Rafiqul Islam · Yun Sing Koh
Yanchang Zhao · Graco Warwick
David Stirling · Chang-Tsun Li
Zahidul Islam (Eds.)

Communications in Computer and Information Science 996

Data Mining

16th Australasian Conference, AusDM 2018
Bahrurst, NSW, Australia, November 28–30, 2018
Revised Selected Papers

Springer

Communications in Computer and Information Science 996

More information about this series at http://www.springer.com/series/7899

Rafiqul Islam · Yun Sing Koh
Yanchang Zhao · Graco Warwick
David Stirling · Chang-Tsun Li
Zahidul Islam (Eds.)

Data Mining

16th Australasian Conference, AusDM 2018
Bahrurst, NSW, Australia, November 28–30, 2018
Revised Selected Papers

Springer

Editors
Rafiqul Islam
School of Computing and Mathematics
Charles Sturt University
Albury, NSW, Australia

Yun Sing Koh
University of Auckland
Auckland, New Zealand

Yanchang Zhao
CSIRO Scientific Computing
Canberra, Australia

Graco Warwick
Data Science and Engineering
Australian Taxation Office
Canberra, Australia

David Stirling
Department of Information Technology
University of Wollongong
Wollongong, NSW, Australia

Chang-Tsun Li
School of Computing and Mathematics
Charles Sturt University
Wagga Wagga, Australia

Zahidul Islam
School of Computing and Mathematics
Charles Sturt University
Bathurst, Australia

ISSN 1865-0929 ISSN 1865-0937 (electronic)
Communications in Computer and Information Science
ISBN 978-981-13-6660-4 ISBN 978-981-13-6661-1 (eBook)
https://doi.org/10.1007/978-981-13-6661-1

Library of Congress Control Number: 2019931946

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721,
Singapore

Preface

It is our great pleasure to present the proceedings of the 16th Australasian Data Mining Conference (AusDM 2018) held at Charles Sturt University, Bathurst, during November 28–30, 2018. The Australasian Data Mining (AusDM) Conference series first started in 2002 as a workshop that was initiated by Dr. Simeon Simoff (then Associate Professor, University of Technology, Sydney), Dr. Graham Williams (then Principal Data Miner, Australian Taxation Office, and Adjunct Professor, University of Canberra), and Dr. Markus Hegland (Australian National University). The conference series has grown significantly since it first starter and is continuing to grow under the leadership of Professors Simoff and Williams as the chairs of the Steering Committee.

The Australasian Data Mining Conference brings together exciting and novel research contributions and their applications for solving real-life problems through intelligent data analysis of (usually large) data sets. The conference focuses on novel contributions on data collection, cleansing, pre-processing, knowledge discovery, making sense of data, knowledge presentation, future prediction, applications of data mining, and various issues related to data mining such as privacy and security. It welcomes researchers and industry practitioners to come together to discuss current problems and share ideas on the possible ways data scientists can contribute to solving the problems through amazing data mining algorithms and their applications. The conference series also gives a wonderful opportunity for postgraduate students in data mining to present their research innovations and network with other students, leading researchers, and industry experts. The conference series has been serving the data mining community with all these aims and achievements for the past 16 years without a miss.

In the past few years, AusDM operated in dual tracks covering both research and application sides of data science. This year AusDM further expanded with a number of main tracks and special tracks. The three main tracks were Research, Application, and Industry Showcase. The Research track also presented four special tracks: Image Data Mining, Identification Through Data Mining, Mobile and Sensor Network Data Mining, and Statistics in Data Science. The Research track and its four special tracks were meant for academic submissions whereas the Application track and Industry Showcase track were for industry submissions.

All academic submissions were reviewed following the same review process to ensure a fair review and all accepted papers were published in the same proceedings, regardless of the categories and tracks. Papers accepted from a special track were grouped together in the same session of the conference.

After partnering with various conferences for the past few years through co-location, this year AusDM was again located on its own as an independent conference on data mining. The conference maintained a well-designed website from almost a year before the actual conference starting date with clear instructions and information on the tracks, submission process, proceedings, author guidelines, keynote speakers, and tutorials.

The inclusion of a new proceedings chair role in the Organizing Committee emphasizes AusDM's commitment to timely publication of the accepted papers. The accepted papers were published by Springer in their CCIS series. The inclusion of several special tracks showed AusDM's impetus to reach out to research communities that are closely related to data mining and bring all parties on the common AusDM platform. The conference was well announced and publicized through various forums including social network sites such as Facebook, LinkedIn, and Twitter. Members of the Organizing Committee and the Steering Committee contributed greatly to make the conference successful. All these efforts have resulted in a significant increase in the number of submissions to AusDM 2018.

This year AusDM received 98 submissions. Of these, 18 submissions were disqualified for not meeting submission requirements and the remaining 80 submissions went through a double-blind review process. Academic submissions and Application track papers received at least three peer review reports and Industry Showcase track papers received two peer review reports. Additional reviewers were considered for a clear review outcome, if review comments from the initial three reviewers were unclear. As a result, some papers received up to six review reports.

Out of these 80 submissions, a total of 30 papers were finally accepted for publication. The overall acceptance rate for this year was 37.5%. Out of 58 academic submissions, 22 papers (i.e., 38%) were accepted for publication. Out of 18 submissions in the Application track, seven papers (i.e., 39%) were accepted for publication. Out of four submissions in the Industry Showcase track, one paper (i.e., 25%) was accepted for publication. All papers from the Industry Showcase track were invited for oral presentation. This year, authors from 19 different countries submitted their papers. The top five countries, in terms of the number of authors who submitted papers to AusDM 2018, were Australia (122 authors), China (17 authors), USA (11 authors), New Zealand (9 authors), and India and the UK (7 authors each).

The AusDM 2018 Organizing Committee would like to give their special thanks to Professors Geoffrey Webb, Junbin Gao, Graham Williams, and Yanchang Zhao for kindly accepting the invitation to give keynote speeches and tutorials. The committee would also like to give their sincere thanks to Charles Sturt University for providing administrative support (including the employment of a new temporary admin officer) and the conference venue that contributed to a successful event. The committee would also like to thank Springer CCIS and the Editorial Board for their acceptance to publish AusDM 2018 papers. This will give excellent exposure of the papers accepted for publication. We would also like to give our heartfelt thanks to all student volunteers at Charles Sturt University, who did a tremendous job. Last but not least, we would like to give our sincere thanks to all delegates for attending the conference this year and we hope you enjoyed AusDM 2018.

November 2018 Zahidul Islam

Organization

Conference Chairs

Md Zahidul Islam	Charles Sturt University, Australia
Chang-Tsun Li	Charles Sturt University, Australia

Program Chairs (Research Track)

Md. Rafiqul Islam	Charles Sturt University, Australia
Yun Sing Koh	University of Auckland, New Zealand

Program Chairs (Application Track)

Yanchang Zhao	CSIRO, Sydney, Australia
David Stirling	University of Wollongong, Australia

Program Chair (Industry Showcase)

Warwick Graco	Australian Taxation Office, Australia

Proceedings Chair

Kok-Leong Ong	La Trobe University, Australia

Tutorial Chairs

Jixue Liu	University of South Australia, Australia
Yee Ling Boo	RMIT University, Australia

Web Chairs

Michael Furner	Charles Sturt University, Australia
Ling Chen	University of Technology Sydney, Australia

Competition Chairs

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

Publicity Chairs

Ji Zhang	University of Southern Queensland, Australia
Ashad Kabir	Charles Sturt University, Australia

Special Track Chairs (Image Data Mining)

Lihong Zheng	Charles Sturt University, Australia
Xufeng Lin	Charles Sturt University, Australia

Special Track Chairs (Statistics in Data Science)

Azizur Rahman	Charles Sturt University, Australia
Ryan Ip	Charles Sturt University, Australia
Hien Nguyen	La Trobe University, Melbourne, Australia

Special Track Chairs (Sensor Data Mining)

Quazi Mamun	Charles Sturt University, Australia
Sabih Rehman	Charles Sturt University, Australia

Special Track Chairs (Identification Through Data Mining)

Xufeng Lin	Charles Sturt University, Australia
Xingjie Wei	University of Bath, UK

Steering Committee Chairs

Simeon Simoff	University of Western Sydney, Australia
Graham Williams	Microsoft

Steering Committee Members

Peter Christen	The Australian National University, Canberra, Australia
Ling Chen	University of Technology, Sydney, Australia
Zahid Islam	Charles Sturt University, Australia
Paul Kennedy	University of Technology, Sydney, Australia
Jiuyong (John) Li	University of South Australia, Adelaide, Australia
Richi Nayak	Queensland University of Technology, Brisbane, Australia
Kok–Leong Ong	La Trobe University, Melbourne, Australia
Dharmendra Sharma	University of Canberra, Australia
Glenn Stone	Western Sydney University, Australia
Andrew Stranieri	Federation University Australia, Mount Helen, Australia
Yanchang Zhao	CSIRO, Sydney, Australia

Honorary Advisors

John Roddick	Flinders University, Australia
Geoff Webb	Monash University, Australia

Program Committee

Research Track

Cheng Li	Deakin University, Australia
Yonghua Cen	Nanjing University of Science and Technology
Kewen Liao	Swinburne University of Technology, Australia
Zheng Lihong	Charles Sturt University, Australia
Ailian Jiang	Taiyuan University of Technology, China
Xuan-Hong Dang	IBM T.J.Watson
Turki Turki	King Abdulaziz University, Saudi Arabia
Philippe Fournier-Viger	Harbin Institute of Technology (Shenzhen), China
Pradnya Kulkarni	MIT World Peace University, India
Wei Shao	RMIT University, Australia
Jianzhong Qi	The University of Melbourne, Australia
Chayapol Moemeng	Assumption University
Imdadullah Khan	Lahore University of Management Sciences, India
Md Anisur Rahman	Charles Sturt University, Australia
Ashad Kabir	Charles Sturt University, Australia
Azizur Rahman	Charles Sturt University, Australia
Richard Gao	Commonwealth Australia, Australia
Saiful Islam	Griffith University, Australia
Md Nasim Adnan	Charles Sturt University, Australia
Xiaohui Tao	University of Southern Queensland, Australia
Ahmed Mohiuddin	Canberra Institute of Technology, Australia
David Huang	The University of Auckland, New Zealand
Hien Nguyen	La Trobe University, Australia
Adil Bagirov	University of Ballarat, Australia
R. Uday Kiran	The University of Tokyo, Japan
Yongsheng Gao	Griffith University, Australia
Lei Wang	University of Wollongong, Australia
Alan Wee-Chung Liew	Griffith University, Australia
Veelasha Moonsamy	Utrecht University
Flora Dilys Salim	RMIT University, Australia
Joy Wang	Union College
Rui Zhang	The University of Melbourne, Australia
Louie Qin	University of Huddersfield
Quang Vinh Nguyen	Western Sydney University, Australia
Mendis Champake	Charles Sturt University, Australia

Sitalakshmi Venkatraman	Melbourne Polytechnic, Australia
Mohammad Saiedur Rahaman	RMIT University, Australia
Xiangjun Dong	School of Information, Qilu University of Technology, China
Dhananjay Thiruvady	Monash University, Australia
Kiheiji Nishida	Hyogo University of Health Sciences, General Education Center, Japan
Md Geaur Rahman	Bangladesh Agricultural University
Zahirul Hoque	UAE University
Fulong Chen	Anhui Normal University, China
Guandong Xu	University of Technology, Sydney, Australia
Xingjie Wei	University of Cambridge, UK
Dharmendra Sharma	University of Canberra, Australia
Truyen Tran	Deakin University, Australia
Qinxue Meng	University of Technology, Sydney, Australia
Gang Li	Deakin University, Australia
Dinusha Vatsalan	CSIRO, Australian National University, Australia
Grace Rumantir	Monash University, Australia
Ji Zhang	The University of Southern Queensland, Australia
Muhammad Marwan Muhammad Fuad	Technical University of Denmark
Kok-Leong Ong	La Trobe University, Australia
Chang-Tsun Li	Charles Sturt University, Australia
Paul Grant	Charles Sturt University, Australia
Nectarios Costadopoulos	Charles Sturt University, Australia
Darren Yates	Charles Sturt University, Australia
Khondker Jahid Reza	Charles Sturt University, Australia
Diana Benavides Prado	University of Auckland, New Zealand
Robert Anderson	University of Auckland, New Zealand
Kylie Chen	University of Auckland, New Zealand

Application Track

Rohan Baxter	Australian Tax Office, Australia
Nathan Brewer	Department of Human Services, Australia
Neil Brittliff	University of Canberra, Australia
Adriel Cheng	Defence Science and Technology Group, Australia
Hoa Dam	University of Wollongong, Australia
Klaus Felsche	C21 Directions, Australia
Markus Hagenbuchner	University of Wollongong, Australia
Edward Kang	Australian Passport Office, Australia

Stefan Keller	Teradata, Australia
Sebastian Kurscheid	Australian National University, Australia
Fangfang Li	oOh! Media, Australia
Jin Li	Geoscience Australia, Australia
Chao Luo	Department of Health, Australia
Atif Mahmood	Department of Home Affairs, Australia
Kee Siong Ng	Australian Government, Australia
Tom Osborn	Epifini.ai, Australia
Martin Rennhackkamp	PBT Group Australia, Australia
Goce Ristanoski	CSIRO Data61, Australia
Chao Sun	The University of Sydney, Australia
Tilly Tan	PwC Australia, Australia
Kerry Taylor	Australian National University and University of Surrey, Australia
Jie Yang	University of Wollongong, Australia
Ting Yu	Commonwealth Bank of Australia, Australia

Contents

Classification Tasks

Transport, Environment and Energy

Applied Data Mining

Privacy and Clustering

Statistics in Data Science

Health, Software and Smart Phone

Image Data Mining

Industry Showcase

Classification Tasks

Misogynistic Tweet Detection: Modelling CNN with Small Datasets

Md Abul Bashar[1](✉), Richi Nayak[1], Nicolas Suzor[2], and Bridget Weir[2]

[1] School of Electrical Engineering and Computer Science, Brisbane, Australia
{m1.bashar,r.nayak}@qut.edu.au

[2] School of Law, Queensland University of Technology, Brisbane, Australia
{n.suzor,bridget.weir}@qut.edu.au

Abstract. Online abuse directed towards women on the social media platform such as Twitter has attracted considerable attention in recent years. An automated method to effectively identify misogynistic abuse could improve our understanding of the patterns, driving factors, and effectiveness of responses associated with abusive tweets over a sustained time period. However, training a neural network (NN) model with a small set of labelled data to detect misogynistic tweets is difficult. This is partly due to the complex nature of tweets which contain misogynistic content, and the vast number of parameters needed to be learned in a NN model. We have conducted a series of experiments to investigate how to train a NN model to detect misogynistic tweets effectively. In particular, we have customised and regularised a Convolutional Neural Network (CNN) architecture and shown that the word vectors pre-trained on a task-specific domain can be used to train a CNN model effectively when a small set of labelled data is available. A CNN model trained in this way yields an improved accuracy over the state-of-the-art models.

1 Introduction

Incidents of abuse, hate, harassment and misogyny have proliferated with the growing use of social media platforms (e.g. Twitter, Facebook, Instagram). These platforms have generated new opportunities to spread online abuse [5]. The experience of online abuse is a common occurrence for women [12]. Often these experiences of online abuse can be categorised as sexist or misogynistic in nature, and can include name-calling and offensive language, threats of harm or sexual violence, intimidation, shaming, and the silencing of women. While it is easy to identify instances of abuse and major abusive campaigns on social media, it is difficult to understand changes in levels of abuse over time, and almost impossible to identify the effectiveness of interventions by platforms in combating abuse [24]. An automated system to identify abusive tweets could help in ongoing efforts to develop effective remedies.

A key challenge in the automatic detection of misogynistic abusive tweets is understanding the context of an individual tweet. We focus here on misogynistic tweets that are abusive towards an individual or group – a subset of the larger

R. Islam et al. (Eds.): AusDM 2018, CCIS 996, pp. 3–16, 2019.
https://doi.org/10.1007/978-981-13-6661-1_1

category of tweets that include sexist or misogynistic words or concepts. Accordingly, this study sought to address the difficult task of separating abusive tweets from tweets that were sarcastic, joking, or contained misogynistic keywords in a non-abusive context. A lexical detection approach tends to have low accuracy [4, 28] because they classify all tweets containing particular keywords as misogynistic. Xiang et al. [28] reported that bag-of-words, part-of-speech (POS) and belief propagation did not work well for the detection of profane tweets because of the significant noise in tweets. For example, tweets do not follow a standard language format, words are often misspelled or altered, and tweets often include words from local dialects or foreign languages. The automated algorithms should look for patterns, sequences, and other complex features that are present, despite the noise, and are correlated with misogynistic tweets. Traditional algorithms (e.g. Random Forest, Logistic Regression and Support Vector Machines) rely on manual process to obtain these kinds of features and are limited by the kinds of features available. Neural Network (NN)-based models, on the other hand, can automatically learn complex features and effectively use them to classify a given instance.

