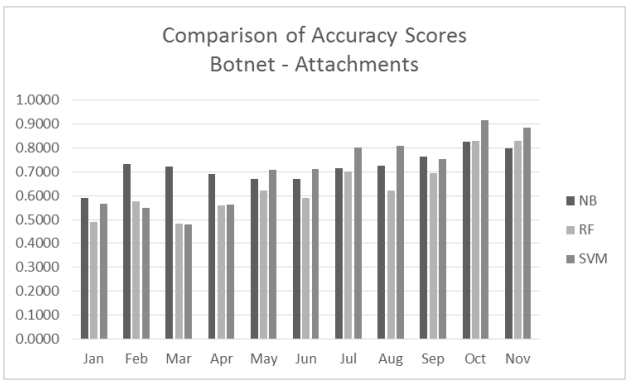
Figure 4: Comparison of AUC-PR scores for three classifiers Naïve Bayes (NB), Random Forest (RF) and Support Vector Machine (SVM)



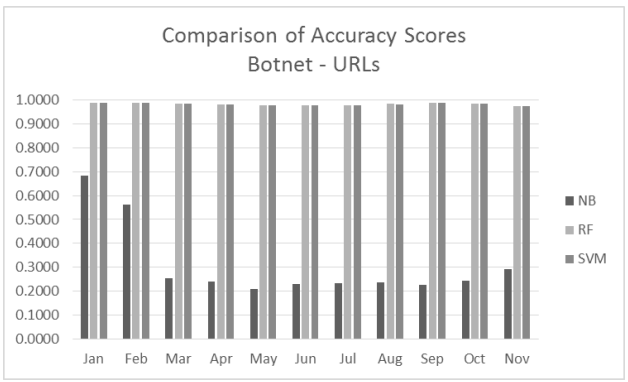
(a)



(b)



(c)



(d)

Figure 3: Comparison of Accuracy (ACC) scores for three classifiers Naïve Bayes (NB), Random Forest (RF) and Support Vector Machine (SVM)

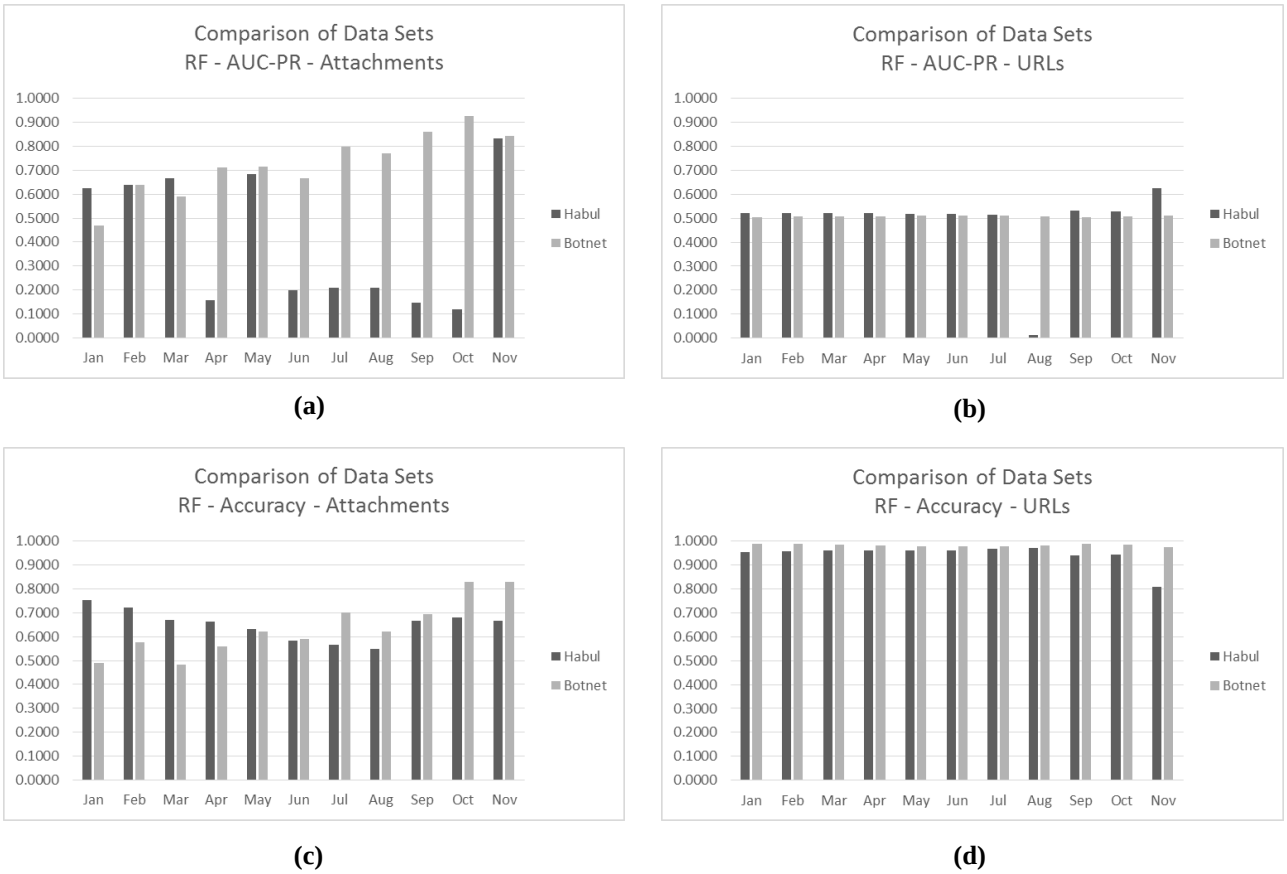


Figure 5: Comparison of RF classification scores across different data sets.

In Figure 5, we compare the classification results between our two data sets for the most robust classifier: Random Forest (RF). As discussed above, the comparatively numerous training samples in the Botnet data set allow for a high classification performance as measured for both AUC-PR and ACC scores. The data split of November which has the most training samples also showed high classification scores, especially in the Habul data set, where there fewer data samples. Figure 5 shows the larger Botnet data set was generally better for predicting malicious content in emails.

Overall, our initial work shows the viability of predicting whether attachments and URLs in emails are malicious. Our proposed feature-rich model shows our hypothesis is true for malicious attachments as those emails can be predicted from the email subject and payload with high AUC-PR and ACC scores. For URLs, the subject and payload of the emails does not indicate malicious URLs. In future work, we look to add more features for URLs, focusing on the lexical content (as in related work) to avoid drawing on external resources, such as blacklists. Our initial success with predicting malicious attachments reduced the need to scan every attachments for malicious content. When the data set is large and has sufficient numbers of risky emails, we can reduce the need to scan over 95% of emails with attachments (from AUC-PR scores) by analysing alone the text in emails with attachments.

8 Discussion

These initial findings are encouraging as they suggest we may be able to correctly identify over 95% of the 6 billion emails with malicious attachments sent everyday (see Section 1) by analysing only the email subject and text content. While our success was not as high when identifying malicious URLs, our results show a manually labelled data set of spam emails with malicious URLs (Habul) can outperform (see Figure 4 (b) and (d)) an automated collection of spam emails with malicious URLs (Botnet). Our results again suggest it may be possible to reduce the need to scan large quantities of emails for malicious content

The main advantage of our approach is the self-contained sets of features extracted from only the email itself may be sufficient to identify risky email, without recourse to external resources such as virus scanners or blacklists. This means our machine learning algorithms can quickly adapt to changes and evolution of spam emails. This can be later verified its results when scanners and blacklists have been updated.

A limitation of our approach is the descriptiveness of our proposed sets of features. Our results show that the features are more suitable for predicting malicious attachments than malicious URLs. This suggests emails with malicious URLs do not have sufficient commonalities when the subject or text content are used to predict the malicious intent of its URLs. Some exploit kits such as the Blackhole Exploit Kit simply inserts malicious URLs into emails without changing their content (Oliver, et al., 2012). Thus, non-malicious spam emails can become, via this method, malicious without any changes to their original spam content. To resolve this limitation, in future work we intend to add lexical features from related work (see Section 2.3) to add to our

own tests for the risk of malware embedded in URLs, and compare their classification performance.

Another limitation is the possibility that a few spam campaigns have been overrepresented in our data sets. We have not yet performed a detailed spam campaign analysis and this would be another research topic worth following up. Reviewing statistics from Tables 1 and 2, for the Habul data set, we found 13 unique malicious attachments (in 19 emails with malicious attachments), and 70 unique malicious URLs (in 78 emails with malicious URLs); and for the Botnet data set, we found 847 unique malicious attachments (in 3,008 emails with malicious attachments), and 889 unique malicious URLs (in 1,647 emails with malicious URLs). If each unique attachment or URL represented one spam campaign (thus having similar features in campaign emails), then the diversity of these spam campaigns would be high, and this would strengthen our results because the classifiers can recognise a wide variety of spam campaigns with high reliability as measured by the AUC-PR and ACC scores for malicious attachments. In future work, we aim to address this issue by performing spam campaign analysis and to see if this will influence on classification results.

Overall, we partly confirm our hypothesis that emails with malicious attachments can be predicted from the features of the email text. Our evaluation on two real-world data sets composed only of spam emails, show the effects of data set size, the cumulative learning of potential spam emails over a year, and the importance of different features useful for classification. The work of identifying the more dangerous types of spam email remains important if we are to prevent this vector for cybercrime by limiting exposure of malware to potential victims.

9 Conclusion

We presented rich descriptive sets of text features for the task of identifying emails with malicious attachments and URLs. We use two real-world data sets of spam emails, sourced from a manually labelled corpus (Habul) and automated collection from spamtraps (Botnet). Our initial results show that emails with malicious attachments can be reliably predicted using text features extracted only from emails, without requiring external resources. However, this is not the case with emails with malicious URLs as their text features do not differ much from emails with non-malicious URLs. We compared the classification performance for three classifiers: Naïve Bayes, Random Forest, and Support Vector Machine. We compare the selected features across our two data sets with the Random Forest classifier generally performing best. We have discussed the effects of differences in size of data set, the potential overrepresentation of spam campaign emails, and advantages and limitations of our approach. Our initial success suggested we might correctly identify over 95% of emails with malicious attachments without needing to scan the attachments. If this can be confirmed in subsequent research, a huge potential saving in resources used to detect and filter high-risk spam may be achieved. In addition, the methods will assist in the prevention of cybercrime, given that an estimated 6 billion emails with malicious attachments are sent every day.

In future work, we intend to add features to improve the classification of emails with malicious URLs. Indeed this form of delivering malware appears to be both evolving and now seem preferred to attachments containing malware. We aim to extract more features from the header of emails, such as graph relationships of common (anonymised) email addresses that could prove useful as alternative classifiers. One important unresolved issue is the possible effects of deliberate (and possibly repetitive) spam campaigns on classification results. We hope both to increase the size of scope of our real world data sets (adding for example prominent email data sets) and plan a comprehensive analysis combining and testing features, taking into account spam campaigns.