Relying on a sufficiently large training dataset, CNN models have shown to be effective in Natural Language Processing (NLP) tasks such as semantic parsing [29], document query matching [22], sentence classification [13], etc. A set of convolving filters are applied to local features (e.g. words) to learn patterns similar to nGrams. Local features are commonly represented as word vectors where words are projected from a sparse representation onto a lower dimensional vector space. Word vectors essentially encode semantic features of each word in a fixed number of abstract topics (or dimensions). The apparent success of CNN in NLP tasks can be credited to its capability to learn text patterns in semantic space. However, given the requirement of setting the large number of parameters in CNN, often in millions, the CNN models are trained on a huge labelled dataset. In general, this is a limitation of any NN-based model [6]. Overall, curating a large set of labelled tweets containing misogynistic abuse is difficult and costly to achieve due to the large amount of data that needs to be manually examined to rigorously identify abusive tweets.

Existing experiments have shown that pre-trained word vectors (vectors trained on general-purpose corpus) can improve the prediction accuracy of CNN models. However, word vectors trained on a general-purpose corpus cannot capture the task-specific semantics because the nature of general-purpose corpus and the misogynistic tweets is completely different. For example, many words used in these tweets are linguistically specific and unique to Twitter-based discussion, and are not covered in a general-purpose corpus. Consequently, a CNN classifier model built using these word vectors cannot adequately detect misogynistic abusive tweets.

In this paper, we investigate the effectiveness of various corpus to generate pre-trained word vectors for building a CNN model when there is a small set of labelled data available. In particular, we trained a CNN using a small set of labelled data to detect misogynistic tweets. We pre-trained word vectors on

0.2 billion unlabelled tweets that contain at least one misogynistic keyword (i.e. *whore, slut, rape*). We customised and regularised the CNN architecture used in [13]. On the test dataset, the trained CNN model achieves significantly better results than the state-of-the-art models. It is better by a large margin in comparison to the CNN models build on word vectors pre-trained on a sizeable general corpus. The experimental results show that a CNN classifier can be trained on a small labelled tweet data, provided that the word vectors are pre-trained in the context of the problem domain and a careful model customisation and regularisation is performed.

This project investigates how to effectively apply data mining methods, with a focus on training a NN model, to detect misogynistic tweets. Three main contributions of this paper are: (a) it shows that word vectors pre-trained on a task-specific domain can be used to effectively train CNN when a small set of labelled data is available; (b) it shows how to customise and regularise a CNN architecture to detect misogynistic tweets; and (c) finally, we present an automated data mining method to detect misogynistic abusive tweets.

2 Related Work

Misogynistic abusive tweet detection falls into the research area of text classification. Popular text classification algorithms used in hate speech and offensive language detection are Naive Bayes [4], Logistic Regression [4], Support Vector Machine (SVM) [4,26,28] and Random Forest [4,28]. Performance of these algorithms depend on feature engineering and feature representation [4,28]. There have been some works where syntactic features are leveraged to identify the targets and the intensity of hate speech. Examples of these features are relevant verb and noun occurrences (e.g. *kill* and *Jews*) [7], and the syntactic structures: I <intensity><user intent><hate target> (e.g. *I f∗cking hate white people*) [23].

Misogynistic tweet detection is challenging for text classification methods because social media users very commonly use offensive words or expletives in their online dialogue [25]. For example, the bag-of-words approach is straightforward and usually has a high recall, but it results in higher number of false positives because the presence of misogynistic words causes these tweets to be misclassified as abusive tweets [14].

Recently, neural network-based classifiers have become popular as they automatically learn abstract features from the given input feature representation [1]. Input to these algorithms can be various forms of feature encoding, including many of those used in the classic methods. Algorithm design in this category focuses on the selection of the network topology to automatically extract useful abstract features. Popular network architectures are CNN, Recurrent Neural Networks (RNN) and Long Short-Term Memory network (LSTM). CNN is well known for extracting patterns similar to phrases and nGrams [1]. On the other hand, RNN and LSTM are effective for sequence learning such as order information in text [1]. The CNN model has been successfully used for sentence classification [13]. To effectively identify patterns in the text, they used word embedding pre-trained on Google News corpus while training a CNN model on the labelled dataset.

The neural network-based classifiers have not yet been applied in misogynistic tweet detection. It requires a rigorous investigation as to what extent patterns and orderly information are present in misogynistic tweets, and how we can optimise a Neural Network for classification accuracy. There are many CNN architectures used in the current literature, but the design of an architecture heavily depends on the problem at hand. Therefore, a customised CNN architecture is needed to classify misogynistic tweets. It also remains to be seen whether the word embedding can be sensitive to the domain knowledge of the corpus. Does the word embedding need to be trained on a similar tweet stream to capture contextual properties?

3 Problem Formulation

Misogynistic abusive tweets may contain misogynistic keywords, but tweets can also be misogynstic abuse without explicitly containing these slurs. Further, not all tweets that contain misogynistic keywords are abusive. Classifying misogynistic abuse in tweets requires close reading, and even humans can struggle to classify these tweets accurately. The focus of this research is to detect abusive tweets that contain misogynistic words. A previous study has identified that three keywords – *whore*, *slut* and *rape* – are useful in identifying a substantial portion of misogynistic tweets [2]. However, these misogynistic words are commonly used in tweets that are not abusive, and separating abusive tweets from non-abusive tweets is difficult when we base our classification purely on the occurrence of these words. We propose a two-step method to approach this problem:

- Pre-filtering: We pre-filter tweets that contain any of the three main misogynistic keywords (slut, rape, whore) to find potentially-misogynistic tweets.
- Training a CNN model: Using a small labelled data set, a CNN model is trained to classify the remaining potentially-abusive tweets. We propose several methods to accurately train the model.

The research team used a systematic approach to generate the labelled data manually. The following contextual information was used in assessing whether a tweet contains targeted misogynistic abuse, or not: (a) Is a specific person or group being targeted in this tweet? (b) Does this tweet contain a specific threat or wish for violence? (c) Does this tweet encourage or promote self-harm or suicide? (d) Is the tweet harassing a specific person, or inciting others to harass a specific person? (e) Does the tweet use misogynistic language in objectifying a person, making sexual advances, or sending sexually explicit material? (f) Is the tweet promoting hateful conduct by gender, sexual orientation, etc.?

The labelled tweets reveal many challenges that need to be addressed to train a classifier effectively. These included: (a) The misogynistic words are not the discriminatory words. Many keywords are overlapping between misogynistic and non-misogynistic tweets, especially misogynistic keywords. (b) Words may be misspelt and spelt in many ways. (c) Sometimes people mix words from local dialects or foreign languages. (d) The data is noisy and does not follow a standard language sequence (format). (e) Effectively detecting misogynistic tweets needs access to semantics and context information that is often not available, e.g. it is difficult to use dictionary-based semantics for the nature of noise in tweets and difficult to know the context because of the small length of a tweet. (f) The labelling process is time consuming and it is extremely difficult to generate a large quantity of labelled data because only a very small portion of tweets can be identified as misogynistic.

Given these challenges, in this paper, we investigate how to effectively train a CNN model with a small set of labelled data to detect misogynistic abusive tweets. We train a CNN on top of the pre-trained word vectors (a.k.a. word embedding or distributed representation of words). The primary focus is to find out what kind of pre-trained word vectors is useful to train a CNN with a small dataset. Another two important focuses are to find out what customised architecture of CNN is effective in the given problem and to test the effectiveness of some simple data and feature augmentation.

4 Word Embedding

Word embedding models map each word from the vocabulary to a vector of real numbers. They aim to quantify and categorise semantic similarities between words based on their distributional property based on the premise that a word is characterised by the company it keeps. Given a sizeable unlabelled corpus, these models can effectively learn a high-quality word embedding. Based on the feed-forward neural network, Mikolov et al. [20] proposed two popular models: Skip-gram and Continuous Bag-of-Words as shown in Fig. 1.

Given the words within a sliding window, the continuous bag-of-words model predicts the current word w_i from the surrounding context words C, i.e. $p(w_i|C)$. In contrast, the skip-gram model uses the current word w_i to predict the surrounding context words C, i.e. $p(C|w_i)$. In Fig. 1, for example, if the current position of a running sliding window contains the phrase *she looks like a crack whore*. In continuous bag-of-words, the context words {she, looks, like, a, whore} can be used to predict the current word {crack}, whereas, in skip-gram, the current word {crack} can be used to predict the context words {she, looks, like, a, whore}.

The training objective is to find a word embedding that maximises $p(t_i|C)$ or $p(C|t_i)$ over a training dataset. In each step of training, each word is either (a) pulled closer to the words that co-occur with it or (b) pushed away from all the words that do not co-occur with it. A *softmax* or *approximate softmax* function can be used to achieve this objective [20]. At the end of the training, the embedding brings closer not only the words that are explicitly co-occurring in a training dataset, but also the words that implicitly co-occur. For example, if t_1 explicitly co-occurs with t_2 and t_2 explicitly co-occurs with t_3, then the model can bring closer not only t_1 to t_2, but also t_1 to t_3. The continuous bag-of-words model is faster and has slightly better accuracy for the words that appear frequently. Therefore, we use this model in this research.

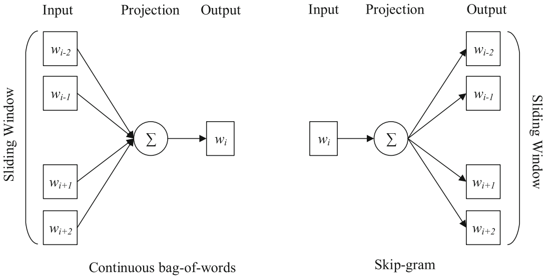


Fig. 1. Word embedding models

5 Model Architecture

We empirically customise and regulate Kim's [13] CNN architecture to detect misogynistic tweets and reduce overfitting. Figure 2 shows the architecture. We use word embedding to represent each word w in an n-dimensional word vector $\mathbf{w} \in \mathbb{R}^n$. A tweet t with m words is represented as a matrix $\mathbf{t} \in \mathbb{R}^{m \times n}$. Convolution operation is applied to the tweet matrix with one stride. Each convolution operation applies a filter $\mathbf{f}_i \in \mathbb{R}^{h \times n}$ of size h. Empirically, based on the accuracy improvement in ten-fold cross validation, we used 256 filters for $h \in \{3, 4\}$ and 512 filters for $h \in \{5\}$. The convolution is a function $\mathbf{c}(\mathbf{f}_i, \mathbf{t}) = r(\mathbf{f}_i \cdot \mathbf{t}_{k:k+h-1})$, where $\mathbf{t}_{k:k+h-1}$ is the kth vertical slice of the tweet matrix from position k to $k + h - 1$, $\mathbf{f}_i$ is the given filter and r is a ReLU function. The function $\mathbf{c}(\mathbf{f}_i, \mathbf{t})$ produces a feature c_k similar to nGrams or phrases for each slice k, resulting in $m - h + 1$ features. We apply the max-pooling operation over these features and take the maximum value, i.e. $\hat{c}_i = \max \mathbf{c}(\mathbf{f}_i, \mathbf{t})$. Max-pooling is carried to capture the most important feature for each filter. As there are a total of 1024 filters $(256 + 256 + 512)$ in the proposed model, the 1024 most important features are learned from the convolution layer.

These features are passed to a fully connected hidden layer with 256 perceptrons that use the ReLU activation function. This fully connected hidden layer allows learning the complex non-linear interactions between the features from the convolution layer and generates 256 higher level new features. Finally these 256 higher level features are passed to the output layer with single perceptron that uses the sigmoid activation function. The perceptron in this layer generates the probability of the tweet being misogynistic.

We randomly dropout a proportion of units from each layer except the output layer by setting them to zero. This is done to prevent co-adaptation of units in a layer and to reduce overfitting. We empirically dropout 50% units from the input layer, the filters of size 3 and the fully connected hidden layer. We dropout only 20% units from the filters of size 4 and 5.

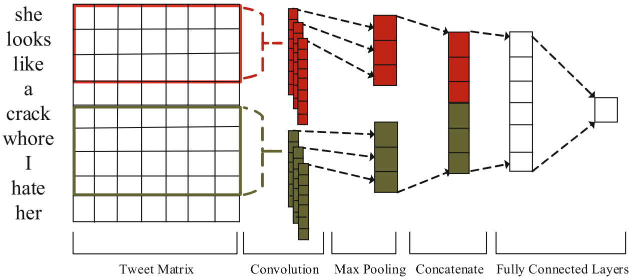


Fig. 2. CNN model architecture

6 Empirical Evaluation

The primary objectives of the experiments are to show: (a) word vectors pre-trained on a task-specific domain is more effective than those pre-trained on a sizeable general corpus; (b) CNN trained on a small dataset and built on word vectors pre-trained on a task-specific domain can perform better than the state-of-the-art models; and (c) the impact of some simple data and word augmentation techniques on training a CNN model.

6.1 Data Collection

Labelled Tweets: We collected tweets using Twitter's streaming API. For the labelled dataset, we identified 10k tweets that contain any of the three main misogynistic keywords (i.e., whore, slut, rape). Following the misogynistic tweet definition in Sect. 3, the research team labelled a total of 5000 tweets with 1800 misogynistic and 3200 non-misogynistic labels. A stratified data selection was made to reduce a trained models' bias to a specific label, i.e. we kept 1800 misogynistic and 1800 randomly selected nonmisogynistic tweets. We used 80% examples for training and 20% for testing. We used ten-fold cross-validation to tune hyperparameters and Porter's suffix-stripping algorithm for preprocessing.

The tweet labelling method has the following limitations: (a) The coding is based on a literal interpretation of the text; with limited context, we are likely to include some sarcasm or humour. (b) We are only labelling tweets written in English. (c) Identifying the tweets by keywords only, we will not catch abuse that appears to be ordinary misogyny, e.g. *get back in the kitchen*. (d) Identifying the tweets by keywords only, we will not identify harassment that is targeted and organised harassment, either ongoing over time or involving many participants, but does not use one of our keywords.

WikiNews: Word vectors of 300-dimension pre-trained on the Wikipedia 2017, UMBC webbase corpus and statmt.org news datasets containing a total of 16 billion words using fastText (a library for learning word embeddings created by Facebook's AI Research lab) [19].

GoogleNews: Word vectors of 300-dimension pre-trained on Google News corpus containing a total of three billion words using the Continuous Bag-of-Words Word2vec model [18].

Potentially Misogynistic Tweets: Word vectors of 200-dimension pre-trained on 0.2 billion tweets that contain any of the three main misogynistic keywords. A Continuous Bag-of-Words Word2vec model is used in pre-training while minimum count for word is set to 100.

6.2 Evaluation Measures

We used six standard evaluation measures of classification performance: Accuracy, Precision, Recall, F_1 Score, Cohen Kappa (CK) and Area Under Curve (AUC). We also report True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) values.

6.3 Baseline Models

We have implemented eight baseline models to compare the performance with the proposed CNN model.

- Long Short-Term Memory Network (LSTM) [10]. We have implemented LSTM with 100 units, 50% dropout, binary cross-entropy loss function, Adam optimiser and sigmoid activation.
- Feedforward Deep Neural Network (DNN) [8]. We have implemented DNN with five hidden layers, each layer containing eight units, 50% dropout applied to the input layer and the first two hidden layers, softmax activation and 0.04 learning rate. For all neural network based models (CNN, LSTM, DNN), hyperparameters are manually tuned based on ten-fold cross-validation.
- Non NN models including Support Vector Machines (SVM) [9], Random Forest [17], XGBoost (XGB) [3], Multinomial Naive Bayes (MNB) [16], k-Nearest Neighbours (kNN) [27] and Ridge Classifier (RC) [11]. Hyperparameters of all these models are automatically tuned using ten-fold cross-validation and GridSearch from sklearn. All the baseline models, except LSTM, are trained using only labelled tweets.