10 Acknowledgements

The research is funded by an ARC Discovery Grant on the Evolution of Cybercrime (DP 1096833), the Australian Institute of Criminology (Grant CRG 13/12-13), and the support of the ANU Research School of Asia and the Pacific. We also thank the Australian Communications and Media Authority (ACMA) and the Computer Emergency Response Team (CERT) Australia for their assistance in the provision of data and support.

11 References

Blanzieri, E., & Bryl, A. (2008). A survey of learning-based techniques of email spam filtering. *Artificial Intelligence Review, 29*, 63-92.

Broadhurst, R., Grabosky, P., Alazab, M., Bouhours, B., Chon, S., & Da, C. (2013). Crime in Cyberspace: Offenders and the Role of Organized Crime Groups. *Social Science Research Network (SSRN).*

Canali, D., Cova, M., Vigna, G., & Kruegel, C. (2011). Prophiler: a fast filter for the large-scale detection of malicious web pages. *Proceedings of the 20th international conference on World wide web.*

John, J. P., Moshchuk, A., Gribble, S. D., & Krishnamurthy, A. (2009). Studying Spamming Botnets Using Botlab. *NSDI.*

Kreibich, C., Kanich, C., Levchenko, K., Enright, B., Voelker, G. M., Paxson, V., & Savage, S. (2009). Spamcraft: An inside look at spam campaign orchestration. *Proc. of 2nd USENIX LEET.*

Le, A., Markopoulou, A., & Faloutsos, M. (2011). PhishDef: URL Names Say It All. *Submitted to IEEE INFOCOM.*

Likarish, P., Jung, E., & Jo, I. (2009). Obfuscated malicious javascript detection using classification techniques. *Malicious and Unwanted Software (MALWARE), 2009 4th International Conference on.*

Ma, J., Saul, L. K., Savage, S., & Voelker, G. M. (2009). Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs. *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining.*

Ma, J., Saul, L. K., Savage, S., & Voelker, G. M. (2009). Identifying Suspicious URLs: An Application of Large-Scale Online Learning. *Proceedings of the 26th Annual International Conference on Machine Learning*.

Ma, J., Saul, L. K., Savage, S., & Voelker, G. M. (2011). Learning to Detect Malicious URLs. *ACM Trans. Intell. Syst. Technol.*

Martin, S., Nelson, B., Sewani, A., Chen, K., & Joseph, A. D. (2005). Analyzing Behavioral Features for Email Classification. *Second Conference on Email and Anti-Spam (CEAS).*

Masud, M. M., Khan, L., & Thuraisingham, B. (2007). Feature based techniques for auto-detection of novel email worms. *Advances in Knowledge Discovery and Data Mining.*

Oliver, J., Cheng, S., Manly, L., Zhu, J., Paz, R. D., Sioting, S., & Leopando, J. (2012). Blackhole Exploit Kit: A Spam Campaign, Not a Series of Individual Spam Runs. *Trend Micro Incorporated.*

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research, 12*, 2825-2830.

Ramachandran, A., Dagon, D., & Feamster, N. (2006). Can DNSBased Blacklists Keep Up with Bots? *In Collaboration, Electronic messaging, Anti-Abuse and Spam Conference.*

Stone-Gross, B., Holz, T., Stringhini, G., & Vigna, G. (2011). The underground economy of spam: A Botmasters perspective of coordinating large-scale spam campaigns. *In USENIX Workshop on Large-Scale Exploits and Emergent Threats.*

Velasco, S. (2010). Wikipedia vandalism detection through machine learning: Feature review and new proposals. *Lab Report for PAN-CLEF*.

Wang, X., Yu, W., Champion, A., Fu, X., & Xuan, D. (2007). Detecting worms via mining dynamic program execution. *Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm 2007. Third International Conference on*.

West, A. G., & Lee, I. (2011). Multilingual Vandalism Detection using Language-Independent {\&} Ex Post Facto Evidence - Notebook for PAN at CLEF 2011. *CLEF (Notebook Papers/Labs/Workshop).*

A Novel Process of Group-oriented Question Reduction for Rule-based Recommendation Websites

Lin Chen
Queensland University of Technology
2 George St, Brisbane
Australia
l33.chen@student.qut.edu.au

Daniel Emerson
QUT
2 George St, Brisbane
Australia
daniel.emerson@qut.edu.au

Richi Nayak
QUT
2 George St, Brisbane
Australia
r.nayak@qut.edu.au

Abstract

Several websites utilise a rule-base recommendation system, which generates choices based on a series of questionnaires, for recommending products to users. This approach has a high risk of customer attrition and the bottleneck is the questionnaire set. If the questioning process is too long, complex or tedious; users are most likely to quit the questionnaire before a product is recommended to them. If the questioning process is short; the user intensions cannot be gathered. The commonly used feature selection methods do not provide a satisfactory solution. We propose a novel process combining clustering, decisions tree and association rule mining for a group-oriented question reduction process. The question set is reduced according to common properties that are shared by a specific group of users. When applied on a real-world website, the proposed combined method outperforms the methods where the reduction of question is done only by using association rule mining or only by observing distribution within the group. .

Keywords: Question reduction, Clustering, Classification

1 Introduction

This paper examines several websites that provide options for users to buy products online after they answer a set of questionnaire. For example, carsales.com.au recommends cars based on the answers that the users provide. Another example is iselect.com.au that offers a range of insurance products after the user fills in a questionnaire. Choosing the right finance product is critical to customers in order to adequately protect themselves from risks (Rokach, et al, 2013). Traditionally, due to the complexity associated with finance products, brokers with the domain knowledge have assisted customers to find the right finance product (Burke, 2008).

Web technology makes online finance product recommendation possible by allowing users to answer some set questions and receive the desired products suggestion and, consequently avoiding the need of paying commissions to brokers (Rokach, et al, 2013). This enables the businesses to gather the users' inputs and to recommend the products matching the user needs and preferences. This benefits both the users and the businesses. Businesses save a considerable amount of resources and efforts by providing the desired products to users online (Lin et al, 2010). Similarly users receive financial benefits by avoiding the brokers in between and buying directly from the businesses (Rokach, et al, 2013).

Usually, domain experts would devise a set of questionnaire for the website after a rigorous consultation with all stakeholders. The website allows the customers to answer these questions sequentially. The numbers of questions asked are fixed, independent of who answers the questions and, how the customers answer the questions raised previously, except for some optional questions.

In prior versions of these systems, , a large number of questions are included in the rule-bases recommender system, depending upon the number of products offered (Nakash, et al, 2006). Consequently, a considerable proportion of customers exit the websites before the products are recommended to them. Data analysis of the website under this study shows that 40% of customers leave before they receive the recommendations.

A condensed questionnaire set is critical for ensuring questionnaire completion and receipt of product recommendations.

A possible approach to deal with this problem is to reduce the questionnaire set using feature selection methods (Lin et al, 2010). Feature selection finds correlations among questions asked, and removes redundant, irrelevant, and noisy questions.

However, the feature selection process fails to provide a satisfactory solution. It reduces the number of questions by removing the questions that are answered by fewer customers. However, each question has been carefully crafted by a domain expert and it exists because it is closely related to the type of product to be recommended. Thus a more sophisticated approach is required to develop a minimal questionnaire set.

In this paper, a method called *group-oriented question reduction* is proposed to reduce the size of the questionnaire set. It directs customers to subsequent questions depending upon how the preceding questions were answered. The method firstly analyses the questionnaire set and identify the reason why the feature selection does not work for the elimination of questions for the rule-based recommender systems. Getting hints from this analysis and collaborative filtering concept are used. Past customers who provide similar answers to the question set are considered similar and are grouped into the same cluster. These customers share more similarities

with each other than those users who fall into different groups. A clustering technique is able to identify the groups. However, how the groups are formed is unknown.

To further understand the clusters characteristics, we apply decision tree classification for identifying a subset of a clustering group by leaf node. By observing the distribution of instances belonging to a given leaf node and conducting association rule mining, prevalent properties and rules are identified that lead to a reduction in the questions for customers who will take this path.

When a new user visits the website, several common questions will be asked. Depending on the response of the new user, the subsequent questions will be asked and the system will decide which leaf node the new user belongs to. Because the common properties are discovered from the distributions and association rule mining, some questions do not need to be raised. Consequently, the questionnaire set is reduced and personalised according to users' preferences.

The proposed method is evaluated with the dataset for a commercial insurance recommendation website. Results show that the proposed method is able to achieve reduction in questionnaire set. Empirical analysis shows that the results are much better than the benchmark methods where the reduction of question is done only by using association rule mining or only by observing distribution within the group.

The contribution of this paper is twofold: (1) a novel data mining based method to produce the condensed questionnaire set to be used in rule-based recommenders is described; and (2) the method has been successfully tested in a commercial insurance recommendation website. To our best of knowledge, this work is the first of its kind to utilise data mining in rule-based recommenders for user gratification.

In the following section, the related work is discussed. In section 3, the background knowledge is presented. In section 4, the reason why feature selection is not suitable for this work is discussed followed by the proposal of our method. Section 5 details how the experiments are conducted for each step of proposed method. Then results are discussed. In section 7, conclusions are drawn.

2 Related Work

Recommenders systems have been successfully utilised in websites in providing information interested to users (Burke, 2008). There are three types of recommender systems; collaborative, content-based and rule-based (Zanker, 2008). Collaborative and content based recommender systems rely on users rating on items offered as well as detailed information about items in order to determine similar users and their possible interests (Burke, 2008). Performance of these systems depends upon the availability of this information. Different from collaborative and content based recommender systems, rule-based systems can avoid the cold-start problem for new users or new items. Rule-based recommender systems engage in a conversational interaction with users to collect explicit user preference and requirements (Zanker, 2008). They use the collected information about users and products, and determine what products meet users' requirements according to rules implemented in the system (Burke, 2000). Usually, the rules are obtained from the domain experts. In some cases, rules are generated from an automatic process where association rule mining is applied on a user item dataset (Zanker, 2008). The problem of deriving a condensed rule-set is significant as rules are backbone for building a successful rule-based recommender system (Zanker, 2008).

Another related research area is feature reduction. Two classes of feature reduction strategies are commonly applied: feature selection and dimensionality reduction (Janecek, 2008). Feature selection is a process of finding the subset of features to remove redundant or irrelevant features from the data set as they can lead to improved classification accuracy and/or reduction in computational cost. While the dimension reduction (or factor analysis) process produces linear combinations of the original attribute set so that the size of attributes is lowered, one main problem with dimension reduction is how much an original attribute contributes is often lost.

The purpose of this study is to reduce the size of questionnaire set. Consequently, dimensionality reduction is not applicable in this research as it does not tell which question is less important for reduction. To our best knowledge, we have not found literature on using feature reduction in rule-based recommender systems.

3 A Case Study: The Insurance Recommendation Website

Insurance companies protect people or organizations from the risk of loss in exchange for premium. The insurance company sells protection packages, often referred to as policies.

- **Insurer:** is the organization that provides the insurance product.
- **Customer:** is the potential insuree who is searching for one or more insurance products.
- **Service** (recommended): is a general insurance product and is not tied to the product available in the insurance company. For example, Compulsory Third Party (CTP) insurance is a service and it is not tied to the company product.
- **Product** (recommended): insurance product available in the insurance company. One product corresponds to one brand. A brand may have many products.

The insurance recommendation website under study offers commercial insurance products to users by asking a series of questions. Once the questionnaire is answered, a shortlist of products is presented. The user may explore the products for the following reasons; to gain a better understanding of their needs, to compare the products offered by other insurers, and subsequently consult the bank to get the recommended products. The questions asked online are multiple-choice and they are related to customers' business structure, business locations, employees, vehicles, equipment and assets, products and stocks, and salary. Some questions require a single answer whereas, some questions accepts multiple answers. Table 1 shows the list of questions divided into the above mentioned categories. Depending on the

answers, more questions may be asked. For example, if the answer to the question "*Do you own stock or material*" is "*Yes*", then two following questions will be asked. Otherwise, the questions will not be asked.

Figure 1 shows the example of an extended question. After all the required questions have been answered, the available services that best match the user need and preferences will be presented. The customer can choose the service(s) of interest from the disclosure of available products that they can purchase.

Type of Questions	Questions
Business structure	• Business type • Business legal structure • Business operation • Business plan
Business location	○ Type of premises • Number of business premises • Postcode of premises • Ownership of premises ○ Regions of other business premises* • Business conducted overseas
Employee	• Number of Owners in the business • Number of employees • Number of Independent contractors • Way of pay employees salary* • Business pay your salary*
Vehicles	• Number of light vehicles • Heavy vehicles ○ Ownership of light vehicles*
Equipment and Assets	• Mechanical or electrical equipment • Portable tools • Capital expenditure • Purchase or lease equipments or vehicles
Products and stocks	• Own stock material • Stock in temperature controlled environment* • Ship goods* ○ Import or Export*
Money	• Annual turnover ○ Type of way to pay others ○ Type of way other pay you • When customer pay you • Way to pay monthly operating costs

Table 1. A sample of the Type of Questions

• indicates compulsory questions; ○ indicates extended question depending on the answers given previously under the same question type; * indicates that multiple choices of answers are allowed. Otherwise only a single choice answer is allowed.

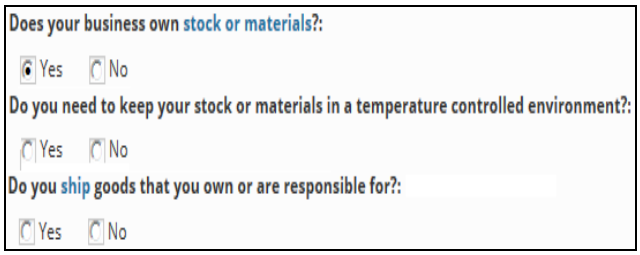


Figure 1. Extended Questions.

4 The Proposed Data Mining based Method

As stated above, the objective of this study is to develop a data mining based method for the diminution of questions in rule-based recommender web sites. The commonly used technique of feature selection used to reduce the number of features (which are questions in this case) was examined in section 4.1 Feature selection is a process of selecting a subset of the original features according to certain criteria. The reduction of the number of features is done by removing irrelevant, redundant, or noisy features. We conducted an analysis whether feature selection method would be suitable for questions reduction in this case.

4.1 Feature Selection analysis

A number of feature selection techniques require a predetermined target for evaluation purpose. We explain the process how the target was selected in the chosen case study. As described above, the aim of a rule-base recommender website is to provide advice to the customer and to allow the customer to select some of the recommended products. A customer can select multiple products. The user log data stored the customer selection. For the feature selection purpose, we are not interested in the specific product or service that the customer has selected. Instead, we are interested in knowing whether the customer has completed the process and made a selection after going through the questionnaire process.

Thus the content of the selection was not set as target, otherwise the feature selection would find the attributes that are closely related to the content of selection.

Thus a target attribute called *Product_selected* was derived as follows. If a customer has selected one or more products before the online session is closed, the *Product_selected* is set to "*Yes*", otherwise it is set to "*No*". The *CfsSubEval* algorithm (witten, 2011) is used for feature selection as the evaluation of a subset of attributes considers both the individual predictive ability of each attribute and the degree of redundancy between attributes.

The results shows that the following attributes would be considered for deletion based on the results of feature selection: *region; business postcode; business_premises_locations;business_premises_multiple ; business_primary_postcode; international_locations; export_goods; home_postcode; import_export; pay_salary_sole; and temp_controlled.*

The data distribution reveals that the subset of attributes, in the removal list, can be identified as questions that are answered by only few people.

This recommendation of removing questions cannot be considered in practice. As every question is well-designed

by the domain experts and each question is tied to the type of insurance products that are being offered to the customer. For example, the question about *region* is closely related to the product availability as some insurance products are only available to businesses in certain regions.

The problem of using feature selection lies in the fact that the solution offered is either the complete removal of the questions or keeping some of them regardless of the customer usage patterns. For example, the feature selection result suggests that "export_goods" should be removed regardless of the customer who answers the question. It is found that the customers who have revenue less than 75,000 do not export goods. By knowing this, the question should not be asked to customers who own revenue less than 75,000. To overcome this problem, dependency of the questions is identified by analysing customers' usage patterns and finding associations among questions. The question reduction then happens with the associated questions. A data mining based method combining clustering, decision tree and association mining is proposed in this study for finding a condensed rule-set.

4.2 The Group Oriented Question reduction

Association rule mining is a commonly used technique to find the associations among items. In this research, associations among customers' responses can be found and the number of questions can be reduced by using this association. Specifically, if response to a question appears in the antecedent, the response to another question that appears in the consequent can be known from the mined association rule, and therefore, the question appears in the consequent can be saved from asking.

Association mining can be applied to the whole dataset and the associations among values can be defined for question reduction. The problem with this approach is that associations found are weak due to the variations in the responses of the entire population. There is not much commonality in the responses and it results in sparse representation.

We propose to overcome this issue of heterogeneity by applying clustering on customers' responses to find groups of responses with a certain degree of similarity and put together 'alike' users. We then apply the decision tree classification to explain the clustering process through heuristic rules. By observing the distribution of instances that fall under a leaf, common properties can be explored. Further, the association rule mining is applied to the leaf node and the relation between responses can be obtained for the question reduction.

4.2.1 Clustering

In order to extract rules from the groups of alike customers based on the website usage, the groups have to be formed first. The following steps are involved to obtain the groups;

1. Data Pre-processing: All of the attributes, except those related to products and services, are incorporated into the clustering process. As mentioned before, a question may include multiple values as a response. In this case, the values have to be separated and transformed. With each attribute-value pair representing a question, the values allowed to the newly formed question are "Yes" and "No".
2. Clustering: Apply a clustering algorithm on the processed data. The form of clustering selected for this study is probabilistic. The probabilistic expectation maximization (EM) is included as it is considered as "more principled and overcoming some of the problems of the heuristic methods" (Witten, 2011). The probabilistic method calculates the likelihood of an instance being in each of the clusters and selects the best cluster. EM provides a model goodness measure, *log likelihood.*
3. Selection: Selection of the optimal clustering setting.

4.2.2 Classification

Decision tree classification is applied on the users' responses and the corresponding clusters assignments to explain how the clusters are formed. Decision tree is selected due to (1) the capability of modelling the complex and non-linear relationships presented in dataset and (2) because the generated result is usually of high accuracy and easy to understand. The steps involved in applying the classification method to the clusters were;

1. Store the clustering assignment for each instance generated in previous step as the target.
2. Apply the decision tree classification tree algorithms on this multi-class dataset. The test regime included the follow algorithms: (1) Decision Stump which is a decision tree with the root directly connected to its leaves; (2) J48 which employs the well-known C4.5 algorithm; and (3) Random Forest which constructs a forest of random trees generated by selecting *k* random attributes at each node of each tree.
3. Select the algorithm with the best classification results.

4.2.3 Association Rule Mining

Association rule mining was applied to instances in each leaf of the decision tree, placed by the previous classification step. This method provided association rules from the attributes of each cluster, showing associated attribute values. The following steps were involved in executing this stage;

1. Prepare the dataset. For each leaf, gather the instances that falls under the leaf.
2. Select the support and confidence threshold. Selecting the right support and confidence threshold is a non-trivial process. In general, high values for these thresholds are preferred. In the experiments, the support is set to 0.3 and the confidence is set to 0.9. These two settings are the highest thresholds which results in rules being returned. No rules are returned when the threshold of support and confidence is set higher than these two thresholds.
3. Conduct association rule mining using the Apriori algorithm with the selected thresholds.

Analysis of the association rule results provided identification of redundant, independent, and competing rules (described as below) for their removal. The following processes were applies to the rule set for the group-oriented question reduction:

- *Redundant rules:* The Apriori algorithm extracts the frequent rules and, therefore includes some

redundant rules those are same except one of the rules has additional antecedents. An example of this is rule 1 and rule 2 given in Table 2. These two rules have same consequent and same confidence scores, however, rule 2 requires an additional variable and is not necessary. Rule 1 is applied in the process of question reduction as it requires fewer criteria to meet and lesser computation time to implement the rule.

- *Independent rules:* Rules are generated that may be contradictory. The contrast values of an antecedent will lead to the same consequent, making the antecedent and consequent variables independent of each other. An example of this is rule 3 and rule 4. For example, in the derived from rules 3 and 4, the attribute *heavy_vehicles* will equal "No" regardless of the value of *Equipment_mechnical*. It can be concluded that *Heavy_vehicles* is independent of *Equipment_mechnical*. These two rules should not be applied in the reduction of the questions.
- *Competing rules:* The competing rules are the rules with same consequent but with different combinations of antecedents. An example of this is rule 5 and rule 6. These rules have different antecedents but same consequent. The selection of one rule is sufficient for use in question reduction. The rule with lower confidence is excluded from the process.

	Antecedent	Consequent	Conf
1	business_plans =Grow equipment_lease_purchase =No	heavy_vehicles =No	0.96
2	business_plans =Grow equipment_lease_purchase =No international=No	heavy_vehicles =No	0.96
3	equipment_mechnical =No	heavy_vehicles =No	0.95
4	equipment_mechnical =Yes	heavy_vehicles =No	0.9
5	legal_structure =SoleTrader Operational =StartUp	business_plans =Grow	0.91
6	Operational =StartUp equipment_mechnical =Yes	business_plans =Grow	0.9

Table 2: Example of rules

4.2.4 Question Reduction

Decision tree classification is only used for parting the branches and collecting the instances at leaf node. The question reduction for each leaf node is implemented by applying association rules on the instances of the leaf node and observing the distributions of the node. Once the meaningful association rule set is obtained, the question reduction process starts. Knowing that the antecedent of a rule leads to the consequent, when the antecedent condition appears in a customer questions path, the question in the form of consequent is not asked. The distribution of attributes for each leaf node is observed and the dominant values/responses which are selected by 95% and more customers are identified. The question which is associated with the attributes with dominant values can be saved from asking as the values that the customer most likely to choose can be known from the dataset observation.

5 Results Analysis

5.1 Finding the Optimal Number of Clusters in EM

The capability of calculating natural clusters in EM is valuable; however the clusters require interpretation. Our tests showed that the cluster outcomes were sensitive to the seed set in the EM configuration.