6.4 Results and Discussion

Word Embedding Performances. We conducted experiments to see the effects of different word embeddings in training the CNN model. A summary of the embeddings is given in Fig. 3 and the experimental results are given in Fig. 4. Three main observations from the results are: (a) Word vectors pre-trained on a large dataset (e.g., WE1, WE2, WE4, WE5) always improves performance. The convolution layer, that captures nGram-like patterns in the tweets while using word vectors to represent the tweets, allows the model to find these patterns in semantic space. The pre-trained word embedding can provide the semantics of words that have fewer appearances in the training dataset. This reinforces the prior finding [21] that the features obtained from a pre-trained deep learning model perform well on a variety of tasks. (b) Updating the word vectors with the labelled data while training the classifier improves the performance (e.g., WE1 over WE2). This allows the semantics of words to be more focused over the training set. (c) Word vectors pre-trained on potentially misogynistic tweets and updated with labelled data performs the best. It improves the CNN accuracy by around 12% compared with word vectors pre-trained on a standard corpus (e.g. Google News corpus). This observation challenges the previous findings [13] that general pre-trained word vectors (e.g. word vectors pre-trained on Google News) are *universal* feature extractors. Due to the small labelled dataset used in training the CNN model, it was not enough to update the necessary word vectors for the problem domain, given that tweets are very noisy and mostly different from standard corpora like Google News or Wikipedia. The word vectors pre-trained on unlabelled datasets in the task-specific domain can address this problem.

The apparent performance correlation of CNN and word vector can be related to the similar learning technique that CNN and word vector use. In the word vector representation, semantically similar words are represented with similar vectors and semantically dissimilar words are represented with dissimilar vectors. This is obtained through training the word vectors on a corpus where it searches for co-occurring words through a filter called sliding window.

To train CNN with a labelled tweet, the words in the tweet are represented with word vectors. CNN discovers patterns in these word vectors based on co-occurring vector elements through varying length filters. Co-occurrence of vector elements depend on the word co-occurrence in the labelled dataset and the pre-training corpus. Therefore, pre-training corpus have significant effect on the CNN model.

In other words, because CNN learns the patterns in the vector space, it harnesses the patterns (or semantic relations) already learned in the vector space. Thus, pre-trained word vectors, especially trained on a corpus from the similar nature domain, may significantly improve the performance of CNN model when only a small labelled dataset is available for training.

Model	Description
WE1	W2V pre-trained on potentially abusive tweets and updated with labelled data
WE2	W2V pre-trained on potentially abusive tweets but not updated with labelled data
WE3	W2V Trained with only labelled data
WE4	W2V pre-trained on google news and updated with labelled data
WE5	fastText pre-trained on Wikipedia pages and updated with labelled data

Fig. 3. Summary of word embeddings

	WE1	WE2	WE3	WE4	WE5
TP	**267**	264	194	217	199
TN	**283**	279	273	274	281
FP	**78**	82	88	87	80
FN	**94**	97	167	144	162
Accuracy	**0.762**	0.752	0.647	0.680	0.665
Precision	**0.774**	0.763	0.688	0.714	0.713
Recall	**0.740**	0.731	0.537	0.601	0.551
F_1 Score	**0.756**	0.747	0.603	0.653	0.622
CK	**0.524**	0.504	0.294	0.360	0.330
AUC	**0.762**	0.752	0.647	0.680	0.665

Fig. 4. Performance of CNN applied on different word embeddings

Classifier Models Comparison. We implemented the proposed CNN model and the eight baseline models to detect misogynistic tweets. Guided by the experimental results in previous section, both CNN and LSTM models were built on word vectors that are pre-trained on potentially abusive tweets and updated with the labelled dataset during the classifier training. Performances of the models are summarised in Table 1.

Result shows that CNN outperforms all other models. For example, the improvement in precision, accuracy, Cohen Kappa score and AUR of CNN over the second best performing model LSTM are 6.120%, 4.364%, 13.855% and 4.364% respectively. LSTM is known to be effective in text datasets and the results reflect this. The reason for CNN outperforming LSTM and other baseline models might be the nature of tweets. Tweets are super condensed texts, full of noise and often do not follow the standard sequence of the language. Traditional models (e.g. RF, SVM, kNN, etc.) are based on bag-of-words representation that can be highly impacted by the significant noise in tweets [28]. Besides, the bag-of-words representation cannot capture sequences and patterns that are very important to identify a misogynistic tweet. For example, if a tweet contains a sequence *if you know what I mean*, there is a high chance that this tweet might be misogynistic, even though individual keywords are innocent.

The performance of LSTM is better than traditional models as it can capture sequences. However, sequences in tweets often get altered by noises (e.g. misspelled or intentionally altered by the author); therefore LSTM might struggles to detect misogynistic tweets. CNN models are well known for effectively discovering a large number of patterns and sub-patterns through many filters with varying size. If a few words of a given tweet are altered by noise it can still match a sub-pattern. This means CNN is less affected by noise. As a result CNN out performs LSTM.

CNN is popularly used in Computer Vision and is known to be effective only if the model is trained on massive datasets. However, in this research, we trained a simple CNN with only three thousand labelled tweets. This simple CNN uses only one layer of convolutions on top of word vectors, and it achieves significantly better results than state-of-the-art models. These results ascertain that a CNN can be trained on a small labelled dataset, provided that word vectors are pre-trained in the context of the problem domain, and a careful model customisation and some regularisations are performed.

Data Augmentation Performances. Data augmentation and document expansion is popularly used in computer vision and information retrieval respectively to artificially inflating a small labelled dataset and/or input vectors. In this paper, we augmented/expanded the data multiple ways and studied their impact on training the CNN model. We used two sources of data to generate augmented data: (1) the word vectors pre-trained on the potential misogynistic tweets; and (2) topics identified by Non-Negative Matrix Factorisation (NMF) [15] on the tweet training dataset, performed separately on each class. A total of six policies were followed. AT1: Words in a labelled tweet are randomly replaced

Table 1. Performances of classification models

	CNN	LSTM	DNN	SVM	RF	XGB	MNB	kNN	RC
TP	267	264	275	257	279	**286**	272	95	263
TN	283	263	171	244	229	223	251	**302**	245
FP	78	98	190	117	132	138	110	**59**	116
FN	94	97	86	104	82	**75**	89	266	98
Accuracy	**0.762**	0.730	0.618	0.694	0.704	0.705	0.724	0.550	0.704
Precision	**0.774**	0.729	0.591	0.687	0.679	0.675	0.712	0.617	0.694
Recall	0.740	0.731	0.762	0.712	0.773	**0.792**	0.753	0.263	0.729
F_1 Score	**0.756**	0.730	0.666	0.699	0.723	0.729	0.732	0.369	0.711
CK	**0.524**	0.460	0.235	0.388	0.407	0.410	0.449	0.100	0.407
AUC	**0.762**	0.730	0.618	0.694	0.704	0.705	0.724	0.550	0.704

Table 2. CNN results from data augmentation policies

	AT0	AT1	AT2	AT3	AT4	AT5	AT6
TP	267	270	242	301	281	292	288
TN	283	275	287	224	203	238	241
FP	78	86	74	137	158	123	120
FN	94	91	119	60	80	69	73
Accuracy	0.762	0.755	0.733	0.727	0.670	0.734	0.733
Precision	0.774	0.758	0.766	0.687	0.640	0.704	0.706
Recall	0.740	0.748	0.670	0.834	0.778	0.809	0.798
F_1 Score	0.756	0.753	0.715	0.753	0.703	0.753	0.749
CK	0.524	0.510	0.465	0.454	0.341	0.468	0.465
AUC	0.762	0.755	0.733	0.727	0.670	0.734	0.733

by semantically similar words from word vector space to create an artificial tweet. AT2: Discriminative Words in a labelled tweet are randomly replaced by semantically similar words from word vector space to create an artificial tweet. A discriminative word is a word that more frequently appears in the tweet of a specific label. AT3: A tweet is expanded by adding its semantically similar words found from word vector space. AT4: A tweet is expanded by adding its semantically similar words found from NMF. AT5: Use the discovered topics in NMF as artificial tweets. AT6: A set of words from word vector space that is semantically similar to a tweet is used as an artificial tweet.

Table 2 reports the performance of model trained with each of these augmentation policies and the CNN model trained with the original labelled dataset before any augmentation (labelled as AT0). The experimental results show that these ways of augmentation do not improve the accuracy. We conjecture that additional external features (i.e. words) may distort the patterns exist in the original tweets, since the CNN classifier largely depends on learning these patterns, the performance degrades.

7 Conclusions

This paper presents a novel method of misogynistic tweet detection using word embedding and the CNN model when only a small amount of labelled data is available. We report the results of a series of experiments conducted to investigate the effectiveness of training a model with a small dataset. We customised and regularised a CNN architecture, and it performs better than the state-of-the-art models, provided that the CNN is built on word vectors pre-trained on the task-specific domain. Experimental results show that a CNN model built on word vectors pre-trained on the task-specific unlabelled dataset is more effective than built on word vectors pre-trained on a sizeable general corpus. Experimental results also show that simple data augmentation policies are not adequate to improve misogynistic tweet detection performance in the CNN model.

Acknowledgement. This research was fully supported by the QUT IFE Catapult fund. Suzor is the recipient of an Australian Research Council DECRA Fellowship (project number DE160101542).

References

1. Badjatiya, P., Gupta, S., Gupta, M., Varma, V.: Deep learning for hate speech detection in tweets. In: Proceedings of the 26th International Conference on World Wide Web Companion, pp. 759–760. International World Wide Web Conferences Steering Committee (2017)
2. Bartlett, J., Norrie, R., Patel, S., Rumpel, R., Wibberley, S.: Misogyny on twitter. Demos (2014)
3. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794. ACM (2016)
4. Davidson, T., Warmsley, D., Macy, M., Weber, I.: Automated hate speech detection and the problem of offensive language (2017). arXiv preprint: arXiv:1703.04009
5. Dragiewicz, M., et al.: Technology facilitated coercive control: domestic violence and the competing roles of digital media platforms. Feminist Media Studies, pp. 1–17 (2018)
6. Fadaee, M., Bisazza, A., Monz, C.: Data augmentation for low-resource neural machine translation. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Short Papers, vol. 2, pp. 567–573 (2017)

7. Gitari, N.D., Zuping, Z., Damien, H., Long, J.: A lexicon-based approach for hate speech detection. Int. J. Multimed. Ubiquitous Eng. **10**(4), 215–230 (2015)
8. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 249–256 (2010)
9. Hearst, M.A., Dumais, S.T., Osuna, E., Platt, J., Scholkopf, B.: Support vector machines. IEEE Intell. Syst. Appl. **13**(4), 18–28 (1998)
10. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
11. Hoerl, A.E., Kennard, R.W.: Ridge regression: applications to nonorthogonal problems. Technometrics **12**(1), 69–82 (1970)
12. International, A.: Toxic twitter - a toxic place for women (2018)
13. Kim, Y.: Convolutional neural networks for sentence classification (2014). arXiv preprint: arXiv:1408.5882
14. Kwok, I., Wang, Y.: Locate the hate: detecting tweets against blacks. In: AAAI (2013)
15. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Advances in Neural Information Processing Systems, pp. 556–562 (2001)
16. Lewis, D.D.: Naive (Bayes) at forty: the independence assumption in information retrieval. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 4–15. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0026666
17. Liaw, A., Wiener, M., et al.: Classification and regression by randomforest. R News **2**(3), 18–22 (2002)
18. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: International Conference on Learning Representations (ICLR) Workshop (2013)
19. Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., Joulin, A.: Advances in pre-training distributed word representations. In: Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018) (2018)
20. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
21. Sharif Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S.: CNN features off-the-shelf: an astounding baseline for recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 806–813 (2014)
22. Shen, Y., He, X., Gao, J., Deng, L., Mesnil, G.: Learning semantic representations using convolutional neural networks for web search. In: Proceedings of the 23rd International Conference on World Wide Web, pp. 373–374. ACM (2014)
23. Silva, L.A., Mondal, M., Correa, D., Benevenuto, F., Weber, I.: Analyzing the targets of hate in online social media. In: ICWSM, pp. 687–690 (2016)
24. Suzor, N., Van Geelen, T., Myers West, S.: Evaluating the legitimacy of platform governance: a review of research and a shared research agenda. Int. Commun. Gazette **80**(4), 385–400 (2018)
25. Wang, W., Chen, L., Thirunarayan, K., Sheth, A.P.: Cursing in English on Twitter. In: Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing, pp. 415–425. ACM (2014)

26. Warner, W., Hirschberg, J.: Detecting hate speech on the world wide web. In: Proceedings of the Second Workshop on Language in Social Media, pp. 19–26. Association for Computational Linguistics (2012)
27. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. J. Mach. Learn. Res. **10**, 207–244 (2009)
28. Xiang, G., Fan, B., Wang, L., Hong, J., Rose, C.: Detecting offensive tweets via topical feature discovery over a large scale twitter corpus. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, pp. 1980–1984. ACM (2012)
29. Yih, W.t., He, X., Meek, C.: Semantic parsing for single-relation question answering. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. Short Papers, vol. 2, pp. 643–648 (2014)

Multiple Support Vector Machines for Binary Text Classification Based on Sliding Window Technique

Aisha Rashed Albqmi[1,2](✉), Yuefeng Li[1], and Yue Xu[1]

[1] School of EECS, Queensland University of Technology,
Brisbane, QLD, Australia
a.albqmi@hdr.qut.edu.au, {y2.li,yue.xu}@qut.edu.au
[2] Department of CS, Taif University, Taif, Saudi Arabia

Abstract. Supervised machine learning algorithms, such as support vector machines (SVMs), are widely used for solving classification tasks. In binary text classification, linear SVM has shown remarkable efficiency for classifying documents due to its superior performance. It tries to create the best decision boundary that enables the separation of positive and negative documents with the largest margin hyperplane. However, in most cases there are regions in which positive and negative documents are mixed due to the uncertain boundary. With an uncertain boundary, the learning classifier is more complex, and it often becomes difficult for a single classifier to accurately classify all unknown testing samples into classes. Therefore, more innovative methods and techniques are needed to solve the uncertain boundary problem that was traditionally solved by non-linear SVM. In this paper, multiple support vector machines are proposed that can effectively deal with the uncertain boundary and improve predictive accuracy in linear SVM for data having uncertainties. This is achieved by dividing the training documents into three distinct regions (positive, boundary, and negative regions) based on a sliding window technique to ensure the certainty of extracted knowledge to describe relevant information. The model then derives new training samples to build a multiple SVMs based classifier. The experimental results on the TREC topics and standard dataset Reuters Corpus Volume 1 (RCV1), indicated that the proposed model significantly outperforms six state-of-the-art baseline models in binary text classification.

Keywords: Support Vector Machines · Binary text classification · Uncertain boundary · Sliding window technique

1 Introduction

The massive amounts of unstructured data sorted in public resources continue to increase. In order to organize and manage this data, the use of efficient and successful methods must be considered. Text classification is an active technique for information organization and management [1]. Different methods and algorithms have been developed for text classification including Support Vector Machines (SVM) [2], Naive Bayes probabilistic Classifier (NB) [3], Rocchio Similarity [4], K-Nearest Neighbour (KNN) [5], and C4.5 integration Decision Trees [1].

R. Islam et al. (Eds.): AusDM 2018, CCIS 996, pp. 17–29, 2019.
https://doi.org/10.1007/978-981-13-6661-1_2

Binary classification is a key type of text classification with two predefined categories, namely, relevant or irrelevant classes [6], on which our research focuses. A binary text classifier determines a decision boundary to classify documents into two groups: positive and negative classes [7]. However, drawing a clear boundary between the positive and negative classes of text documents is not easy for a classic binary text classifier [8, 9].

The solution of classification issues using SVM, which was proposed by Vapinik in 1995, has gained increasing recognition and popularity among researchers due to its ability to handle high dimensional data such as textual documents [10, 11]. SVM performs classification by finding a decision boundary (separating hyperplane) that partitions the feature space into two distinct classes of data, positive and negative, with the maximum margin and represents the decision boundary using a set of support vectors (SV) generated from the training dataset [12, 13]. However, it is difficult for an SVM classifier to deal with non-separable data because the margin between positive and negative objectives is still unclear. In such situations, due to the uncertainty, an SVM classifier might not be completely effective in providing the optimal classification.

In practical problems, most training datasets include uncertainties. With an uncertain boundary, the learning classifier is more complex and difficult to find the optimal line to classify related objects and a full separation of relevant and irrelevant documents would require a curve. However, it is not easy to achieve the curve in a direct way with high precision because it requires too much computation [8]. Even if this were possible, there is no guarantee that it can be applied to completely classify all unknown testing samples because of the differences between training and testing document sets [9]. Thus, a nonlinear classifier is inefficient for a prediction task where an uncertain boundary exists in the training set. It is, therefore, desirable to design a classifier model able to linearly cope with non-separable data. Therefore, how to cope with data having uncertainties into the learning phase to improve the performance of binary classifier is a challenging problem.

This paper aims to present an effective binary classification model, called the Multiple-SVMs with Sliding Window model (MSVMs-SW model), in order to overcome the limitations in the existing classifiers and achieve the best performance in linear SVM for data having uncertainties. Different from traditional binary classifiers, the MSVMs-SW model aims to understand uncertainty by partitioning training samples (with two labels) into three regions, namely, positive, boundary, and negative regions in order to understand the decision boundary. Allowing this partitioning of the training set can help to describe relevant and non-relevant information and support to design a multiple-SVMs based classifier. We developed three different SVM classifiers ($\mathrm{SVM_P}$, $\mathrm{SVM_N}$, and $\mathrm{SVM_B}$), each of which is trained using its own training set that is derived by using the three regions. The training set for each classifier was different in order to obtain a greater improvement of the prediction results, to increase the certainty of all objects in positive and negative regions and to resolve the uncertainty in boundary region. The main motivation for using multiple-SVMs to classify new incoming documents is that a problem which requires expert knowledge will be better solved by a committee of experts rather than by a single expert [6]. Therefore, this research made *three innovative contributions* to the fields of text classification: (a) A new and effective model that deals with the uncertain decision boundary for text classification.

Our proposed model uses a training set with only minimal experimental parameters to identify the uncertain boundary, which makes it efficient; (b) An alternative solution for the hard uncertain boundary problem that was traditionally solved by non-linear SVMs; (c) A structure to guide the design of a fusion of multiple classifiers. To measure the effectiveness of the proposed model, extensive experiments were conducted, based on the RCV1 dataset and TREC assessors' relevance judgements. The results show a significant improvement on F_1 and *Accuracy* in the performance of binary text classification.

2 Related Work

Automated binary text classification is a significant research problem in information filtering and information organization fields [15]. It provides a way to determine a decision boundary that classifies textual documents into two distinct classes: relevant or irrelevant. Several approaches to binary text categorization, such as NB, KNN, decision tree, Rocchio, and SVM, have been developed to identify an efficient way to separate all related documents from a large dataset to determine a clear boundary between the classes in the text dataset [1]. However, in practice, the decision boundary includes much uncertainty because of the limitation of traditional machine learning algorithms, the presence of noise in text documents and feature scalability [16, 17].

SVM represents the training dataset as vectors, where each vector comprised of its words with their frequencies, and then tries to locate the linear hyperplane which separates two classes [13]. SVM can solve linear and nonlinear classifications and works well when applied to many practical problems [18, 19]. Although nonlinear SVM is effective when classifying nonlinear data, it has much higher computational complexity than linear SVM when making predictions for sparse data [19]. In addition, linear SVM performs better than nonlinear SVM when the number of features is very high, for example, in document classification [20, 21]. Therefore, if the number of features is extremely large, it is better to select linear SVM, due to the difficulty in finding the optimal parameters of a classifier when using nonlinear SVM [22]. Because linear SVM still has no effective way to deal with the uncertain factors it is, therefore, desirable to have a classifier model with the efficiency of a linear classifier to deal with data having uncertainty. The linear SVM is chosen in this study due to its computational and algorithmic simplicity.

The above limitations can be alleviated by employing the SW technique to divide the training set into three regions based on scores that present their degree of relevance and then to design a multiple-SVMs based classifier in order to derive a linear decision boundary for each classifier. In our proposed model the SW technique can be optimized by using Entropy. The entropy measurement is chosen in this research because it is a commonly understood measure in information theory and it is a fundamental measure for describing randomness and uncertainty of data [14, 23].

3 Description of MSVMs-SW Model

The MSVMs-SW model attempts to use the training dataset effectively to deal with the probable uncertainty and to improve the accuracy of the classifier. Our proposed model uses SVM as a high-performance classifier and generates new training set by dividing a universal set of documents into three disjointed parts (the positive region (POS), the boundary region (BND), and the negative region (NEG)). However, a single SVM may not be sufficient to classify all unknown testing samples. Therefore, we propose to use a multiple-SVMs based classifier. The proposed model contains two stages, a training stage and a testing stage, as shown in Fig. 1.

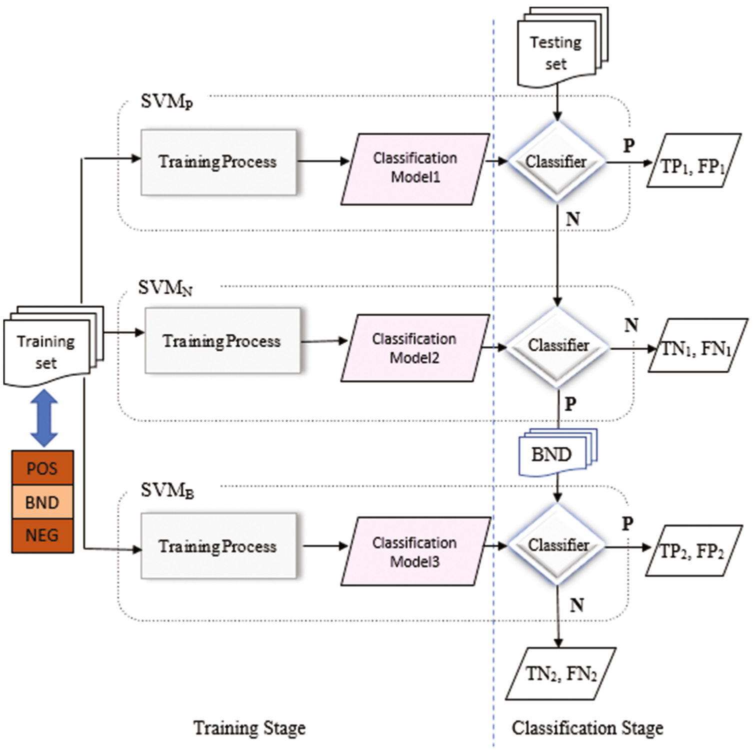


Fig. 1. Architecture of a multiple SVMs classifier

3.1 Training Stage of MSVMs-SW Model

To achieve the best performance in binary classification, the objective is to determine a decision boundary between classes. Our proposed model uses the training set only to set the decision boundary and to explore the uncertainty situation as shown in Fig. 2. It starts with the calculation of the score of training documents, and further regroups the training samples into three regions using the SW technique.

Document Scoring. Scoring documents to indicate their importance is an effective way for ranking relevant information. For a collection of documents in the datasets consisting of two sets (positive document sets, D^{+}; and negative document sets, D^{-}), the MSVMs-SW model calculates the weight of terms extracted from D^{+} and ranks them to use the *top-k* features based on their values, for example, $T = \{t_1, t_2, t_3, \ldots, t_k\}$. However, identifying the value of *k* is experimental. In our proposed model, we use the Okapi BM25 as a term weighting function. BM25 is a probabilistic state-of-the-art retrieval model [24], which can be calculated as follows:

$$w(t) = \frac{tf.(k_1 + 1)}{k_1.\left((1-b) + b\frac{DL}{AVDL}\right) + tf} . \log \frac{\frac{(r+0.5)}{(n-r+0.5)}}{\frac{(R-r+0.5)}{(N-n-R+r+0.5)}} \tag{1}$$

where N is the total number of training documents; R is the number of relevant documents; n is the number of documents which contain the term t; r is the number of relevant documents which contain the term t; tf is the term frequency; DL and $AVDL$ are the document length and average document length, respectively; and k_1 and b are the tuning parameters.

The reason for using the BM25 to calculate term weight is that the BM25 is a probabilistic model and in binary text classification we deal with uncertain information [24]. Probability is the measure used to understand the uncertainty in the information. Therefore, probability theory is the best way to quantify uncertainties. Next, the weighted terms are used to calculate the scores for all training documents $d \in D$ as follows:

$$score(d) = \sum\nolimits_{\mathrm{t} \in \mathrm{T}} w(t) \cdot \tau(t, d) \tag{2}$$

where $w(t) = \mathrm{BM25}(t, D^{+})$; and $\tau\ (t, d) = 1$ if $t \in d$; otherwise $\tau\ (t, d) = 0$.

Once the scores of the documents are calculated, the documents are ranked in descending order based on their scores.

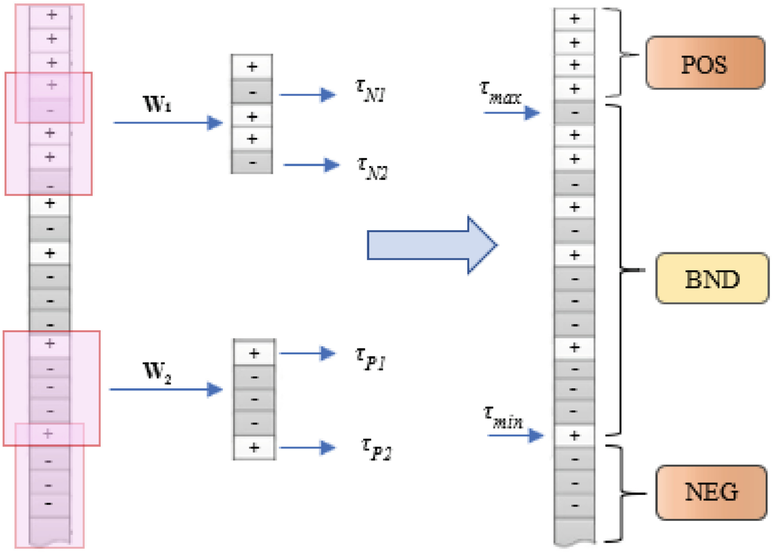
Sliding Window Technique. After ranking the training documents in the previous step, the most related documents will be located at the top of the list, while irrelevant ones will be located at the bottom of the ranked list, as shown in Fig. 2 (step 1). However, in most cases there are regions in which positive and negative documents are mixed due to the uncertain boundary. To find this area with many noisy documents, a sliding window technique and entropy are used to effectively determine the boundary region. Ko and Seo [25] used entropy and a sliding window to remove noisy data and solve the problem of the One-Against-All method. Our proposed model extends this idea to use a sliding window and entropy measurement to construct the decision boundary.

In this research, the sliding window was used to identify the boundary values which denote the region with the highest rate of noisy documents [25, 26]. The window size in this paper was set to 5 documents. The model starts to slide the window from the top documents in the ranked list, and then calculates the entropy value for the window. The window then slides over one document and yields a new entropy value. It continues to slide and stop when the entropy is greater than the threshold. We chose a high entropy threshold (95%). The same process applies from the bottom of the ranked list as shown in Fig. 2 (step 1).

Entropy Algorithm. Entropy is commonly used to define the uncertainty of variable [23, 26]. In this paper, for each sliding window(s), the entropy value can be calculated using the following function based on the number of positive and negative documents as follows:

$$E(s) = -\left[\frac{P}{P+N}\log_2\left(\frac{P}{P+N}\right) + \frac{N}{P+N}\log_2\left(\frac{N}{P+N}\right)\right] \tag{3}$$

where P and N are the numbers of positive and negative documents in SW, respectively.



Step 1: A sliding window technique over ranked documents

Step 2: Identify the boundary values and three regions.

Fig. 2. Decision Boundary Setting

Next, we select two windows with the greatest degree of entropy value. The first window (W_1) is from the top of the list and the second window (W_2) is from the bottom of the list. For W_1, the irrelevant documents are denoted as τ_N. For W_2, relevant documents are denoted as τ_P. In this study, the values of the boundary are calculated based on the scores of the relevant documents $\left(\tau_p\right)$ and the irrelevant documents (τ_N);

we selected the highest score of irrelevant documents in W_1 as a maximum threshold (τ_{max}), and the lowest score of relevant documents in W_2 as a minimum threshold (τ_{min}), as shown in Fig. 2 (step 2). Hence, the upper and lower decision boundary values τ_{max} and τ_{min} are calculated as follows:

$$\tau_{\max} = \max_{d_i \epsilon D^- \cap W_1}\{score(d_i)\} \tag{4}$$

$$\tau_{\min} = \min_{d_i \epsilon D^+ \cap W_2}\{score(d_i)\} \tag{5}$$

Three Regions for Partitioning the Training Set. The SVMs-SW model aims to group training sets into three regions rather than two classes. The training set *D* can be split into three regions based on the document scores and threshold settings in the previous step: the positive region (POS, possible relevant); the boundary region (BND, uncertain); and the negative region (NEG, possible irrelevant). The ranges of these regions are defined as follows:

$$\begin{aligned} POS &= \{d \in D | score(d) > \tau_{\max}\} \\ BND &= \{d \in D | \tau_{\min} \le score(d) \le \tau_{\max}\} \\ NEG &= \{d \in D | score(d) < \tau_{min}\} \end{aligned}$$

The boundary region *BND* contains many relevant and irrelevant documents under uncertain decisions which can be divided into two subsets: $B^+ = BND \cap D^+$ and $B^- = BND \cap D^-$.

Design Multiple SVMs Based Classifier. Building a classifier is achieved by training the SVM using chosen training documents via three regions. As shown on the left side of Fig. 1, we constructed three different SVMs classifiers; SVM_P, $\text{SVM}_\text{N,}$ and $\text{SVM}_\text{B.}$ To explain this process, the Algorithm 1 describes the training stage to learn the classifiers. The First classifier, SVM_P (step 8), takes strong positive documents *POS* and all negative documents $(B^- \cup NEG)$ as input, and uses the SVM classifier to build a predication model. The SVM_P generates the hyperplane between *POS* and $(B^- \cup NEG)$ to maintain the maximum margin between them. However, a potential problem with this approach can arise when the number of training samples in the *POS* part is very low and, in this case, the boundary of class would not be accurate due to insufficient positive training samples provided for text classification. To overcome this issue, we use a *pseudo feedback* technique. We selected the *top-k* scoring documents from the unlabeled testing set *U* and add them to the *POS* part as shown in step 1 to step 6. Different numbers of *top-k* have been tested and we found that using 5 documents improved the performance compared with using $k > 5$, which reduced the performance.

The second classifier, SVM_N, is constructed from the all positive documents $(POS \cup B^+)$ and strong negative documents *NEG*, as in step 9. For SVM_B, it is difficult to construct a classifier from the documents in the boundary region because SVM is very sensitive to noise, especially when noise is large and, in this case, the classifier will be very poor. Therefore, for even better classification we used the strong positive and negative samples (*POS, NEG*) to build SVM_B in our model, as in step 10.

```
Algorithm1: Multiple SVMs classifier Learning
  Input:   POS, NEG, BND; and parameter $k$;
           Unlabelled document in testing set, U;
           SVM classification model;
  Output:  SVM classification models, $SVM_P, SVM_N, SVM_B$;
    // Add top-k unlabelled documents to build first classifier
 1  let $n = |U|$
 2  for each $d \in U$ do
 3      $score(d) = \sum_{t \in T} weight(t). \tau(t, d)$
 4  end
 5  let $P = \{d_1, d_2, \ldots, d_n\}$ in descending ranking order,
 6  $D_P = \{d_i \mid d_i \in U, 1 < i \leq k\}$;
   // Learn training dataset using SVM classifier, get SVM models.
 7  $B^+ = BND \cap D^+, B^- = BND \cap D^-$;
 8  $SVM_P = Classifier_{SVM}(POS \cup D_P, B^- \cup NEG)$;
 9  $SVM_N = Classifier_{SVM}(POS \cup B^+, NEG)$;
10  $SVM_B = Classifier_{SVM}(POS, NEG)$;
```

3.2 Testing Stage of MSVMs-SW Model

In this phase, each stage has a different classification model, as shown on the right side of Fig. 1. The SVM_P classification model concentrates on identifying positive documents. In this stage, the documents that are classified as positive are denoted by TP_1 (true positive one) if they are true positive or grouped as FP_1 (false positive one) if they are actually negative. The objective of this stage is to achieve a high precision rate for positive documents and to minimize the *FP* rate, with an acceptable False Negative rate *FN*. The SVM_N classifier, which is generated in stage two, is applied to classify the documents that were predicted as negative in stage one. This stage focuses on increasing the precision rate for negative documents. In this stage, the documents that are classified as negative are denoted by TN_1 (true negative one) if they are negative or grouped into the FN_1 if they actually are positive. However, as the documents that were predicted as positive in this stage are still uncertain, the classifier will collect them into the boundary set *BND*. To classify these documents, we used the final classifier, SVM_B. This classifier can then assign those documents as positive or negative and produce four outputs, namely, TP_2, FP_2, TN_2, *and* FN_2. In our proposed classifier model, true positive $TP = TP_1 + TP_2$, false positive $FP = FP_1 + FP_2$, true negative $TN = TN_1 + TN_2$, and false negative $FN = FN_1 + FN_2$, as listed in Algorithm 2.

```
Algorithm 2: Multiple SVMs Classifier Testing
Input:    Incoming document without label for testing, $U$;
          SVM classification models, $SVM_P$, $SVM_N$, $SVM_B$;
Output:   Positive documents $POS$, and negative documents $NEG$;
$POS := NEG := BND := \emptyset$;
for each $d \in U$ do
    // Predict label of new documents($d_{unlabeled}$)using $SVM_P$.
    $d_{labeled} = SVM_P\ (d_{unlabeled}, Model_1)$;
    // If $SVM_P$ label it as positive, the label of document is
       positive
    if $d_{labeled}$ is positive then
        $POS = POS \cup \{d\}$;
    else
        $d_{labeled} = SVM_N\ (d_{unlabeled}, Model_2)$;
        // If $SVM_N$ label it as negative, the label of document
           is negative
        if $d_{labeled}$ is negative then
            $NEG = NEG \cup \{d\}$;
        else
            $BND = BND \cup \{d\}$;
    end
end
    // Predict label the rest documents in $BND$ using $SVM_B$.
for each $d \in BND$ do
    $d_{labeled} = SVM_B\ (d_{unlabeled}, Model_3)$
    if $d_{labeled}$ is positive then
        $BND = BND - \{d\}$; $POS = POS \cup \{d\}$;
    else
        $BND = BND - \{d\}$; $NEG= NEG \cup \{d\}$;
end
```

4 Experiments and Evaluation

4.1 Dataset and Evaluation Metrics

To evaluate the performance of our proposed model, the RCV1 dataset, which consists of 100 topics, was used. Each topic has been divided into training and testing sets with relevance judgements. The RCV1 corps has more than 804,000 documents which are news stories in English published by Reuters journalists [27]. These documents are grouped into 100 collection with 100 different topics. However, in our experiments in this study, we used the first 50 topics where the experiments are more reliable.