To evaluate the best combination of seed vs. cluster count, we used an additional validation method to the 10-fold cross-validation provided by EM, by splitting the instances into 66% for training and the remainder for testing. Results are shown in Table 3. In each case, an optimal seed value gives a combination of testing and training cluster counts that are equal, and we consider them to be optimal value. According to Table 3, the optimal number of clusters is 6 at seed 25. Although the training and testing cluster counts are equal at seed 10, the log likelihood is slightly higher at seed 25 than at seed 10.

Seed	Training Cluster	Testing Cluster	Log likelihood
100	6	3	-48.029
75	7	6	-51.554
50	3	5	-44.941
25	**6**	**6**	**-53.471**
10	3	3	-54.362

Table 3. Seeds and Clustering Results

As a result of clustering, each instance in the dataset is assigned a cluster label. Associating the instances with the cluster label, the distribution of clusters for each attribute can be observed. However, this does not lead to the understanding of the formation of clusters. Decision tree classification and the resulting rule sets would provide knowledge of the role of the attributes in determining the clusters, and add comprehensibility to the clusters.

5.2 Choose the Classification Algorithm with Best Performances

As discussed above, to provide understanding of the clustering process, post-clustering classification is introduced. Table 4 shows the accuracy of various classification methods using default settings with 10-fold cross-validation. J48 achieves the best accuracy results and thus it is used for the explanation on how the clusters are formed.

Classification Method	Accuracy
Decision Stump	90.2%
J48	**91.3%**
Random Forest	74.1%
Random Tree	66.3%

Table 4. Classification Accuracy

5.3 Decision Tree Pruning

Pruning is a technique that reduces the size of decision trees by removing branches of the tree that provide little power to classify instances. The objective of pruning is to reduce the complexity of the classifier, while ameliorating the accuracy by removing overfitting caused by noisy data.

The C4.5 and Reduced Error algorithms are available for pruning; within which options include the use of Laplace and Minimum Nodes. C4.5 pruning is based on statistical confidence estimates and allows all the available labelled data to be used for training (Witten, 2011). Reduced-error pruning separates data to training, testing and pruning set. When the error rate of testing set exceeds that of the training set, the pruning process starts (Witten, 2011). The minimum number of nodes is the number of nodes allowed for each leaf; the default being 2. Laplace smoothing is a technique used to smooth categorical data in order to prevent probabilities being calculated as zero (Kirchner, 2006).

Settings	Accuracy	Coverage	# Tree	#Leaves
Reduced error pruning	89.6%	95%	379	395
Default C4.5 pruning	91.3%	97%	37	28
C4.5 with Laplace pruning	90.7%	98%	37	28
C4.5 with 30 minimum nodes	**91.3%**	**97%**	**23**	**19**
C4.5 with 40 minimum nodes	87.3%	97%	17	14
C4.5 with 50 minimum nodes	86.5%	97%	14	12

Table 5. Pruning Results Comparisons

Table 5 presents the results with different settings. Applying reduced error pruning decreases the accuracy and increases the number of trees and leaves which will cause the overfitting problem. The reason leading to the poor performances of reduced error pruning may be caused by the limited number of training instances available as the dataset is split into 3 sets. The Laplace option improves the coverage; however, it decreases the accuracy. From the results, an increase in the number of minimum nodes option above 30 leads to the building of a more general tree but it also lowers the accuracy. When C4.5 algorithm with 30 minimum nodes is used, the performance is best and therefore this configuration is used for decision tree classification.

5.4 An Example of Group-oriented Question Reduction Process

Due to the space limitation, only small set of rules are displayed in Figure 2. "|" shows the depth of the tree. *Business_premises=BusinessPremises* is the root of the classification tree. According to Figure 2, if an instance has following values provided to the described attributes *Business_premise=BusinessPremises,Equipment_lease_purchase = No* and *Employees = Employees0*, *Portable_tools=No* and *Owners_partners=Directors2-4* then the instance belongs to cluster0. Let this path be identified as leaf node 1. If *Business_premises=BusinessPremises*, *Equipment_lease_purchase=No*,and *Employees=Employees5-9*, then the instance belongs to cluster 0 as well.

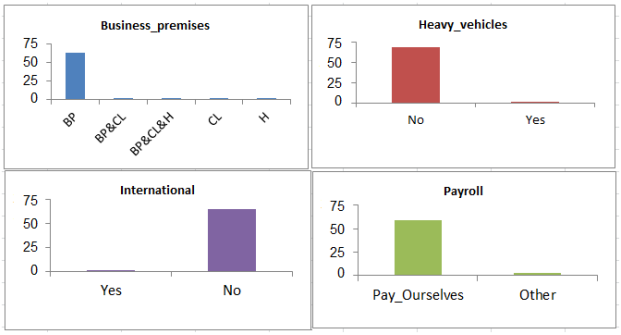
Taking leaf node 1 for illustrations on how the questions can be reduced, two steps are involved before the questions, which can be removed, are decided. Firstly, the distributions of attributes for a leaf node are reviewed. Figure 3 shows the interesting distributions of values under the left node 1. It is noticed that about 98% of times, instances falling under leaf node 1 include the following values for the attributes: *Business_premises=Business_premises*, *Heavy_vehicles=No*, *International=No*, and *Payroll=Pay_ourselves*. So these four questions should not be asked if an instance belongs to cluster 0. In this example, *Business_premises* is asked before the instances are deemed as cluster 0 members. Therefore, three questions which are *Heavy_vehicles*, *International*, and *Payroll*, can be removed from the analysis of attributes distributions.

Next, association rules are applied on leaf node 1 to further reduce the number of questions. From experiments conducted on leaf node 1 using association rules, 20 rules can be found which satisfies the condition of confidence>0.9 and support>0.3. Because there are redundant rules, independent rules, and competing rules, the number of rules which can be applied comes down to four. Figure 4 shows the four association rules.

Gathering all the required information of question reduction for users who belong to cluster 0, Figure 5 demonstrates how the questions can be reduced. Firstly, the classification rules are applied to decide the membership of cluster 0. *Business_premises*, *Employees*, *Portable_tools*, *Owner_partners*, and *Independent_contractors* are the questions that decide whether an instance belongs to cluster 0 or not. The following questions are then asked in order to apply the mined association rules. If the answers to the questions are of certain values, some questions do not need to be raised. For example, if *Owners_partners=Directors1* and *Equipment_lease_purchase=Yes*, then the question *Capital_expenditure* should not be asked as the answer can be known from the association rules. By following the flowchart in Figure 5, five questions can be saved from being asked. The five questions are on *Heavy_vehicle*, *International*, *Payroll*, *Capital_expenditure*, and *Operational*.

```
Business_premises = BusinessPremises
|  Equipment_lease_purchase = No
|  |  Employees = Employees5-9: cluster0
|  |  Employees = Employees0
|  |  |  Portable_ tools = No
|  |  |  |  Owners_partners = Directors2-4: cluster0
|  |  |  |  Owners_partners = DirectorsGT4: cluster2
```

Figure 2. Example of Classification Rules



BP-Business Premises, CL-Customer Location, H-Home

Figure 3. Predominate Attributes Distributions

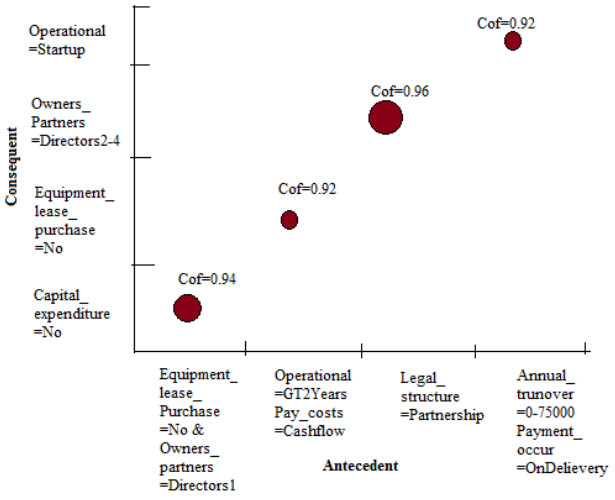


Figure 4. Association Rules

5.5 Comparisons of different techniques to reduce the number of questions

Two benchmarking methods including association rule mining and cluster distribution are applied to assess the effectiveness of the proposed method. In the first benchmarking method, association rule mining is applied to each individual cluster to obtain rules which can be applied for the reduction of the questions. Different from the proposed method, this approach mines the rules from the whole cluster, whereas the proposed method derives rules from each leaf node. In the second benchmarking method, each cluster is analysed to find the dominant distribution where 95% and above responses choose the same value to the question. By knowing the value that a customer is likely to choose for a question, the question can be saved from being asked. The distribution analysis is conducted on each cluster, whereas the proposed method observes the distribution on each leaf node.

Results in Table 6 show the percentage of questions that can be reduced for each cluster. The proposed method reduces the questions for each leaf node and a cluster can have multiple leaves. We produce the results in two ways: (1) by using the "best" results for each cluster; and (2) by calculating the "average" number of questions for each cluster.

In general, the proposed approach is able to outperform *AR* and *Distribution methods*. The proposed method applies the question reduction strategies on the instanced belonging to the leaf of the decision tree where instances have more homogeneous values than the instances falling under a cluster. From the results, it can also be seen that clusters 2, 4, and 5 perform better than clusters 0, 1, and 3. A reason is that the better performing clusters hold smaller sized collections of instances than the worse performing clusters. The instances in clusters with smaller size are more similar than the instances that fall under big clusters showing heterogonous distributions.