Three evaluation metrics were used to measure the effectiveness of the MSVMs-SW model and the baselines. The measures are the *F_1*-score and *Accuracy*. These evaluation metrics are widely used in text classification research. For more details of these measures refer to [6]. We also used the t-test *p-values* to analyse the significance of the difference between the results of the MSVMs-SW model and the baselines.

4.2 Baseline Models and Settings

In order to make an extensive evaluation, we compared our proposed model with six different baseline models. These models are the state-of-the-art influential models, which include statistical methods *libSVM,* SVMperf [28], J48 [29], NB [3], IBk (Instance-Based Learning), and Rocchio. All six models were trained and tested with the same dataset to conduct the experiments. They were also run with their best settings obtained through experimental practice. For *libSVM*, some default setting were utilized because the F_1-scores of the classifier are low when using the default setting. Different types of kernel functions and values of C were conducted, and we found that if we set $k = 0$ (linear kernel) and $C = 1$, we could get better results. In addition, we set $C = 10$ in SVMperf as it is the best value recommended in [9]. For our proposed model we used the linear kernel because it is quick and efficient with very large numbers of features as in document classification. For the experimental parameters of the BM25, k_1 and b values were set at 1.2 and 0.75, respectively.

4.3 Experimental Results

The experimental results of the MSVMs-SW and the baseline models are presented in Table 1. These results are the average of the 50 collections of the RCV1 dataset. The comparison between the proposed model, MSVMs-SW model, and other six baseline models was completed using two measures, *F_1* and *Accuracy.* The results in Table 1 have been categorized into two groups. The first group includes two *SVM* models (*libSVM* and SVMperf); the second group includes a popular influential classifier.

Table 1 shows that our proposed model outperformed all baseline models for text classification. Compared to the SVM models, the MSVMs-SW was significantly better

Table 1. Evaluation results of our model compared with the baselines.

Models	F_1	Accuracy
MSVMs-SW model	**0.4157**	**0.8621**
libSVM	0.3271	0.8557
SVMperf	0.2864	0.8001
improvement%	**+36.1%**	**+4.3%**
J48	0.3449	0.8263
Naïve Bayes	0.1851	0.8131
IBk	0.2970	0.8404
Rocchio	0.3681	0.5646
improvement%	**+49.5%**	**+16.4%**

on average with a minimum improvement of 4.3% and a maximum improvement of 36.1%. Compared to the IBK model, which has the highest *Accuracy* value in the second group, F_1 and *Accuracy* of the MSVM-SW model were significantly improved by 40.1% and 2.6%, respectively.

The t-test *p-values* evaluation in Table 2 also indicated that the proposed model is extremely statistically significant with a $p\text{–}value < 0.05$, compared with other baseline models on both F_1 and *Accuracy* for both one-tail and two-tails.

In order to test the effectiveness of using multiple-SVMs in our proposed model, we performed the same experiments with a single SVM classifier which used the original training set. The aims of using multiple-SVMs was to provide a way to make the decision boundary better. Therefore, we tried to separate the uncertain boundary to identify a clear boundary for both relevant and irrelevant parts. Table 3 shows the results of the performance of a single SVM classifier and multiple-SVMs on the RCV1 dataset. We used the *precision*, F_1-measure and *Accuracy* as measures for comparison.

Table 2. The p-values (one/two-tails) of the baseline models in comparison with MSVMs-SW model on RCV1.

No	Models	Tail(s)	F_1	Accuracy
1	libSVM	one	0.001908	0.337803
		two	0.0038162	0.675605
2	SVMperf	one	1.96E-05	2.42E-07
		two	3.91E-05	4.85E-07
3	J48	one	1.08E-14	1.24E-16
		two	2.15E-14	2.49E-16
4	Naïve Bayes	one	4.84E-09	0.000343
		two	9.67E-09	0.000686
5	IBk	one	7.85E-05	0.003296
		two	0.000157	0.006591
6	Rocchio	one	0.041785	4.71E-15
		two	0.083571	9.42E-15

In Table 3 we found that using multiple-SVMs achieved an average increase of 30.4% for $F_{1.}$ When considering precision value, multiple-SVMs showed the best performance, especially for the relevant part (Precision$^+$) with 7.8% improvement on average. It is clear that using multiple-SVMs instead of a single one can lead to better classification and improve the overall accuracy with data having uncertainty.

Based on the results presented earlier, the MSVMs-SW model improved the binary classification with the highest score in both F_1 and *Accuracy* (and particularly in $F_{1)}$ that best expresses the real situation in text classification.

Table 3. Multiple-SVMs Results compared with single SVM on RCV1.

Models	*Precision*$^+$	*Precision*$^-$	F_1	*Accuracy*
MSVMs-SW model (multiple-SVMs)	**0.5407**	**0.8767**	**0.4157**	**0.8621**
Single SVM	0.5016	0.8623	0.3187	0.8543
Improvement %	+7.8%	+2.0%	+30.4%	+1.0%

5 Conclusion

The MSVMs-SW model was proposed to deal with data having uncertainty, a situation in which is difficult to obtain good results when using non-linear SVM. This model uses the training set effectively to achieve super machine learning with high classification accuracy and to improve the performance of binary text classification. It tries to understand uncertainty by dividing the training set into three regions, namely, positive, negative, and boundary, in order to improve the certainty of both relevant and irrelevant parts and to reduce the impact of uncertainty in the boundary part. The partition of training sets was achieved by applying an effective SW technique and threshold setting and then organizing training samples to generate new training sets. After the boundary region was identified, we used multiple-SVMs instead of a single one to learn the classifiers and to classify new incoming documents. The experimental results using the standard RCV1 dataset show that the proposed model achieved significant improvements in F_1 and *Accuracy*, especially F_1, and outperforms existing classifiers, including state-of-the-art classifiers.

References

1. Lata, S., Loar, M.R.: Text clustering and classification techniques using data mining. Int. J. on Futur. Revolut. Comput. Sci. Commun. Eng. **4**(4), 859–864 (2018)
2. Joachims, T.: Transductive inference for text classification using support vector machines. In: ICML 1999, pp. 200–209. ACM, San Francisco (1999)
3. John, G.H., Langley, P.: Estimating continuous distributions in Bayesian classifiers. In: UAI 1995, pp. 338–345. ACM, Canada (1995)
4. Aggarwal, C.C., Zhai, C.: A Survey of Text Classification Algorithms. In: Mining Text Data, pp. 163–222. Springer, Boston, (2012). https://doi.org/10.1007/978-1-4614-3223-4_6
5. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theory **13**(1), 21–27 (1967)
6. Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surv. **34** (1), 1–47 (2002)
7. Forman, G.: An extensive empirical study of feature selection metrics for text classification. J. Mach. Learn. Res. **3**, 1289–1305 (2003)
8. Zhang, L., Li, Y., Bijaksana, M.A.: Decreasing uncertainty for improvement of relevancy prediction. In: Proceeding of the Twelfth Australasian Data Mining Conference, AusDM 2014, Brisbane, pp. 157–162 (2014)
9. Li, Y., Zhang, L., Yue, X., Yiyu, Y., Raymond, L., Yutong, W.: Enhancing binary classification by modeling uncertain boundary in three-way decisions. IEEE Trans. Knowl. Data Eng. **29**(7), 1438–1451 (2017)

10. Wardaya, P.D.: Support vector machine as a binary classifier for automated object detection in remotely sensed data. In: IOP Conference Series: Earth and Environmental Science, vol. 18, no. 1. IOP Publishing, Bristol (2014)
11. Wei, L., Wei, B., Wang, B.: Text classification using support vector machine with mixture of Kernel. J. Softw. Eng. Appl. **5**(12), 55–58 (2012)
12. Joachims, T.: Text categorization with support vector machines: learning with many relevant features. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0026683
13. Burges, C.J.: A tutorial on support vector machines for pattern recognition. Data Min. Knowl. Disc. **2**(2), 121–167 (1998)
14. Shannon, M.: Forensic relative strength scoring: ASCII and entropy scoring. Int. J. Digit. Evid. **2**(4), 1–19 (2004)
15. Lau, R.Y., Bruza, P.D., Song, D.: Towards a belief-revision-based adaptive and context-sensitive information retrieval system. ACM Trans. Inf. Syst. (TOIS). **26**(2), 1–38 (2008)
16. Bekkerman, R., Gavish, M.: High-precision phrase-based document classification on a modern scale. In: KDD 2011, pp. 231–239. ACM, San Diego (2011)
17. Li, Y., Algarni, A., Zhong, N.: Mining positive and negative patterns for relevance feature discovery. In: KDD 2010, pp. 753–762. ACM, New York (2010)
18. Fu, Z., Robles-Kelly, A., Zhou, J.: Mixing linear SVMs for nonlinear classification. IEEE Trans. Neural Netw. **21**(12), 1963–1975 (2010)
19. Rodriguez-Lujan, I., Cruz, C.S., Huerta, R.: Hierarchical linear support vector machine. Pattern Recogn. **45**(12), 4414–4427 (2012)
20. Gao, Y., Sun, S.: An empirical evaluation of linear and nonlinear kernels for text classification using support vector machines. In: FSKD 2010, pp. 1502–1505. IEEE, Yantai (2010)
21. Lan, M., Tan, C.L., Low, H.B.: Proposing a new term weighting scheme for text categorization. In: AAAI 2006, pp. 763–768. ACM, Boston (2006)
22. Hsu, C.W., Chang, C.C., Lin, C.J.: A practical guide to support vector classification. Technical Report, Department of Computer Science, National Taiwan University, Taipei (2003)
23. Du, L., Song, Q., Jia, X.: Detecting concept drift: an information entropy based method using an adaptive sliding window. Intell. Data Anal. **18**(3), 337–364 (2014)
24. Robertson, S., Zaragoza, H.: The Probabilistic Relevance Framework: BM25 and Beyond. Now Publishers Inc., Breda (2009)
25. Ko, Y.J., Seo, J.Y.: Issues and empirical results for improving text classification. J. Comput. Sci. Eng. **5**(2), 150–160 (2011)
26. Hall, G.A.: Sliding window measurement for file type identification. Technical Report, ManTech Security and Mission Assurance (2006)
27. Lewis, D.D., Yang, Y., Rose, T.G., Li, F.: Rcv1: a new benchmark collection for text categorization research. J. Mach. Learn. Res. **5**, 361–397 (2004)
28. Joachims, T.: A support vector method for multivariate performance measures. In: ICML 2005, pp. 377–384. ACM, Germany (2005)
29. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Francisco (1993)

High-Dimensional Limited-Sample Biomedical Data Classification Using Variational Autoencoder

Mohammad Sultan Mahmud(✉), Xianghua Fu,
Joshua Zhexue Huang, and Md. Abdul Masud

Big Data Institute, College of Computer Science and Software Engineering,
Shenzhen University, Shenzhen 518060, People's Republic of China
{sultan, fuxh, zx.huang, masud}@suz.eud.cn

Abstract. Biomedical prediction is vital to the modern scientific view of life, but it is a challenging task due to high-dimensionality, limited-sample size (also known as HDLSS problem), non-linearity, and data types tend are complex. A large number of dimensionality reduction techniques developed, but, unfortunately, not efficient with small-sample (observation) size dataset. To overcome the pitfalls of the sample-size and dimensionality this study employed variational autoencoder (VAE), which is a powerful framework for unsupervised learning in recent years. The aim of this study is to investigate a reliable biomedical diagnosis method for HDLSS dataset with minimal error. Hence, to evaluate the strength of the proposed model six genomic microarray datasets from Kent Ridge Repository were applied. In the experiment, several choices of dimensions were selected for data preprocessing. Moreover, to find a stable and suitable classifier, different popular classifiers were applied. The experimental results found that the VAE can provide superior performance compared to the traditional methods such as PCA, fastICA, FA, NMF, and LDA.

Keywords: Variational autoencoder · High dimensional and small sample size dataset · Biomedical diagnosis · Computational biology

1 Introduction

Biomedical prediction is a vital research and application area. The purpose of prediction is to minimize the risk in decision-making. In the area of computational biology, genomic microarray data plays a crucial role to assess the pathological diagnosis and classification of diseases, but it is a challenging task due to the properties of gene expression data such as small sample, high dimensions (features), and data types tend are complex and may correspond to discrete sequence data [1]. There are many features of genomic microarray affecting the structure and function of the body. These might be difficult for doctors to diagnose quickly and accurately. Therefore, it is necessary to employ computational intelligence in diagnosis to assist doctors to diagnose faster with high accuracy. For this purpose, in the last few decades, computational intelligence techniques have been proposed and exploited for biomedical prediction.

R. Islam et al. (Eds.): AusDM 2018, CCIS 996, pp. 30–42, 2019.
https://doi.org/10.1007/978-981-13-6661-1_3

Several applications, especially in the area of biomedical the measurements tend to be very expensive; consequently, the number of samples is very limited (can be below 100) whereas several thousand of features (dimensions). These datasets are called high dimension low sample size (HDLSS) datasets; are characterized with a large number of features P and relatively small number of samples n, often written $P > > n$ [2]. These HDLSS problems create significant challenges for the development of computational science.

The accuracy of prediction tends to deteriorate in high-dimensions due to the curse of dimensionality [3]. Hence, dimension reduction is invaluable for the analysis of high-dimensional data and several methods have been proposed including principal components analysis (PCA), independent components analysis (ICA), feature analysis (FA), non-negative matrix factorization (NMF), and latent Dirichlet allocation (LDA).

However, existing dimensional reduction techniques are unable to cope with the nonlinear relationships of the data but also unable to perform well with HDLSS datasets [4, 5]. For that reason, there needs an investigation to find an effective method that can deal with HDLSS datasets and improvement of the accuracy. In recent years, variational autoencoder (VAE) has emerged as one of the most popular approaches to unsupervised learning of complicated distributions and successfully applied in the area of image processing, text mining, and computer vision which involve high-dimensional data. VAE reduces dimensions in a probabilistically way, theoretical foundations are solid.

The objective of this paper is to illustrate the value of dimensional reduction of HDLSS datasets. Moreover, the paper also demonstrates the potential difficulties and the over-fitting dangers of performing dimensional-reduction in small sample size situations. Several well-known dimension reduction techniques have been implemented and compared. Also, the effectiveness of the VAE is tested on HDLSS dataset and comparisons with various dimensions in the prediction shown.

2 Literature Review

Due to the curse of dimensionality, dimensionality reduction is often crucial. The complexity of many decision trees [8] and decision forest [9–12] algorithm is $\mathrm{O}(\mathrm{nm}^2)$, where n is the number of records and m is the number of attributes. Over the past decades, many dimensionality reduction techniques have been proposed. An interesting approach [13] automatically computes weights for attributes, where a weight zero means complete deletion of the attribute, a weight 1 means full consideration of the attribute and anything in between 0 and 1 means a weighted inclusion of the attribute. Most commonly used dimensionality reduction methods are principal component analysis (PCA), independent component analysis (ICA), factor analysis (FA), linear discriminant analysis, and nonnegative matrix factorization (NMF). PCA has been applied in classification and clustering of relevant genes expression microarray or RNA-sequence data [4, 5]. Moreover, a number of prior techniques also investigated to classification and clustering of genes expression data for disease diagnosis [5, 6].

Traditional PCA is a frequently used method for reducing data for visualization and clustering [7]. In the case where sample sizes are larger than features $(n > P)$, classical methods such as PCA, ICA, and FA are likely to perform well. PCA essentially depends on the empirical covariance matrix in which a large number of samples are necessary. This work considered the problem where P is much larger than n. Yeung and Ruzzo [4] explored PCA for clustering gene expression data, the experimental result showed that PCA not suitable for dimensionality reduction in $P > \; > n$ datasets. Moreover, PCA and ICA have a major disadvantage in that they assume data is linearly separable, but the linear model is not always reliable in capturing nonlinear relationships of real-world problems, especially with limited samples [14].

NMF is another efficient way of high-dimensional data analysis [15]. In the bioinformatics area, NMF has been used for microarray data and protein sequence analysis [16, 17]. In PCA the principal components are defined by the number of samples using the eigenvalue decomposition, while for NMF the number of learned basis experiments is not limited. It appears that NMF can derive more features than samples for further analysis, and this may be why it got higher clustering accuracy as shown in the experimental results [18]. PCA, ICA, and FA are deterministic while NMF is stochastic; so NMF appears to be more suitable for HDLSS data analysis than PCA, ICA, and FA.

In the recent years, a particular class of probabilistic graphical model called topic models is found be a useful tool for mining microarray data. Latent Dirichlet allocation (LDA) is one of the most popular topic models, applied to mitigate overfitting of high-dimensionality in various fields including biomedical science [19–22]. Deep learning is a competent way for nonlinear dimensionality reduction which provides an appealing framework for handling high-dimensional datasets. Deep learning techniques have been successfully applied to extract information from high-dimensional data [23, 24].

Dimensionality reduction is effective if the loss of information due to mapping to a lower-dimensional space is less than the gain from simplifying the problem. A further challenge is that high-dimensionality and limited-sample size both increase the risk of overfitting and decrease the accuracy of classification [25, 29]. It is essential to building a classification model with good generalization ability, expected that perform equally well on the training set and independent testing set.

3 Methodology

The detailed design of the diagnosis system consists of three major states (see Fig. 1): preprocessing, variational autoencoder (VAE) based dimensionality reduction, and classification. Input dataset is first divided into two sub-datasets: a training set and testing set. Then variational autoencoder is applied to select desirable encoded dimensions of attributes which reduces computational burden and enhances the performance of classification. Finally, for disease classification, on the obtained reduced dimensional data the classifier is applied.

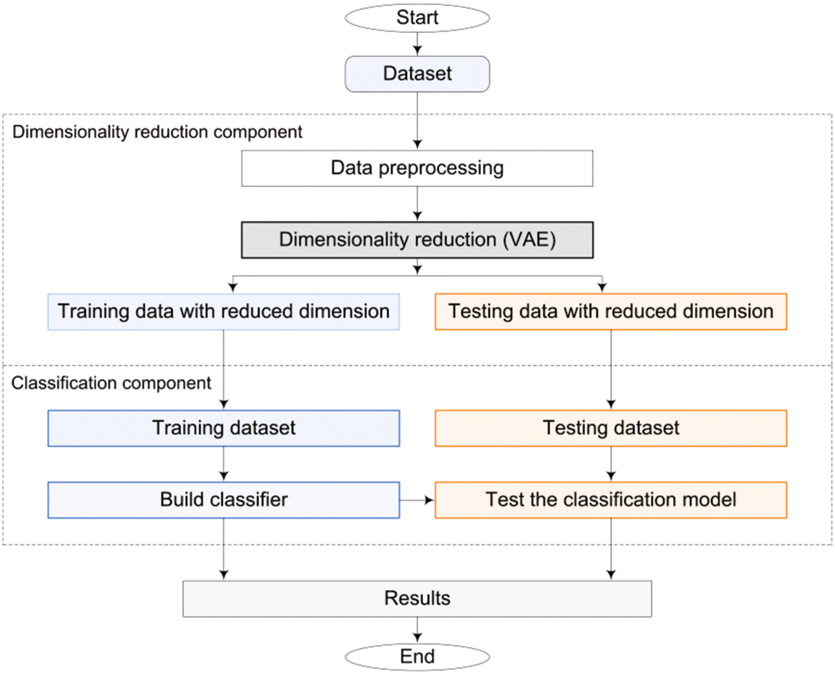

Start
Dataset
Dimensionality reduction component
Data preprocessing
Dimensionality reduction (VAE)
Training data with reduced dimension
Testing data with reduced dimension
Classification component
Training dataset
Testing dataset
Build classifier
Test the classification model
Results
End

Fig. 1. Diagnosis framework.

3.1 Data Preprocessing

Firstly, the dataset was divided into two parts: one for training and another for the testing, where 60% samples were used as training set, and remaining 40% were used as test set. More importantly, to overcome from overfitting and unbiased classification accuracy, separate datasets (training/testing) are used in the training and testing dimension reduction and classification. The training dataset used to train the dimension reduction technique and to build the classifier only same as the test dataset is used to test dimension reduction and classifier. Nonetheless, the experiment emphasis the importance of testing data unseen at any part of the dimensionality reduction and the classifier training.

3.2 Variational Autoencoder

Variational autoencoder (VAE) introduced by Kingma and Welling [26] and Rezende et al. [27], an exciting development in machine learning for combined generative modeling and inference. The main ideas of the VAE are comprising of a probabilistic model over data and a variational model over latent variables. VAE is rooted in Bayesian inference, i.e., it wants to model the underlying probability distribution of data so that it could sample new data from that distribution. A VAE consists of an encoder, a decoder, and a loss function. Figure 2 shows the basic structure of the variational autoencoder.

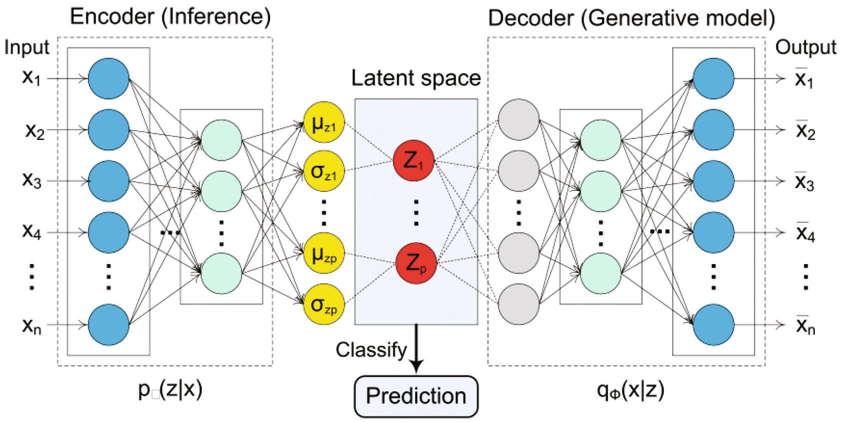


Fig. 2. Architecture of variation autoencoder based diagnosis.

Encoder. The encoder compresses data x into a latent variable z (lower-dimensional space). The lower dimensional space is stochastic, the encoder output parameter is $p_\theta(z|x)$, which is a Gaussian probability density. Data x can sample from this distribution to get noisy values of the representations z. θ is the weight and bias parameter.

Decoder. The decoder reconstructs the data is denoted by $q_\phi(z|x)$, gets input as the latent representation z and output the parameters of a probability distribution of the data. It goes from a smaller to a larger dimension. Information loss computed using the reconstruction log-likelihood, $\log q_\phi(z|x)$. This measure states how effectively the decoder has learned to reconstruct an input x given its latent representation z. ϕ is the weight and biases parameter.

Loss Function. The loss function of the VAE is the negative log-likelihood with a regularizer. Because there are no global representations that are shared by all data points, loss function can decompose into only terms that depend on a single data point l_i. The total loss is $\sum_{i=1}^{n} l_i$ for n total data points. The loss function l_i for data point x_i is:

$$l_i(\theta, \varphi) = -E_{z\sim p_\theta(z|x_i)}[\log q_\theta(x_i|z)] + KL(p_\theta(z|x_i)||p(z)) \tag{1}$$

In Eq. (1), the first term is reconstruction loss or expected negative log-likelihood of the ith data point. The second term is a regularizer, the Kullback-Leibler divergence between the encoder's distribution $p_\theta(z|x)$ and p(z). This divergence measures how close q and p. If the encoder outputs representations z are different than a standard normal distribution, it gets a penalty.

3.3 Classification

The classification algorithm is applied to the obtained reduced dimensional data. An ideal classifier is only a fiction. Since the classifier model is never a perfect classifier, a substitute is usually chosen from the area of machine learning. Classification algorithms can be grouped into the Bayesian classifier, functions, lazy algorithm, meta-algorithm, rules, and trees algorithm [28]. Some of the widely used classification algorithms are ANN, decision tree, KNN, logistic regression, Naïve Bayes, fuzzy logic, and SVM. Each classifier has own strength, but the challenge is an appropriate selection in the application of a complex and growing dataset.

4 Simulation Results and Discussion

4.1 Dataset

In this research, six well-known high-dimensional genomic microarray datasets from the Kent Ridge Biomedical Dataset Repository used. More details of datasets are shown in Table 1.

Table 1. Datasets at a glance.

Dataset	Genes	Samples	Classes	Authors
Breast cancer	24481	97	2	Van't Veer et al. (2002)
Central nervous system	7129	60	2	Pomeroy et al. (2002)
Colon tumor	6500	62	2	Alon et al. (1999)
Leukaemia	7129	72	2	Golub et al. (1999)
Lung cancer	12600	203	5	Bhattacharjee et al. (2001)
Ovarian cancer	15155	253	2	Pertricoin et al. (2002)

4.2 Experiment Setup

The experiments are setup to diagnose the diseases using genomic microarray data; six datasets are tested for different reduced dimensions e.g., 60, 100, 200, 300, 400, 500, and 600. To demonstrate the effectiveness of the VAE two kinds of comparisons are investigated in this research (1) single layer VAE, and (2) multiple layers VAE from two to four layers. Firstly, the dimensionality reduction technique is applied. For reduction methods performance evaluation, principal components analysis (PCA) [29], independent components analysis (fastICA) [30], factor analysis (FA) [31], latent Dirichlet allocation (LDA) [32], mini-batch dictionary learning (MBDL) [33], and non-negative matrix factorization (NMF) [34] results are compared. Then, the classification algorithm is applied to the obtained reduced dimensional data. Classification performance is compared to nine widely used classifiers namely AdaBoost (AB) [35], decision tree (DT) [36], Gaussian Naive Bayes (GNB) [37], Gaussian process (GP) [38], kNeighbors (KN) [39], logistic regression (LR) [40], multilayer perceptron (MLP) [41], random forest (RF) [42], and support vector classification (SVC) [43].

The experiments carried out in Python Tensorflow Keras framework. The configuration of the machine that was used to run these experiments was: Intel(R) Core i3-23S0M, CPU speed 2.30 GHz, RAM 4.00 GB, OS Windows 10 Pro 64-bit, x64-based processor.

4.3 Performance Measures

To evaluate the classification performances accuracy used in this research. Accuracy is the most used standard for evaluation classification techniques as well as for the comparison of performance defined by Eq. (2).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \tag{2}$$

where *TP* and *FP* are the number of true positive and false positive, respectively; *TN* and *FN* are the number of true and false negative, respectively.

4.4 Results Analysis and Discussions

Table 2 shows the classification accuracy of different classifiers by using the original features (all features) of the datasets. Average classification accuracies using different classifiers on a particular selected number of genes by several dimension reductions methods are shown in Figs. 3, 4, 5, 6, 7 and 8.

Table 2. Classification accuracy by using original features.

Dataset	AB	DT	GNB	GP	KN	LR	MLP	RF	SVC	Average
Breast cancer	0.51	0.51	0.33	0.64	0.59	0.72	0.46	0.56	0.69	0.56
CNS	0.81	0.70	0.66	0.62	0.63	0.66	0.48	0.54	0.70	0.64
Colon tumor	0.84	0.84	0.64	0.32	0.84	0.88	0.64	0.72	0.84	0.73
Leukaemia	0.89	0.89	0.96	0.68	0.89	0.89	0.72	0.68	0.86	0.83
Lung cancer	0.78	0.76	0.87	0.17	0.95	0.97	0.71	0.74	0.93	0.76
Ovarian cancer	0.96	0.94	0.89	0.90	0.94	0.99	0.57	0.73	1.00	0.88

AB: AdaBoost; DT: decision tree; GNB: Gaussian Naive Bayes; GP: Gaussian process; KN: kNeighbors; LR: logistic regression; MLP: multi-layer perceptron; RF: random forest; SVC: support vector classification.

In experiment 1, for the Breast cancer dataset train-test test, on the 60-dimension the average classification accuracy of PCA and VAE is 62% and 63% respectively, whereas in the case of using more dimension with VAE and multilayer VAE model provide better accuracy.

In experiment 2 for CNS dataset, the highest average accuracy of fastICA and 4-layer VAE is around 66% on the 60-dimension data. It is observed that as the dimension increased as the classification accuracy of LDA, MBDL, NMF, VAE and multi-layer VAEs has gained. Figure 4 shows that VAE and multi-layer VAEs performed better with high accuracy compare to other techniques. Moreover, NMF and LDA also obtained significantly better accuracy.

In experiment 3, for the Colon tumor dataset, this study's method VAE and multi-layer VAEs provides significantly better accuracy than other traditional algorithms. Figure 5 shows that the classification accuracy for VAE and multi-layer VAEs has small growth with dimension relatively large.

In another experiment the Lung cancer dataset has 203 samples, class distribution is **139**/17/**6**/21/20. Thus, it is an imbalanced dataset (139 and 6 sample in class one and three respectively). Figure 6 reveals a clear difference between the VAE and other methods. VAE and multi-layer VAEs perform significantly better accuracy compared to the other methods. The method of this study achieves a higher accuracy of 91% on the 300-dimension space, while multi-layer VAEs perform consistently better in all the different latent spaces (dimension).

The same is true for another two experiments for Leukemia and Ovarian dataset in Figs. 7 and 8 respectively. The results thus showcase the robustness of the proposed model compare to other methods. The experimental analysis found that the accuracy rate with application of 60 or 100 dimensions is comparatively smaller than 200 to 600 dimension and accuracy gained when dimension is relatively large.