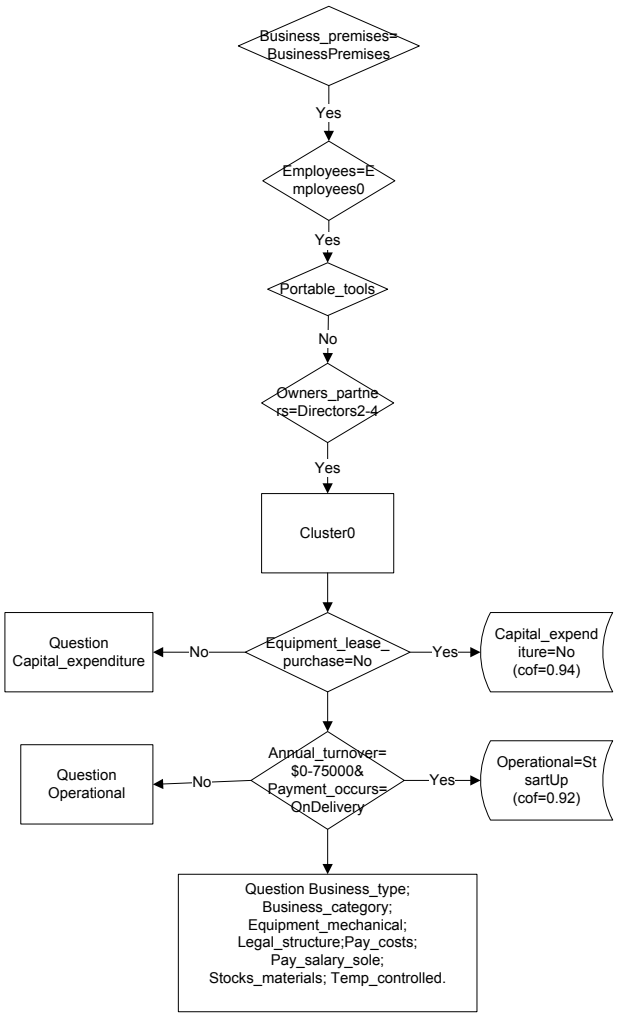


Figure 5. An example of Question Reduction

Cluster No	0	1	2	3	4	5
Association Rule (AR)	5.7%	2.8%	5.7%	5.7%	8.5%	8.5%
Distribution	5.7%	11.4%	8.5%	8.5%	20%	17.1%
Proposed Approach (Best Scenario)	17%	20%	40%	17.1%	34%	22.8%
Proposed Approach (Average)	14%	14%	28.5%	8.5%	25.7%	22.8%

Table 6. Percentage of Questions Reduced by Using Different Techniques

5.6 Quality of Question Reduction

In order to test the quality of question reduction, the dataset is split into a training dataset and testing dataset. For each cluster, 10% of the data is taken to form the

testing dataset. Based on the customers' responses to questions, cluster label and leaf node are assigned to each instance in the testing data. From the training dataset, questions which can be reduced for each leaf node are known. Therefore, these questions that can be reduced are not asked in the testing dataset. Based on the mined association rules and data distribution statistics, the accuracy of answer prediction test is conducted. The accuracy scores are calculated by checking the degree of agreement between the customer's answers to the questions which are not asked due to the question reduction and the derived answers based on the mined association rules and data statistics for each leaf node.

For comparison purposes, the distribution only question reduction is performed on each cluster alone, as well as association rule only. As shown in Figure 6, the accuracy of answer prediction of the proposed method ranges from 95% to 100%. The results indicate that the proposed method usually performs the best. The reason lies in that the proposed method applies question reduction on leaves, while the distribution only method and association rule method only conduct the question reduction on clusters which have more instances and distributions that are more heterogeneous. In general, cluster 2 and cluster 4 achieve better results than the other clusters. This is because the size of cluster 2 and cluster 4 are much smaller than the other clusters.

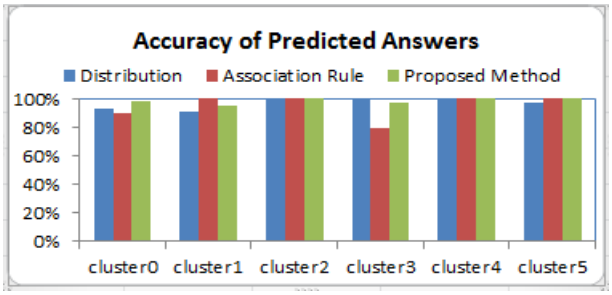


Figure 6. Accuracy of Predicted Answers.

6 Conclusion

This study provides an effective way to reduce the number of questions to be used in recommender websites to prevent customer burnout and loss. The questions are constructed by domain experts and each question is closely tied to a service and/or product to be offered. Based on the past customer behaviour on the website, it groups the usage patterns and finds associations among them to predict what should be the next question for the incoming customer. Feature selection is a widely used technique to reduce attributes in the datasets for mining useful information. The attributes with the least correlations with the target are removed. However, the recommended attributes generated from feature selection for removal are not satisfactory for the task in this paper.

The proposed method discusses a possible way to reduce the questions by analysing the customers' usage data and finding association rules among instances that belong to each cluster, in which characteristics are successively identified by decision tree. The usage data, detailing how customers are selecting the questions and getting the products recommended, is used in the analysis. Instead of mining the rules using the whole dataset, the dataset is grouped into clusters based on 'alike' users. The rules from the decision tree classification enable an explanation of how the clustering is formed. A leaf in the decision tree explaining the cluster rule demonstrates high homogeneity where some uniform distributions can be identified. Association rules are mined from each leaf. By knowing the values in the antecedent, the consequent question should not be asked because the value/answer to the consequent question can be derived from the mined association rules with high support and confidence.

The proposed method has been tested on a live website data. Results show that the combination of two techniques (association rules and values distributions of leaf) produces the best question reduction over that of the results achieved by employing association mining only or the results achieved from the combined techniques being applied to the whole cluster (without decision tree mining).

Acknowledgement

The content presented in this paper is part of an ongoing cooperative study between Queensland University of Technology (QUT) 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 for analysis. The views presented in this paper are of the authors and not necessarily the views of the organizations.

7 References

Burke, R. (2000) Knowledge-based Recommender Systems. Encyclopedia of library and information.

Felfernig, A., Burke, R. (2008): Constraint-based Recommender Systems: Technologies and Research Issues. Conference on Electronic Commerce.

Han, J. et al., (2011) Data Mining Concepts and Techniques. ISBN: 978-0-12-381479-1. Elsevier.

Janecek, A., et al., (2008) On the Relationship Between Feature Selection and Classification Accuracy. JMLR (4). 90-105.

Lin, C., Hong, W., Chen, Y., Dong, Y. (2010): Application of salesman-like recommendation system in 3G mobile phone online shopping decision support. Journal of Expert Systems. 37(2010): 8065-8078.

Nakash, R. A., et al., (2006) Maximizing response to postal questionnaires. BMC Medical Research Methodology 6(5). DOI: 10.1186/1471-2288-6-5.

Rokach, L., Shani, G., Shapira, B., Siboni, G. (2013): Recommending Insurance Riders. SAC, 253-260.

Witten, I. et al., (2011) Data Mining Practical Machine Learning Tools and Techniques. ISBN 978-0-12-374856-0. Elsevier

Zanker, M. (2008) A collaborative Constraint-based Meta-level Recommender. RecSys'08. 139-145

An investigation on window size selection for human activity recognition

Anthony Blond **Wei Liu** **Rachel Cardell-Oliver**

School of Computer Science & Software Engineering,
University of Western Australia,
35 Stirling Highway, Crawley (Perth), Western Australia 6009,
Email: {anthony.blond,wei.liu,rachel.cardell-oliver}@uwa.edu.au

Abstract

Low power sensors installed in a smart house can collect data that can be mined using activity recognition techniques, allowing the actions of the resident to be inferred. This is a key step towards the dream of smart houses assisting or enhancing everyday living. Human activity recognition can be formulated as a classification problem; given a window of time, and data from sensors within that window, should it be classified as "preparing dinner", "going to bed" or "no activity"? Efficient machine learning techniques have been applied to this problem, but the *discretisation* data preparation step , also known as feature definition, selection, encoding or segmentation, despite its critical impact on the quality of the generated dataset, has received inadequate attention. In this paper, we investigate this fundamental problem of how to best discretise raw sensor data from human activities to create the feature vectors for classifier training, focusing on the effect of window length on classifier performance. We contribute a probabilistic model for characterising the association between window length and detection rate and introduce a modular architecture for performing experiments using different discretisation techniques. The performance of a selected Naïve Bayes classifier at different window lengths has been evaluated to demonstrate the importance of window selection.

Keywords: Activity recognition, Sliding window, Data mining, Discretisation

1 Introduction

The dream of smart houses assisting or enhancing everyday living is gradually becoming a reality, thanks to increasingly available embedded devices for collecting data and the data mining algorithms for understanding human activity data.

Sensors installed in a smart house can collect simple data that indicates for instance whether a particular cupboard was opened or closed or the oven was turned on. Despite such simplicity in data, patterns of sensor activations can be used to recognise useful behaviours such as preparing dinner or going to bed, as well as long term observable habits. Techniques for mining smart house sensor data to reveal the associations of activities and patterns of sensor activations constitute the research area of *activity recognition*. Such intelligence is vital to keep a smart house system informed to provide personalised services to either improve quality of life by automatically actuating devices, or provide critical care and monitoring data while offering home comfort. It is worth noting that an alternative for obtaining behavioural data is the use of a small number of specialised sensors such as wearable or body sensors. To differentiate from body sensing techniques, Wang et al. (2007) describe the sensing from the living environment as *dense sensing*. In this research, we are focusing on activity recognition from data obtained through dense sensing.

Activity recognition can be formulated as a classification problem for known activities of interest. For example, given a window of time, and data from sensors within that window, should it be classified as "preparing dinner", "going to bed" or "no activity"? Supervised learning algorithms for classification have been applied to this problem with some success. The algorithm learns what sensor activations are present from a training set of typical examples for an activity, along with other information such as their order and duration, resulting in a classifier. Significant research efforts have been focusing on refining the learning algorithms to improve the performance of the classifiers, with little or no systematic study of the more fundamental issues on how to discretise given sequences of sensor activation data in the first place (Krishnan & Cook 2012).

Ultimately the purpose of activity recognition is to detect the occurrence of an activity either as it happens, or retrospectively from historical data. In either case, the period of time the sensor data is captured in is not known in advance. This is a defining characteristic distinguishing this problem from traditional data mining when data are often readily defined as discrete 'feature vectors'. In data mining, data sets are collections of discrete instances, each comprising a set of features. Classification involves mapping the values of features in an instance to a single class or label, chosen from a finite set. However, when applying classification algorithms for activity recognition, converting the raw sensor data into the required input format for the classification algorithms is a non-trivial task. Take for instance the example activity of "Preparing lunch" as illustrated in Figure 1, lets consider the difficulty one would encounter when converting the raw sensor activation data to the tabular input instances of attribute values required by a classification algorithm. How long should be the observation window? What criteria should be used to determine the value of an attribute, i.e. the encoding process, a change in sensor values or a complete sensor activation and deactivation cycle? Technically speaking, the data is sequential in nature and must first be *discretised* (i.e.

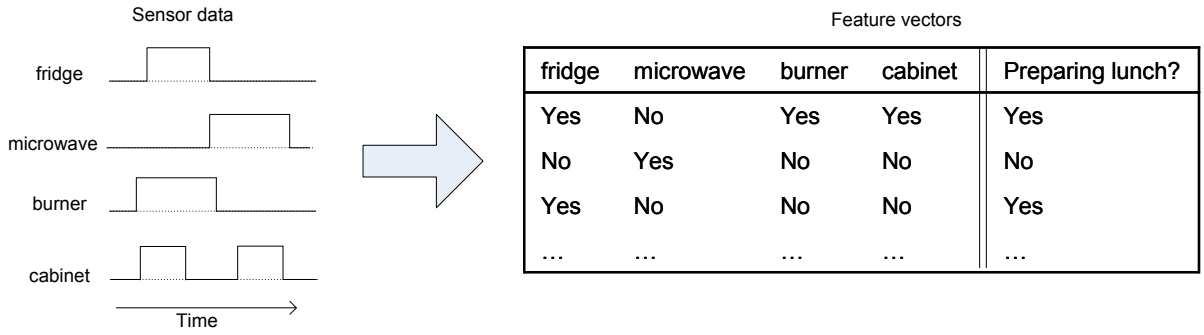


Figure 1: Illustration of the Process of Discretisation.

semantically processed) into feature vectors, before being passed to a classification algorithm (Galushka et al. 2006). In terms of observation window alone, possible discretisations include dividing the data into evenly spaced slices (van Kasteren et al. 2010*a*), moving a fixed length sliding window over the data resulting in overlapping instances (Logan et al. 2007, Stikic et al. 2008) and dividing the data according to activity labels (Tapia et al. 2004). However, none of the existing research has systematically investigated the effects of such choices over the classification results.

In this research, we begin a quest on the more fundamental problem of how to best discretise the raw sensor data of human activities to create the feature vectors for classifier training. The question, however, encompasses an array of subproblems such as selecting the length and offset of sliding windows, labelling windows and encoding features. This paper focuses on the effect of window length and sliding offset on classifier performance, and the main contributions are as follows.

- A probabilistic model for characterising association between window length and classifier performance;
- A modular architecture for performing experiments using different discretisation approaches;
- Experimental results, obtained using this system, examining the effect of window length on the performance of a selected classifier (Naïve Bayes).

The rest of this paper is structured as follows. Section 2 describes the background and related work in the field of human activity recognition in smart homes. Section 3 describes the discretisation problem and is followed by a model of the effect of window length and offset on the labelling of intervals in Section 4. Section 5 introduces our modular system architecture enabling the investigation of human activity recognition at different stages. In Section 6 preliminary results of an experiment measuring the effect of window length on a Naïve Bayesian classifier are presented, as compared to the predicted results using the model in Section 4, followed by concluding remarks and future work in Section 7.

2 Background and related work

In the field of activity recognition in smart homes the specific problem addressed is associating patterns in data generated by a number of sensors with discrete tasks performed by humans.

Tapia et al. (2004) present the results of an experiment applying Naïve Bayesian classifiers to the data from a large number of simple state-change sensors (a.k.a contact sensors). The labelling of the ground truth is done using a PDA carried by the resident, supplemented with manual annotation of clusters of sensor readings. The discretisation process used for generating training sets makes use of the start and end times of known activities. For generating test sets, a sliding window with a length equal to the mean duration of the activity in question is used, but without taking into account the variance of the duration. Three measures are used to assess the performance: the percentage of time an activity is correctly detected within the known duration of the activity, whether the activity was detected in the "best interval" shortly after the known activity ends and whether the activity was detected at least once during the known activity. Evaluation is performed using a *leave one day out* cross validation approach. They have found that temporal ordering does not improve the classification performance (at least with the relatively small set of data) and that the "best interval" method has the highest detection accuracy.

Logan et al. (2007) describe an experiment with the goal of analysing behavioural data of residents in a natural environment. Data from a range of sensors including wireless object usage sensors similar to those used in the work of Tapia et al. (2004), accelerometer body sensors and RFID tags on objects is annotated manually using video footage. Both Naïve Bayesian and C4.5 decision tree classifiers are trained and evaluated, using leave one day out cross validation. The discretisation process was different than that used by Tapia et al. (2004), with the data being split into 30 second windows with 50% overlap, for both the training and test sets. In this case the activities were found to have an average duration of 1 minute. Decision tree classifiers were found to have the best performance. Again, the leave one day out cross validation approach is used because it was found that conventional 10-fold cross validation resulted in over-fitting.

The work of Tapia et al. (2004) and Logan et al. (2007) demonstrate that it is possible to infer human activities from the interactions with the objects they use. Similarly, the Proact system by Philipose et al. (2004) also demonstrates the same research approach.

One alternative to dense sensing is the use of wearable sensors. This is used to some degree by Logan et al. (2007), but it is combined with other data. Stikic et al. (2008) focus on the wearable sensor subset of the work of Logan et al. (2007) and investigate the feasibility of semi-supervised learning approaches. Semi-supervised learning involves using a small set of labeled training instances to label unknown instances, which are then fed back into the training set in an iter-

ative fashion. Other experiments involving wearable motion sensors include identifying physical activities such as walking or sitting (Lester et al. 2006, Bao & Intille 2004) or location (Lee & Mase 2002).

Other methods exist that do not require the same discretisation process, such as hidden Markov models (Patterson et al. 2005) and conditional random fields (van Kasteren et al. 2010*a*), where transitions from one activity to the next are modelled. Because of this, the windows used tend to be significantly shorter than the length of the activities being identified, otherwise the relationships between intervals would not be captured during the activity. These approaches are outside the scope of this paper, which deals with the relationship between the sensor events and the windows they are in, but we plan to incorporate these other models in future work.

3 The discretisation problem

The first requirement of supervised learning is to have available a *ground truth*, with which to train the classifiers and later evaluate them against. What this generally means is each feature vector has an activity *label* attached to it, assumed to be correct with 100% confidence. In the case of temporal sensor data, the raw ground truth is annotated time intervals within which sensor measurements fall into. Extracting feature vectors therefore requires making a choice on how to interpret this information.

In order to make use of the existing ground truth, it would be tempting to simply create a feature vector for every known activity interval and use sensor data in the intervals as features. Applying a label to each feature vector for training and testing is then a simple one-to-one mapping.

The reason this approach cannot be used to evaluate classifiers in activity recognition is that it is not representative of the data available when the classifier is deployed in the real world (Krishnan & Cook 2012); here there is no ground truth and therefore no clearly defined intervals to form feature vectors from. The recognition system must decide the start and the end of an activity. A typical approach is to use sliding windows on the incoming data, creating feature vectors using data from an interval defined by a certain window length, and shifting the window by some offset in time (or waiting for a buffer to fill if processing real-time data).

This same approach can be used to create the feature vectors for training and testing classifiers. The problem is that there is no longer a simple one-to-one mapping from the raw ground truth to the labels on the feature vectors, except for a few select windows that happen to intersect perfectly with the known activities. A decision must be made on how to map the ground truth annotations to the labels for the feature vectors. How the 'correct' label is chosen and assigned to an interval depends on the goal of the system, which can be divided into three broad categories:

Interval *matches* activity: Here the goal of the fielded system is to identify intervals that best represent the activity. In this case the label may be set to the activity whose start and end points match the interval's perfectly, match approximately or overlap by a threshold percentage. This situation is important if the identification of the start and end times of an activity is desired, or false repeats are unwanted. For example, the activity in question may be used to detect a higher-level one (Huỳnh et al. 2007);

Interval has activity as it's *context*: The goal in this situation is to decide whether, given an interval of time, the activity is currently occurring. Here the label can be set to an activity if it was ongoing over the course of the interval. This may be desirable in a smart home where music is to be played while the resident is cooking, where its only important that part of the activity is detected.

Interval *contains* activity: In this case, the goal is to identify intervals where the activity occurred at some point between its start and end. Therefore the label may be set to any activity that occurred during the interval, or occurred a number of times. This approach is desirable when the events that identify the activity do not occur while it is ongoing. For example if a "sleeping" activity is annotated only when the subject is actually in bed then, without a sensor in a bed, the only way to identify if a subject is sleeping may be noting the bedroom light is turned on, the door closed and the light turned off. These events would not be encompassed by the actual activity, so the context and activity matching labelling techniques would not be effective;

In the rest of this paper we focus on the *contains* approach, which is of interest because it allows for the fact that the defining sensor activations for a particular activity may precede or proceed the activity itself. In future work we explore the differences between these approaches in further detail.

3.1 Effect of window length and offset

The problem then becomes how to select the window structure (in terms of length and offset between them) to maximise the performance of the classifier using the discretised version of the data. To do this the following constraints must hold:

1. Every instance of an activity is in at least one window;
2. The offset between windows is as large as possible to reduce the number of training windows needed;
3. Windows are as large as possible to reduce the number of training windows needed;
4. Windows are not too large, to ensure sensor data from other activities do not behave as noise in the wrong windows.

With this in mind we are interested in the number of windows that each occurrence (an *instance*) of an activity is likely to fall into, and therefore label.

4 A probabilistic model for classifier window length

4.1 Single activity instance

We begin by defining a single instance of an activity $A_j \in A$ as an interval with start time $s_j \geq 0$, measured from time $t = 0$, and a length $l_j > 0$:

$$A_j = [s_j, s_j + l_j) \tag{1}$$

Referring to Figure 2, we interpret a sliding window mechanism as placing a number of overlapping

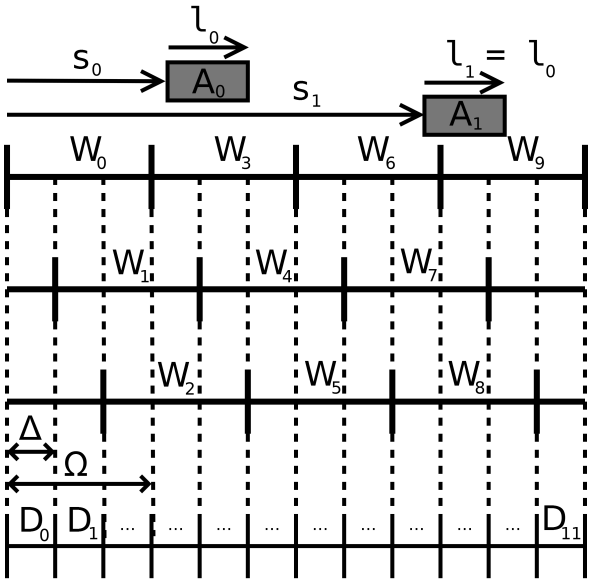


Figure 2: An example of windows with overlap. Here the window offset is $\Delta = \frac{\Omega}{3}$ and there are two activity instances of equal length. Activity instance A_0 falls into windows W_2 and W_3, while A_1 falls into window W_8.

windows W_i of length Ω at offsets of Δ. For simplicity, and without loss of generality, we assume[1] Δ is an integer divisor of Ω, that is:

$$\Omega \ \% \ \Delta = 0 \tag{2}$$

$$W_i = [i\Delta, i\Delta + \Omega) \tag{3}$$

We then note the intervals between the start times of each window, in terms of Δ:

$$D_i = [i\Delta, (i+1)\Delta) \tag{4}$$

We now consider the relationship between the location of an activity instance A_j and the determination of whether a window W_i will be assigned the corresponding ground truth label in the training and test sets. This varies between applications (Section 3), but the most common (and the one we consider in this paper) is that A_j starts and ends within its bounds ($A_j \subseteq W_i$):

$$\begin{aligned} label(W_i, A_j) \equiv \ & s_j \in [i\Delta, i\Delta + \Omega) \ \wedge \\ & (s_j + l_j) \in [i\Delta, i\Delta + \Omega) \end{aligned} \tag{5}$$

Given the above definitions, we are interested in finding the number of windows A_j will cause to be labelled A. In Figure 2, A_0 and A_1 both have the same length, but while A_0 starts and ends in two windows, thus satisfying (5), A_1 is only completely captured by a single window. It is clear that an activity instance can only satisfy (5) for a limited number of windows and that these will be consecutive. A given activity instance will fall into windows $\{W_{i-n+1}, ..., W_{i+n-1}\}$, where $n = \frac{\Omega}{\Delta}$. If we consider a single activity instance starting in D_i, then the whole activity will fall into the windows $\{W_{i-n+1}, ..., W_i\}$. In this case, for A_j to cause $1 < k \leq n$ windows to be labelled as A the following must hold:

$$s_j \in D_i \ \wedge \ (s_j + l_j) \in D_{(i+n-k)} \tag{6}$$

[1]Without this assumption, the condition in (6) would be necessary but not sufficient because each interval D_i would no longer be covered by the same number of windows.