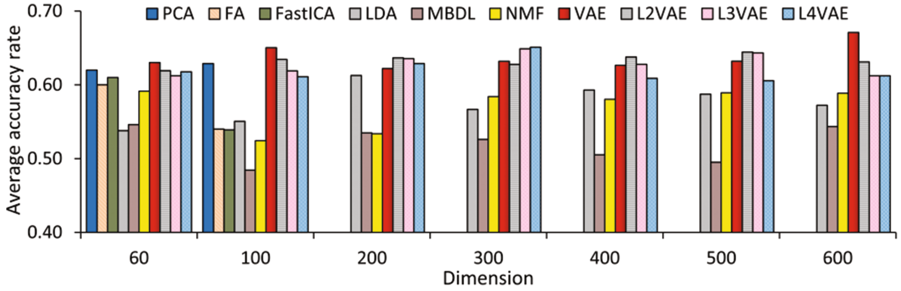


Fig. 3. Average accuracy of different methods in different dimension of Breast cancer dataset. (PCA: principal components analysis; FastICA: independent components analysis; FA: factor analysis; LDA: latent Dirichlet allocation; MBDL: mini-batch dictionary learning; VAE: single layer VAE; L2VAE: 2-layer VAE; L3VAE: 3-layer VAE; L4VAE: 4-layer VAE)

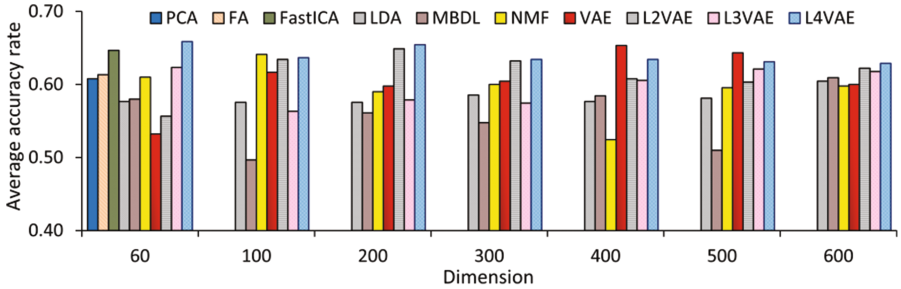


Fig. 4. Average accuracy of different methods in different dimension of CNS dataset. (PCA: principal components analysis; FastICA: independent components analysis; FA: factor analysis; LDA: latent Dirichlet allocation; MBDL: mini-batch dictionary learning; VAE: single layer VAE; L2VAE: 2-layer VAE; L3VAE: 3-layer VAE; L4VAE: 4-layer VAE)

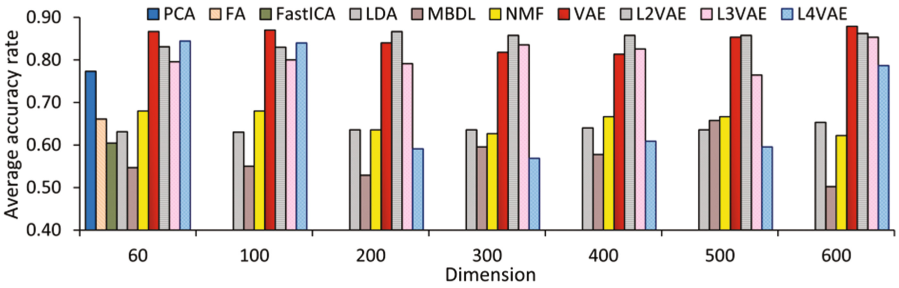


Fig. 5. Average accuracy of different methods in different dimension of Colon tumor dataset. (PCA: principal components analysis; FastICA: independent components analysis; FA: factor analysis; LDA: latent Dirichlet allocation; MBDL: mini-batch dictionary learning; VAE: single layer VAE; L2VAE: 2-layer VAE; L3VAE: 3-layer VAE; L4VAE: 4-layer VAE)

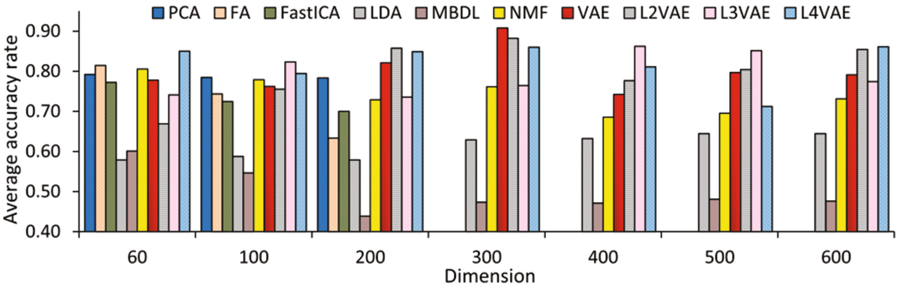


Fig. 6. Average accuracy of different methods in different dimension of Lung cancer dataset. (PCA: principal components analysis; FastICA: independent components analysis; FA: factor analysis; LDA: latent Dirichlet allocation; MBDL: mini-batch dictionary learning; VAE: single layer VAE; L2VAE: 2-layer VAE; L3VAE: 3-layer VAE; L4VAE: 4-layer VAE)

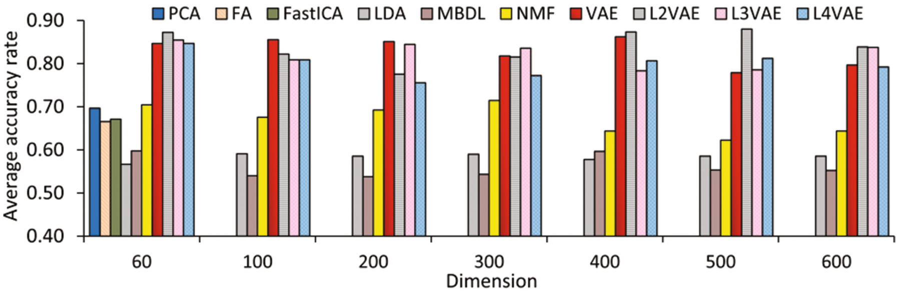


Fig. 7. Average accuracy of different methods in different dimension of Leukaemia dataset. (PCA: principal components analysis; FastICA: independent components analysis; FA: factor analysis; LDA: latent Dirichlet allocation; MBDL: mini-batch dictionary learning; VAE: single layer VAE; L2VAE: 2-layer VAE; L3VAE: 3-layer VAE; L4VAE: 4-layer VAE)

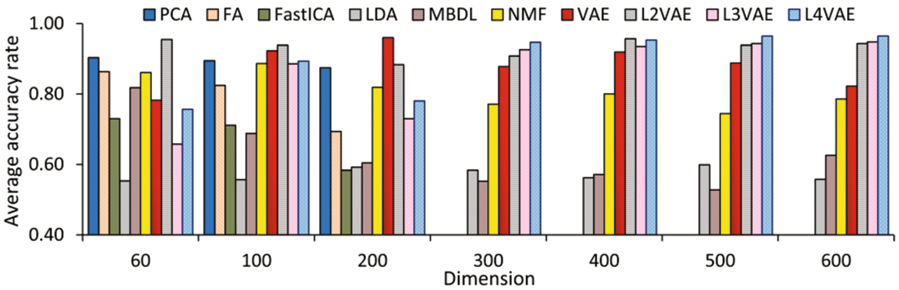


Fig. 8. Average accuracy of different methods in different dimension of Ovarian cancer dataset. (PCA: principal components analysis; FastICA: independent components analysis; FA: factor analysis; LDA: latent Dirichlet allocation; MBDL: mini-batch dictionary learning; VAE: single layer VAE; L2VAE: 2-layer VAE; L3VAE: 3-layer VAE; L4VAE: 4-layer VAE)

Figure 9 shows the loss curve for 100 epochs of the training and validation data of the single layer VAE of the different datasets. It is observed that the loss function usually converged after 150 iterations, here used 200 iterations. More iterations can be used, but there is a risk of overfitting.

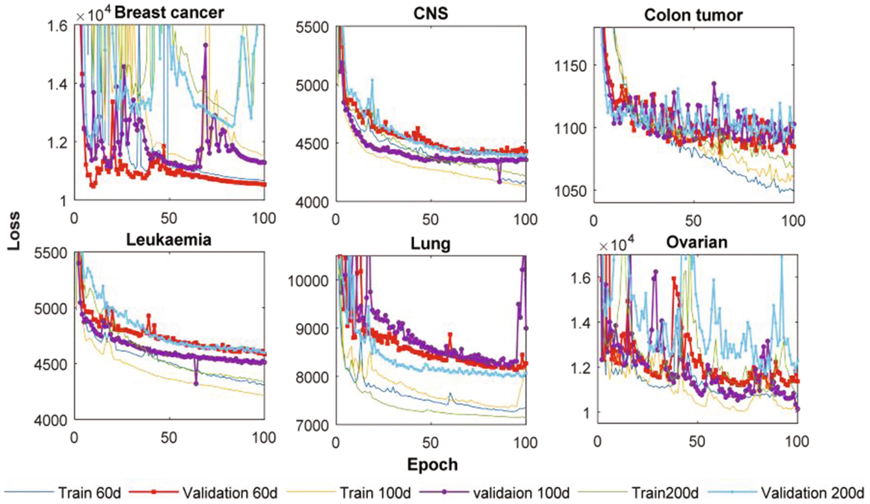


Fig. 9. Loss curve of the different dimensions for the training and validation data of the single layer VAE of different datasets.

5 Conclusion

This study presents a variational autoencoder based dimensionality reduction for high-dimensional small-sample biomedical diagnosis. In contrast to PCA, ICA, and FA, while using VAE can reduce the dimension as suitable from the high-dimensional dataset that enhances the performance of the classification system. Here, the authors have demonstrated the effectiveness of the proposed model by testing it on six gene expression microarray datasets. Moreover, compared with the traditional methods of PCA, fastICA, FA, MBDL, LDA, and NMF. The performance comparison of different reduction techniques to several popular classifiers in term of accuracy presented. The experimental results show that the proposed model performs superior to traditional methods. This reliable prediction will aid for biomedical diagnosis.

It is difficult to design an efficient prediction system for the small-sample-size dataset because limited-sample can easily contaminate the performance. A large test sample is required to evaluate a model accurately. Notably, the classification accuracy of small-sample-size gene expression microarray datasets are still poor; there needs further investigation to find an effective model to improve the accuracy.

Acknowledgments. This research is funded in part by the National Natural Science Foundations of China (Grant No. 61472258 and 61473194) and the Shenzhen-Hong Kong Technology Cooperation Foundation (Grant No. SGLH20161209101100926).

References

1. Clarke, R., et al.: The properties of high-dimensional data spaces: implications for exploring gene and protein expression data. Nat. Rev. Cancer **8**(1), 37–49 (2008)
2. Friedman, J., Hastie, T., Tibshirani, R.: The Elements of Statistical Learning. Springer Series in Statistics, 2nd edn. Springer, New York (2008). https://doi.org/10.1007/978-0-387-84858-7
3. Köppen, M.: The curse of dimensionality. In: 5th Online World Conference on Soft Computing in Industrial Applications (WSC5) (2000)
4. Yeung, K.Y., Ruzzo, W.L.: Principal component analysis for clustering gene expression data. Bioinformatics **17**(9), 763–774 (2001)
5. Dai, J.J., Lieu, L., Rocke, D.: Dimension reduction for classification with gene expression microarray data. Stat. Appl. Genet. Mol. Biol. **5**(1), 1–21 (2006)
6. Mishra, D., Dash, R., Rath, A.K., Acharya, M.: Feature selection in gene expression data using principal component analysis and rough set theory. Adv. Exp. Med. Biol. **696**, 91–100 (2011)
7. Jolliffe, I.: Principal Component Analysis, 2nd edn. Springer, New York (2002). https://doi.org/10.1007/b98835
8. Islam, M.Z.: EXPLORE: a novel decision tree classification algorithm. In: MacKinnon, L.M. (ed.) BNCOD 2010. LNCS, vol. 6121, pp. 55–71. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-25704-9_7
9. Islam, M.Z., Giggins, H.: Knowledge discovery through SysFor: a systematically developed forest of multiple decision trees. In: Proceedings of the Ninth Australasian Data Mining Conference (AusDM 2011), Ballarat, Australia. CRPIT, vol. 121 (2011)
10. Adnan, M.N., Islam, M.Z.: Forest PA: constructing a decision forest by penalizing attributes used in previous trees. Expert. Syst. Appl. (ESWA) **89**, 389–403 (2017)
11. Siers, M.J., Islam, M.Z.: Novel algorithms for cost-sensitive classification and knowledge discovery in class imbalanced datasets with an application to NASA software defects. Inf. Sci. **459**, 53–70 (2018)
12. Adnan, M.N., Islam, M.Z.: Optimizing the number of trees in a decision forest to discover a subforest with high ensemble accuracy using a genetic algorithm. Knowl. Based Syst. **110**, 86–97 (2016). ISSN 0219-1377

13. Rahman, M.A., Islam, M.Z.: AWST: A novel attribute weight selection technique for data clustering. In: Proceedings of the 13th Australasian Data Mining Conference (AusDM 2015) (2015)
14. Gupta, A., Wang, H., Ganapathiraju, M.: Learning structure in gene expression data using deep architectures with an application to gene clustering. In: IEEE International Conference on Bioinformatics and Biomedicine (BIBM) (2015)
15. Berry, M.W., Brown, M., Langville, A.N., Paucac, P., Plemmons, R.J.: Algorithms and applications for the nonnegative matrix factorization. Comput. Stat. Data Anal. **52**(1), 55–173 (2007)
16. Pascual-Montano, A., Carmona-Saez, P., Chagoyen, M., Tirado, F., Carazo, J.M., Pascual-Marqui, R.D.: bioNMF: a versatile tool for nonnegative matrix factorization in biology. BMC Bioinform. **7**, 366 (2006)
17. Gao, Y., Church, G.: Improving molecular cancer class discovery through sparse non-negative matrix factorization. Bioinformatics **21**(21), 3970–3975 (2005)
18. Liu, W., Kehong, Y., Datian, Y.: Reducing microarray data via nonnegative matrix factorization for visualization and clustering analysis. J. Biomed. Inform. **41**, 602–606 (2008)
19. Zhao, W., Zou, W., Chen, J.J.: Topic modeling for cluster analysis of large biological and medical datasets. BMC Bioinform. **15**, S11 (2014)
20. Lu, H.M., Wei, C.P., Hsiao, F.Y.: Modeling healthcare data using multiple-channel latent Dirichlet allocation. J. Biomed. Inform. **60**, 210–223 (2016)
21. Kho, S.J., Yalamanchili, H.B., Raymer, M.L., Sheth, A.P.: A novel approach for classifying gene expression data using topic modeling. In: Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics (2017)
22. Tan, J., Ung, M., Cheng, C., Greene, C.S.: Unsupervised feature construction and knowledge extraction from genome-wide assays of breast cancer with denoising autoencoders. Pac. Symp. Biocomput. **20**, 132–143 (2015)
23. Danaee, P., Ghaeini, R., Hendrix, D.A.: A deep learning approach for cancer detection and relevant gene identification. Pac. Symp. Biocomput. **22**, 219–229 (2017)
24. Smialowski, P., Frishman, D., Kramer, S.: Pitfalls of supervised feature selection. Bioinformatics **26**(3), 440–443 (2010)
25. Diciotti, S., Ciulli, S., Mascalchi, M., Giannelli, M., Toschi, N.: The ‘peeking’ effect in supervised feature selection on diffusion tensor imaging data. Am. J. Neuroradiol. **34**(9), E107 (2013)
26. Kingma, D.P., Welling, M.: Auto-encoding variational Bayes. In: International Conference on Learning Representations (2014)
27. Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models. In: Proceedings of the 31st International Conference on Machine Learning, vol. 32(2), pp. 1278–1286 (2014)
28. Witten, L.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
29. Tipping, M.E., Bishop, C.M.: Mixtures of probabilistic principal component analysers. Neural Comput. **11**(2), 443–482 (1999)
30. Hyvarinen, A., Oja, E.: Independent component analysis: algorithms and applications. Neural Netw. **13**(4–5), 411–430 (2000)

31. Barber, D.: Bayesian Reasoning and Machine Learning, Algorithm 21.1. Cambridge University Press, Cambridge (2012)
32. Hoffman, M.D., Blei, D.M., Bach, F.: Online learning for latent Dirichlet allocation. In: Proceedings of the 23rd International Conference on Neural Information Processing Systems, vol. 1, pp. 856–864 (2010)
33. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online dictionary learning for sparse coding. In: Proceedings of the 26th Annual International Conference on Machine Learning (2009)
34. Cichocki, A., Phan, A.H.: Fast local algorithms for large scale nonnegative matrix and tensor factorizations. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **92**(3), 708–721 (2009)
35. Zhu, J., Zou, H., Rosset, S., Hastie, T.: Multi-class AdaBoost. Stat. Interface **2**, 349–360 (2009)
36. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. CRC Press, Boca Raton (1984)
37. Manning, C.D., Raghavan, P., Schuetze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)
38. Rasmussen, C.E., Williams, C.: Gaussian Processes for Machine Learning. MIT Press, Cambridge (2006)
39. Altman, N.S.: An introduction to kernel and nearest-neighbor nonparametric regression. Am. Stat. **46**(3), 175–185 (1992)
40. Yu, H.F., Huang, F.L., Lin, C.J.: Dual coordinate descent methods for logistic regression and maximum entropy models. Mach. Learn. **85**(1–2), 41–75 (2011)
41. Hinton, G.E.: Connectionist learning procedures. Artif. Intell. **40**(1), 185–234 (1989)
42. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
43. Wu, T.F., Lin, C.J., Weng, R.C.: Probability estimates for multi-class classification by pairwise coupling. J. Mach. Learn. Res. **5**, 975–1005 (2004)

SPAARC: A Fast Decision Tree Algorithm

Darren Yates[1(✉)], Md Zahidul Islam[1], and Junbin Gao[2]

[1] Charles Sturt University, Panorama Avenue, Bathurst, NSW 2795, Australia
{dyates, zislam}@csu.edu.au
[2] University of Sydney Business School, The University of Sydney, Sydney, NSW 2006, Australia
junbin.gao@sydney.edu.au

Abstract. Decision trees are a popular method of data-mining and knowledge discovery, capable of extracting hidden information from datasets consisting of both nominal and numerical attributes. However, their need to test the suitability of every attribute at every tree node, in addition to testing every possible split-point for every numerical attribute can be expensive computationally, particularly for datasets with high dimensionality. This paper proposes a method for speeding up the decision tree induction process called SPAARC, consisting of two components to address these issues – sampling of the numeric attribute tree-node split-points and dynamically adjusting the node attribute selection space. Further, these methods can be applied to almost any decision tree algorithm. To confirm its validity, SPAARC has been tested and compared against an implementation of the CART algorithm using 18 freely-available datasets from the UCI data repository. Results from this testing indicate the two components of SPAARC combined have minimal effect on decision tree classification accuracy yet reduce model build times by as much as 69%.