4.2 Introducing probability

We model the starting time and length of an activity as random variables S and L, with distributions $f_S(s)$ and $f_L(l)$ respectively. The probability of a single instance of an activity $A_j \in A$ being falling completely within k windows $\{W_{i-k+1}, ..., W_i\}$ is the probability that it starts in D_i and ends in $D_{(i+n-k)}$, which (assuming S and L are independent[2]) is

$$P(K = k) = \sum_{s=i\Delta}^{(i+1)\Delta-1} P((S+L) \in D_{(i+n-k)}) P(S = s) \tag{7}$$

(where i can be determined by (5))

Expanding the definitions of D and defining the cumulative distribution function of L as $F_L(l)$, we can write this as:

$$\begin{aligned} P(K = k) = \sum_{s=i\Delta}^{(i+1)\Delta-1} & [F_L((i+n-k+1)\Delta - s) - \\ & F_L((i+n-k)\Delta - s)] f_S(s) \end{aligned} \tag{8}$$

If m is the total number of windows, to obtain the probability that a given activity occurrence A_j will have membership in $1 < k \leq n$ windows (causing them to be labelled as A), noting only consecutive windows will be valid, we sum over all windows:

$$\begin{aligned} P(K = k) = \sum_{i=0}^{m} \sum_{s=i\Delta}^{(i+1)\Delta-1} & [F_L((i+n-k+1)\Delta - s) - \\ & F_L((i+n-k)\Delta - s)] f_S(s) \end{aligned} \tag{9}$$

The above general formula can be simplified by making some assumptions about the distribution of activities. Although it is not reasonable to assume they are uniformly distributed across the whole data set, it is reasonable to assume that on a small scale, relative to the window length, they are reasonably uniform. Since the windows are repeating, in this case we do not need to sum over all sub intervals and (9) becomes simply:

$$\begin{aligned} P(K = k) = \sum_{s=0}^{\Delta-1} & [F_L((n-k+1)\Delta - s) - \\ & F_L((n-k)\Delta - s)] \end{aligned} \tag{10}$$

4.3 Predictions

We predict that if these windowing structures are used for both the training and testing of HAR classifiers, then the use of traditional recall and precision evaluation measures will result in the following observations:

1. Recall will improve as the probability of an activity labelling no windows ($P(K = 0)$) falls;
2. Precision will improve as the probability of an activity labelling a single window ($P(K = 1)$) increases;

[2]The length of activities is not independent of the starting times when looking across an entire day of data. However, our focus is on single-vector classification, so we are only interested in intervals of time that are significantly shorter than a day. Therefore, for our purposes, it is reasonable to approximate the starting time and length as being independent.

3. Precision will decrease as the probability of an activity labelling more than one window ($P(K > 1)$) increases.

Without specifying the relative importance of precision and recall, it is not possible to define a single optimal point.

5 A modular architecture for discretisation

In order to investigate how to best represent the information in the time window, we have divided the problem into 6 explicit steps, or modules: interval generation, activity event membership, interval labelling, interval grouping, sensor event interval membership and feature encoding. We use the term *interval* to denote an object storing a start and end time with attached information, and the term *feature vectors* to denote the final output used for machine learning. Figure 3 shows the layout of our implementation of these modules written for use in Octave or MATLAB, which produces ARFF files suitable for machine learning using the WEKA software package (Hall et al. 2009).

Each module has a number of different functions implementing the required interface that can be used to process the data at each step. The whole discretisation process can be defined as a chain of these functions, one for each module, along with parameters for each (the chain used in our experiment is described in Section 6). The training and test sets resulting from each combination are stored in a database for later evaluation.

5.1 Interval/window generation module

Typically, the sensor data and activity data are available in the form of start and duration pairs for each activation or annotation, respectively. The purpose of the interval generation module is to take this raw data, which can can be interpreted on a time line as a sequence of (possibly overlapping) intervals, and generate the time divisions that will define what information gets captured by the final feature vectors. There are two broad choices of how to generate intervals, using a repeating set of windows or defining boundaries using known data.

5.1.1 Using a repeating set of windows

The simplest technique in this case is to use "time-slicing", where non-overlapping intervals of length Ω are used. If time is measured from 0 and the maximum recorded time is t_{max}, then this results in $\lfloor \frac{t_{max}}{\Omega} \rfloor$ intervals. Extending this to the overlapping case is equivalent to using a sliding window; the window has a length Ω and is shifted by increments of $\Delta = \frac{\Omega}{n}, n \in \mathbb{Z}$, resulting in $\lfloor \frac{t_{max}}{\Delta} \rfloor$ intervals.

5.1.2 Defining boundaries using known data

It is possible to define the intervals using known data such as the activity annotations. This would result in feature vectors that match perfectly with each instance of an activity. This is, however, only useful for the training set; if used for the test sets, they would not realistically represent information known to a classifier, since the duration would be included implicitly.

A combination of these two approaches, which is valid for the test set, is to use a sliding window with the length determined by the known activities. This is equivalent to the approach used by Tapia et al. (2004). The choice then is what values to use for Ω and Δ, which is investigated in the experiment in Section 6.

5.2 Activity event interval membership and labelling modules

Once the intervals have been generated, they are passed to the activity interval membership module. This assigns each instance of every activity a membership to every interval. The output is then sent to the labelling module, which is responsible for attaching an activity (or class) label to each one. At this point the intervals for the training and test sets may be different owing to different interval generation functions, but will both use the same membership and labelling functions. The choice in activity membership and labelling functions is determined by the desired *ground truth* definition for the data and what the purpose of the classifiers used on the data will be, as discussed in Section 3.

5.3 Grouping module

Depending on the final evaluation technique used, it may be desirable to divide the labelled intervals into different groups for consistency and folding purposes. For a discussion of why ignoring natural groupings of intervals and simply using conventional cross validation techniques can lead to over-fitting, see the work of Logan et al. (2007). The most obvious choice for smart home data is to group the intervals into separate days, which will make the use of leave-one-day-out cross validation straight forward.

5.4 Sensor event interval membership and encoding module

Selecting which features of each interval to use is the next decision. Like the process of interval labelling, this is a two-stage process. First every sensor event is assigned membership to every interval, using the same possible set of functions as for activities. Then, this output is combined with the original sensor data in the encoding module to create the features to represent each interval. Every interval generates a feature vector.

The simplest feature to use is just a measurement of whether a particular sensor was activated at all during an interval. Other similar encoding functions may include using an overlap threshold, using the actual overlap as a real valued feature or recording whether the sensor value changed, as done by van Kasteren et al. (2010*b*). Using only a single attribute in this way causes the temporal ordering of the events to be lost, so another option would be to create a feature for every pairing of events, to represent the ordering, as reported by Tapia et al. (2004).

There are a large number of possible encoding functions. Separating this step from the others assists with experiments investigating which functions are more suitable in different situations being conducted.

5.5 Pre-cleaning and post-cleaning modules

The pre-cleaning module can be used for deleting unwanted sensor and activity data before it is passed to the rest of the discretisation modules. The post-cleaning module operates on the feature vectors themselves.

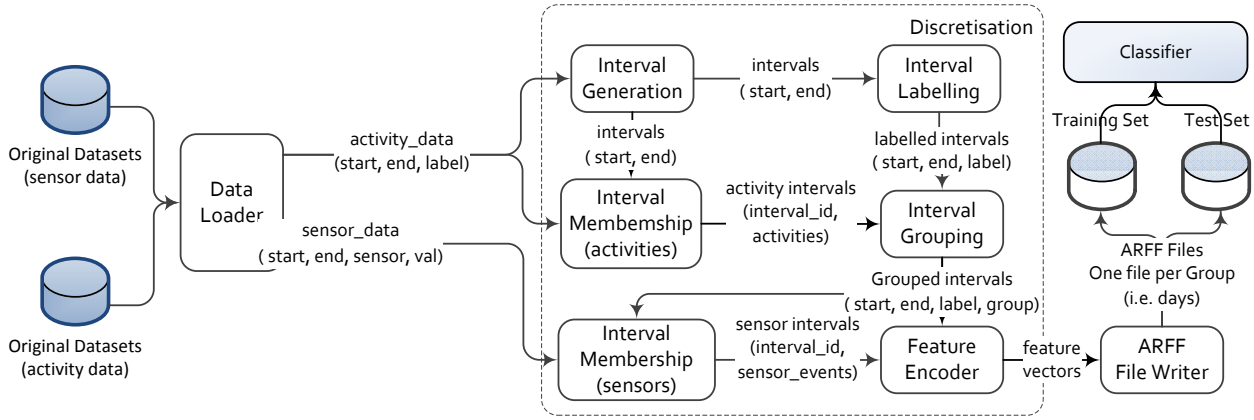


Figure 3: A modular architecture for discretisation in feature based activity

6 Experiment and results

To investigate the effect of choices on the interval generation step, we measure the performance of a classifier using varying window lengths on real human activity and sensor data. The following chain of discretisation functions is used:

Interval generation: We used a sliding window with a constant 50% overlap between intervals, while varying the window length;

Activity event membership and labelling: We associate an interval with an activity if an instance of that activity fell completely inside it, at least once;

Grouping: Since leave one day out cross validation is be used for evaluation, we simply divide the intervals into separate days;

Sensor event membership and encoding: A binary feature was used for every available sensor, measuring whether it was activated at all during the corresponding interval;

Pre-cleaning: No pre-cleaning of the data was performed;

Post-cleaning: No post-cleaning of the data was performed.

6.1 Description of dataset

We use the 'subject1' dataset collected by Tapia et al. (2004), discussed in Section 2. Data of 14 days from 77 state-changes sensors is generated from the 30-year-old resident performing normal day to day activities. Activities are annotated by the resident in real-time, by selecting from a list of 35 possible activities adapted from the work of Szalai (1973). This was found to have many omissions, and was therefore supplemented by indirect observations over the sensor data by an expert assisted by the subject, prompted by photos of sensors. This may have introduced a degree of bias (Logan et al. 2007).

6.2 Measurement of performance

A common technique employed to evaluate a learning algorithm is N-fold cross validation, which allocates an equal number of vectors into N "folds" of data and trains a classifier using $N-1$ of them, testing it on the remaining. This process is repeated N times, maximising the use of available data, while minimising over-fitting (Witten & Frank 2005). A related technique exists called "leave one day out" cross validation, where the folds are the days of data and the performance metrics are averaged (Tapia et al. 2004, Logan et al. 2007, Stikic et al. 2008). We extended WEKA to support the leave one day out validation approach.

The dataset contained a large number of negatives, so measuring only accuracy is not informative; the true-negative rate would be very high and result in a overly optimistic assessment of the performance. For this reason, we measured the precision and recall as the window length was varied.

6.3 Results and discussion

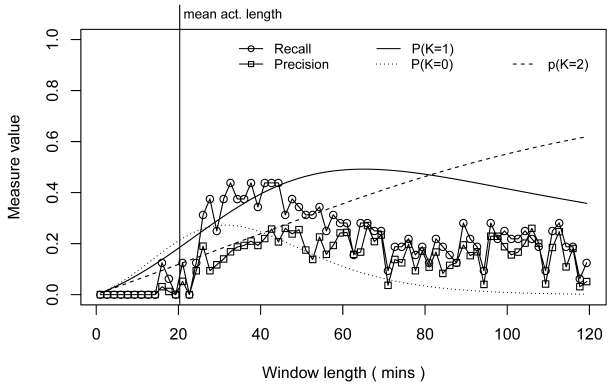


Figure 4: Performance of classifiers for varying window length (at increments of 100 sec) for **"Preparing dinner"** activity, compared to modelled probabilities of activity falling into k windows. The model used a normal distribution with mean of 1220 seconds and standard deviation of 1108 seconds.

For brevity, only the results of three activities are shown. Figures 4, 5 and 6 show classifier performance (measured separately as precision and recall) for the "Preparing dinner", "Toileting" and "Grooming" activities, respectively. We compare the performance to the modelled probability of that activity falling into k windows. The parameters for the distribution of each activity's duration, F_L in equation (10), were calculated using the original activity annotations in the dataset.

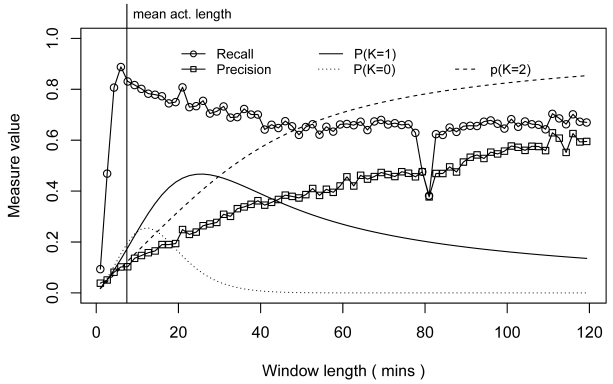


Figure 5: Performance of classifiers for varying window length (at increments of 100 sec) for **"Toileting"** activity, compared to modelled probabilities of activity falling into k windows. The model used a normal distribution with mean of 447 seconds and standard deviation of 455 seconds.

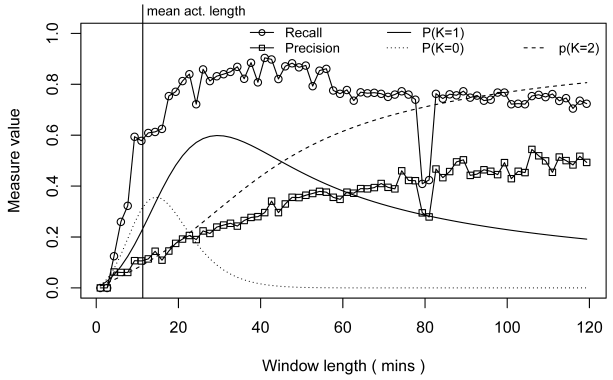


Figure 6: Performance of classifiers for varying window length (at increments of 100 sec) for **"Grooming"** activity, compared to modelled probabilities of activity falling into k windows. The model used a normal distribution with mean of 679 and standard deviation of 404 seconds.

As expected, using a window length set to the mean activity length did not produce the best results. In the case of the "Toileting" activity, recall was maximised at this point, but precision continued to improve as the window length was increased.

Referring to the predictions in Section 4.3, we can make the preliminary observations that recall does improve as $P(K = 0)$ falls and that precision improves with an increasing $P(K = 1)$. The final prediction of precision decreasing as $P(K > 1)$ only holds for the "Preparing dinner" activity. Overall the relationships seem tighter with the longer dinner activity. More generally we can observe:

- Precision and recall do vary significantly with changes in window length;
- A relationship between $P(K = k)$ and recall is observed for both "Preparing dinner" and "Grooming" activities. Recall does follow $P(K = 1)$ to a degree (particularly visible in the "Grooming" activity), and the relationship merits further study;
- The precision of both "Toileting" and "Grooming" activities seem to favour longer windows.

Our results do indicate that when choosing the window length for a particular activity, it is not always effective to set it to the average length of that activity, and the deviation should be taken into account. However, the benefit of increasing it further is not clear-cut. The context of the activity becomes important; what activities occur at nearby times and whether they share sensors.

7 Conclusion and future work

In this paper, we presented a preliminary investigation into the effect of window length when using data mining techniques for activity recognition. We contributed a model for the labelling of windows by activities with known probability distribution. We also presented a modular architecture for testing the effectiveness of different feature vector representations, and measured performance when varying window length. Although the classifiers performed relatively poorly (our focus was on the discretisation stage rather than the classifiers themselves), results indicate that choices made regarding the window length used for the discretisation of sensor data into feature vectors needs careful consideration.

In the future we plan to extend this work to include different distributions for the activities and include varying values for the window offset.

Later work will involve making use of the modular system to compare different discretisation chains. For the sake of simplicity we have only used the single attribute for each event, but plan to include temporal ordering in the future, including temporal distance measurements. Although existing work with the same dataset found that this did not increase the discriminatory power of trained classifiers, it was also suggested that the cause may have been the small size of the data set (Tapia et al. 2004). This further work is particularly desirable as the evaluation process used by Tapia et al. (2004) only measures classifier accuracy and does not take into account false positives, so a comparison may prove to be useful.

The design makes it possible to incorporate different data sets for comparison, so this is an avenue that will be explored, as the current dataset has a limited number of positive examples for each activity, and the significance of this is not immediately clear.

References

Bao, L. & Intille, S. S. (2004), Activity recognition from user-annotated acceleration data, *in* 'Pervasive Computing', Vol. 3001/2004 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 1–17.

Galushka, M., Patterson, D. W. & Rooney, N. (2006), Temporal data mining for smart homes, *in* J. C. Augusto & C. D. Nugent, eds, 'Designing Smart Homes', Vol. 4008 of *Lecture Notes in Computer Science*, Springer, pp. 85–108.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. & Witten, I. H. (2009), 'The WEKA data mining software: an update', *ACM SIGKDD Explorations Newsletter* **11**(1), 10–18.

Huỳnh, T., Blanke, U. & Schiele, B. (2007), 'Scalable recognition of daily activities with wearable sensors', *Lecture Notes in Computer Science* **4718**, 50–67.

van Kasteren, T., Englebienne, G. & Kröse, B. (2010*a*), 'An activity monitoring system for elderly care using generative and discriminative models', *Personal and Ubiquitous Computing* **14**, 489–498. 10.1007/s00779-009-0277-9.

van Kasteren, T., Englebienne, G. & Kröse, B. (2010*b*), Human activity recognition from wireless sensor network data: Benchmark and software, *in* L. Chen, C. Nugent, J. Biswas & J. Hoey, eds, 'Activity Recognition in Pervasive Intelligent Environments', Atlantis Ambient and Pervasive Intelligence, Atlantis Press.

Krishnan, N. C. & Cook, D. J. (2012), 'Activity recognition on streaming sensor data', *Pervasive and Mobile Computing* **In Press**.

Lee, S.-W. & Mase, K. (2002), 'Activity and location recognition using wearable sensors', *IEEE Pervasive Computing* **1**, 24–32.

Lester, J., Choudhury, T. & Borriello, G. (2006), A practical approach to recognizing physical activities, *in* 'Pervasive Computing', Vol. 3968/2006 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 1–16.

Logan, B., Healey, J., Philipose, M., Tapia, E. M. & Intille, S. S. (2007), A long-term evaluation of sensing modalities for activity recognition, *in* 'UbiComp 2007: Ubiquitous Computing', Vol. 4717/2007 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 483–500.

Patterson, D. J., Fox, D., Kautz, H. & Philipose, M. (2005), Fine-grained activity recognition by aggregating abstract object usage, *in* 'Proceedings of the Ninth IEEE International Symposium on Wearable Computers (ISWC'05)', IEEE Computer Society, Washington, DC, USA, pp. 44–51.

Philipose, M., Fishkin, K. P., Perkowitz, M., Patterson, D. J., Fox, D., Kautz, H. & Hahnel, D. (2004), 'Inferring activities from interactions with objects', *IEEE Pervasive Computing* **3**, 50–57.

Stikic, M., Laerhoven, K. V. & Schiele, B. (2008), Exploring semi-supervised and active learning for activity recognition, *in* 'Proceedings of the 12th IEEE International Symposium on Wearable Computers (ISWC)', pp. 81–88.

Szalai, S. (1973), *Daily Activities of Urban and Suburban Populations in Twelve Countries*, Mouton, The Hague.

Tapia, E. M., Intille, S. S. & Larson, K. (2004), Activity recognition in the home setting using simple and ubiquitous sensors, *in* 'Pervasive Computing', Vol. 3001/2004 of *Lecture notes in computer science*, Springer, pp. 158–175.

Wang, S., Pentney, W., Popescu, A.-M., Choudhury, T. & Philipose, M. (2007), Common sense based joint training of human activity recognizers, *in* 'Proceedings of the 20th international joint conference on Artifical intelligence (IJCAI'07)', Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 2237–2242.

Witten, I. H. & Frank, E. (2005), *Data mining: practical machine learning tools and techniques*, 2nd edn, Morgan Kaufmann, 500 Sansome Street, Suite 400, San Francisco, CA 94111.

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 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.

Volume 147 - Computer Science 2014
Edited by Bruce Thomas, University of South Australia and Dave Parry, AUT University, New Zealand. January 2014. 978-1-921770-30-2.
Contains the proceedings of the Australian System Safety Thirty-Seventh Australasian Computer Science Conference (ACSC 2014), Auckland, New Zealand, 20 – 23 January 2014.

Volume 148 - Computing Education 2014
Edited by Jacqueline Whalley, AUT University, New Zealand and Daryl D'Souza, RMIT University, Australia. January 2014. 978-1-921770-31-9.
Contains the proceedings of the Sixteenth Australasian Computing Education Conference (ACE2014), Auckland, New Zealand, 20 – 23 January 2014.

Volume 149 - Information Security 2014
Edited by Udaya Parampalli, University of Melbourne, Australia and Ian Welch, Victoria University of Wellington, New Zealand. January 2014. 978-1-921770-32-6.
Contains the proceedings of the Twelfth Australasian Information Security Conference (AISC 2014), Auckland, New Zealand, 20 – 23 January 2014.

Volume 150 - User Interfaces 2014
Edited by Burkhard C. Wünsche, University of Auckland, New Zealand and Stefan Marks, AUT University, New Zealand. January 2014. 978-1-921770-33-3.
Contains the proceedings of the Fifteenth Australasian User Interface Conference (AUIC 2014), Auckland, New Zealand, 20 – 23 January 2014.

Volume 151 - Australian System Safety Conference 2013
Edited by Tony Cant, Defence Science and Technology Organisation, Australia. May 2013. 978-1-921770-38-8.
Contains the proceedings of the Australian System Safety Conference (ASSC 2013), Adelaide, Australia, 22 – 24 May 2013.