Keywords: Decision tree · Processing speed · Classification accuracy · Node Attribute Sampling

1 Introduction

Classification is a long-studied method for data-mining and knowledge discovery, featuring in applications as varied as water dam management [1] and heart disease prediction [2]. It extracts information from a dataset of records as a set of rules or 'model' summarising the relationships between the feature values. Features are also known as 'attributes'. Moreover, the model learned can then be applied to a new previously-unseen record to predict the category or 'class' it belongs to. Decision trees are a popular classification technique due to their flowchart-like visualisation that is easy to follow and understand [3]. Popular examples include CART [4] and C4.5 [5].

A decision tree aims to discover relationships within a dataset $\boldsymbol{D}$ containing n records or 'instances' such that $\boldsymbol{D} = \{R_1, R_2, \ldots, R_n\}$ between an m-dimensional vector of non-class attributes $\boldsymbol{A} = \{A_1, A_2, \ldots, A_m\}$ and a class attribute, $\boldsymbol{C}$, consisting of p class values, with $\boldsymbol{C} = \{\mathrm{c}_1, \mathrm{c}_2, \ldots, \mathrm{c}_p\}$. If an attribute $\boldsymbol{A_i}$ is numerical, its value can range between $\boldsymbol{A_i} = [\boldsymbol{L_i}, \boldsymbol{U_i}]$, where $\boldsymbol{L_i}$ is the lower limit and $\boldsymbol{U_i}$ the upper limit of attribute $\boldsymbol{A_i}$. However, if an attribute $\boldsymbol{A_i}$ is categorical, the domain of $\boldsymbol{A_i} = \{a_1, a_2, \ldots$

R. Islam et al. (Eds.): AusDM 2018, CCIS 996, pp. 43–55, 2019.
https://doi.org/10.1007/978-981-13-6661-1_4

$a_x\}$ where $|\boldsymbol{A_i}| = x$, such that the attribute $\boldsymbol{A_i}$ has x possible values. Each record $\boldsymbol{R_L} \in \boldsymbol{D}$ will draw values for each attribute from that attribute's domain.

Data-mining has long been performed on computers, servers and 'cloud computing', but the growing capabilities of smartphones and 'Internet of Things' (IoT) microcontroller units (MCUs) has seen increasing attempts to implement data-mining algorithms on these constrained devices [6–9]. Improving algorithm efficiency would further their implementation on these devices, potentially offering new applications.

At the same time, moves by major cloud computing service providers such as Amazon and Google in late-2017 saw changes to service fees from a 'per hour' to a 'per second' basis [10]. This will likely renew corporate focus on algorithm efficiency, for if an algorithm can build a model on cloud computing with almost identical accuracy but in less time, the time saved may translate into direct cost savings.

This paper introduces a method for accelerating classification model build speed called SPAARC, consisting of two components – Node Attribute Sampling (NAS) and Split-Point Sampling (SPS). Experiments show SPAARC cuts model build times by as much as 69% with minimal loss in classification accuracy. This improvement could deliver cost savings for cloud-based data-mining or potentially boost implementation of locally-executed machine learning on performance-constrained devices.

While the concepts of NAS and SPS are not new, our novel contribution is the combination of these components into a single effective algorithm. Moreover, our NAS component incorporates a novel feature that aims to balance the disparate needs of classification accuracy and processing speed. This paper continues in Sect. 2 with a summary of previous research into algorithm speed optimisation. Section 3 details our proposed method, while Sect. 4 reports on its implementation and testing within the CART classification algorithm. Section 5 provides further analysis of the results and discussion before Sect. 6 concludes this paper.

2 Related Work

Decision tree induction speed optimisation can be traced back to Fayyad and Irani [11]. While the purpose of their research was primarily to provide supporting evidence for using entropy as a heuristic in decision tree induction, a noted 'side benefit' of their work was the improvement in algorithm speed.

A decision tree is a classification method using a tree-like structure to split a dataset's instances into subsets based on their attribute values. As shown in Fig. 1, a decision tree consists of nodes (denoted by ovals), each representing an attribute being tested; branches (edges), the possible outcomes for values of that attribute; and leaf nodes or 'leaves' (rectangles) holding class values that classify each instance [3]. At each node, all attributes are tested in turn to determine which one provides the most distinct partitioning of dataset records or 'information gain'. The test is referred to as the 'attribute selection measure'. There are many attribute selection measures, including Gain Ratio, as used by C4.5 [5], and Gini Index, implemented in CART [4].

Attribute selection measures typically handle attributes with categorical values differently to those with numerical values. Each distinct value of a categorical attribute becomes a possible outcome or 'split point', however, a numerical attribute will more

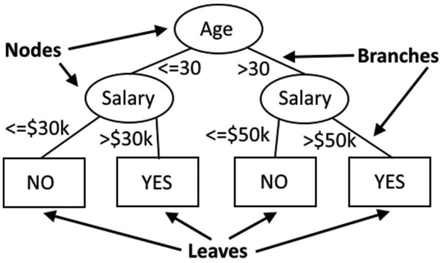


Fig. 1. A simple decision tree showing nodes (ovals), branches (edges) and leaves (rectangles).

commonly be split by two edges only. In this case, a numerical value $\boldsymbol{t}$ of attribute $\boldsymbol{A_i}$ is selected that best splits the instances of dataset $\boldsymbol{D}$ into distinct subsets such that one edge splits instances with values of $\boldsymbol{A_i} \leq \boldsymbol{t}$ and the other edge splits on values of $\boldsymbol{A_i} > \boldsymbol{t}$.

To select the value $\boldsymbol{t}$, the $\boldsymbol{j}$ distinct values of numeric attribute $\boldsymbol{A_i}$ are sorted in increasing order. The mid-point between each pair of adjacent values $\boldsymbol{j}$ and $\boldsymbol{j + 1}$ is tested for information gain. The mid-point of the adjacent pair showing the maximum gain is then held for that attribute as value $\boldsymbol{t}$. All $\boldsymbol{m}$ attributes are tested and attribute $\boldsymbol{A_k}$ is selected where $G(A_{\boldsymbol{k}}) > max_{i=1}^{m} G(\boldsymbol{A_i})$, $G(\boldsymbol{A_k})$ is the information gain of $\boldsymbol{A_k}$ and $1 \leq \boldsymbol{i} \leq \boldsymbol{m}$. If $\boldsymbol{A_k}$ is numerical, its recorded value $\boldsymbol{t}$ is used as the 'split point', as noted above. But as Fayyad and Irani indicate in [11], testing every adjacent pair of $\boldsymbol{j}$ distinct values of a numeric attribute implies $\boldsymbol{j{-}1}$ possible split-points. However, they found that if the sorted sequence of $\boldsymbol{j}$ values resulted in their corresponding class values grouping together neatly into $\boldsymbol{k}$ groups, only the split-points between those $\boldsymbol{k}$ groups need to be tested. This would require just $\boldsymbol{k{-}1}$ tests and since $\boldsymbol{k} \ll \boldsymbol{j}$, the computational workload would be significantly reduced. Nevertheless, if the sequence of class values is all jumbled, each adjacent pair of distinct numeric attribute values would need to be tested and the computational workload is back to $\boldsymbol{j{-}1}$ calculations.

However, in practice, Fayyad and Irani deemed this expanded testing would not be necessary and that only $\boldsymbol{k{-}1}$ testing points would be required, with those tests occurring at steps equal to $|\boldsymbol{C_i}|$, where $1 \leq \boldsymbol{i} \leq \boldsymbol{k}$ and $\boldsymbol{k}$ is the number of possible class values. Yet, while the authors empirically tested this concept, processing speed results were considered only as far as the split-point discretisation process – the effect on overall algorithm performance was a secondary consideration [11].

This method of reducing the number of possible split-points was furthered considered in [12] to improve support for very large datasets, resulting in the development of the CLOUDS algorithm. The technique implemented is described as 'Split-Point Sampling with Estimation' (SSE). It samples the $\boldsymbol{j{-}1}$ possible split-points by dividing them into intervals using an undefined 'similar frequency' method, estimating the optimum split-point within each interval via a hill-climbing algorithm; after all intervals were tested, the leading split-point was chosen for that attribute. While improving processing speed was not the main purpose of CLOUDS, further tests were carried out comparing SSE with a more basic split-point sampling technique. While the magnitude of improvement varied, results indicate testing fewer split point reduced the overall time required for the algorithm to build its model of the test dataset.

This sampling idea has also been applied to the attributes themselves. Rather than test every attribute at each node, 'feature subset selection' reduces the number of attributes. Decision tree induction assumes that each non-class attribute has a relationship with the class attribute. Thus, each non-class attribute is tested at each node point. However, using irrelevant attributes can reduce a tree's classification ability due to the unnecessary information or 'noise' these attributes can contain [13]. Removing the irrelevant attributes not only can improve overall classification accuracy, but also reduce the computational workload, since fewer attributes are tested.

Attribute subset selection has evolved with the many techniques grouped into three broad categories – wrapper, filter and embedded. Wrapper methods use the classifier itself to determine the most suitable subset of attributes. Yet, while accurate, wrapper methods can be computationally costly for datasets with high dimensionality and are considered a 'brute force' method of subset selection [14]. Filter methods, by contrast, select the attribute subset prior to tree induction through 'feature ranking'. Still, filtering can remove attributes that alone may not provide information, but in combination with other attributes hold knowledge that would otherwise be lost. Alternatively, embedded methods incorporate attribute selection within the decision tree algorithm, making them more efficient than wrapper methods. The CART algorithm is deemed to have an embedded mechanism for attribute selection [14], however, still tests all attributes at every node. The question we aim to answer in this paper is how to reduce the number of attributes required for testing in a way that provides meaningful processing speed gains whilst minimising any negative effect on classification accuracy.

3 Proposed Method

Our proposed method for accelerating tree induction involves combining components of split-point sampling and attribute subset selection into a single novel implementation we have named Split-Point And Attribute Reduced Classifier or SPAARC. Moreover, this method can be applied to any classification algorithm that implements its numeric attribute split-point analysis and node-attribute selection recursively.

The two specific components of SPAARC will now be detailed individually, starting with Node Attribute Sampling (NAS) covered in Sect. 3.1 and Split-Point Sampling (SPS) in Sect. 3.2. This will be followed by empirical evaluation in Sect. 4.

3.1 Node Attribute Sampling (NAS)

The NAS component in our proposed method avoids testing every non-class attribute at every tree node and further avoids the limitation of preselecting a subset of attributes before tree induction begins. Instead, our method dynamically selects the attribute space, switching between the full and subset attribute lists based on the tree depth level of the current node being tested. A simple example is shown in Fig. 2.

Induction begins at the root node, N1, which sits on Tree Depth Level (TDL) 1. Here, the full non-class attribute space $\boldsymbol{A} = \{\text{A1...A6}\}$ is always used to find the most

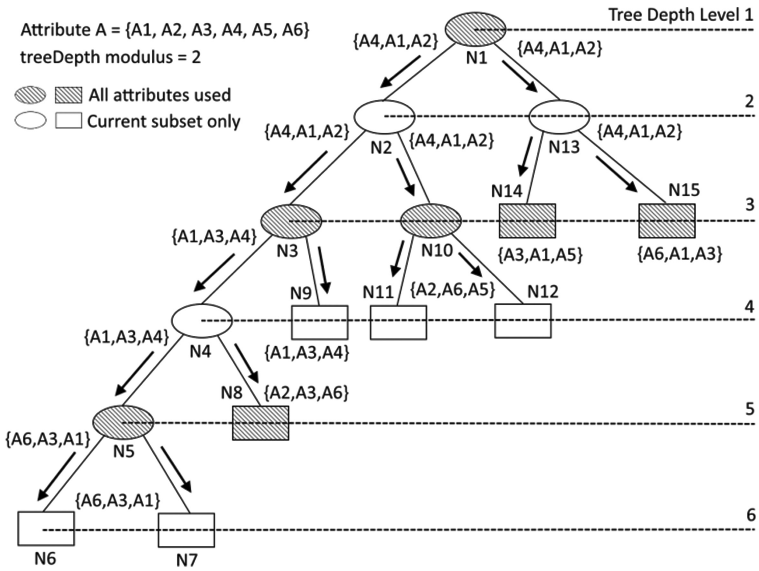
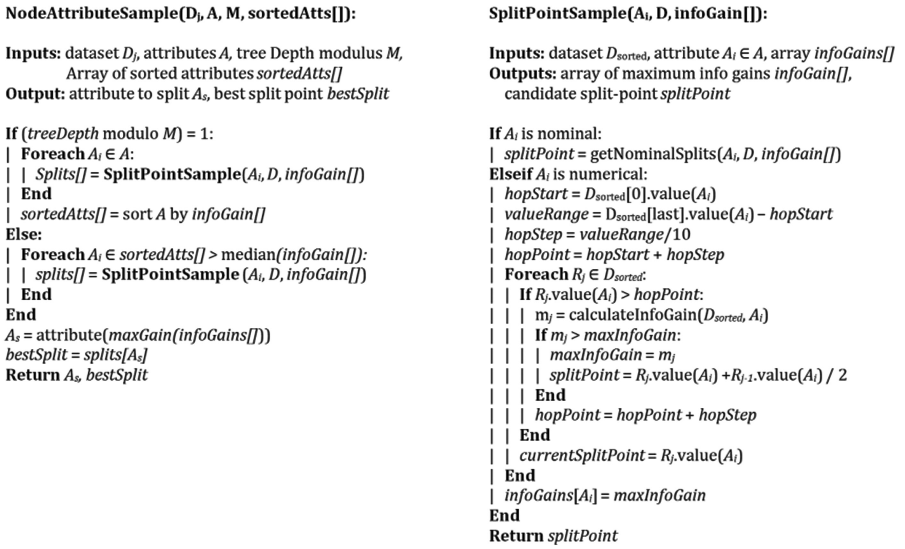


Fig. 2. NAS samples the full attribute space on each treeDepth modulus level (in this case, treeDepth modulus = 2)

appropriate attribute for the root node. As TDL 1 is also designated as a modulus level, the information gain scores from each non-class attribute tested for N1 are recorded. Also, the indices of these non-class attributes are sorted in decreasing order of their information gain, stored in array 'sortedAtts' and recursively passed onto the next node level. Now at node N2, the '***treeDepth***' value is set to 2. Since N2 is not on a modulus TDL, the node attribute is selected from a subset of sorted attributes received from node N1 with above-median information gain. In this example, that subset is {A4, A1, A2}. Tree induction continues recursively down the left-hand branch to node N3. As N3 sits on a modulus TDL, the attribute for this node again comes from the full attribute space, with the new sorted attribute list passed to the next level as before. This process continues down until leaf nodes N6 and N7 are set, using the current subset derived from N5 {A6, A3, A1}. Induction continues recursively to node N8. Again, as N8 is also on a modulus TDL, the full attribute space is used. Following this recursive path, the next node for processing is node N9. As N9 is not on a modulus TDL, its attribute is selected from the above-median subset passed down from N3 {A1, A3, A4}. Induction continues to N10, which being on a modulus TDL, selects from the full attribute space. Nodes N11 and N12 receive the new sorted list from N10, deriving the above-median subset {A2, A6, A5}. However, since the next node in the sequence, N13, also sits on a non-modulus TDL, it selects its attribute using the subset from N1 {A4, A1, A2}. The final two nodes, N14 and N15, both sit on a modulus TDL and choose an attribute from the full attribute space.

The pseudo-code for NAS appears in the *NodeAttributeSample* function in Fig. 3. It takes as parameters, the dataset D_j, attributes A along with the tree depth modulus factor M and an array of sorted attributes *sortedAtts*. It returns the split-attribute A_s and, if numeric, its best split point *bestSplit*. If the tree depth level modulus *M* is one (1), the function looks for the best split-point from every attribute, retaining the information gain (*infoGain*) factor from each. Following this, the attributes are sorted by information gain in decreasing order and stored in the *sortedAtts* array.



```
NodeAttributeSample(D_j, A, M, sortedAtts[]):

Inputs: dataset $D_j$, attributes $A$, tree Depth modulus $M$,
        Array of sorted attributes sortedAtts[]
Output: attribute to split $A_s$, best split point bestSplit

If (treeDepth modulo M) = 1:
| Foreach $A_i \in A$:
| | Splits[] = SplitPointSample($A_i$, D, infoGain[])
| End
| sortedAtts[] = sort A by infoGain[]
Else:
| Foreach $A_i \in$ sortedAtts[] > median(infoGain[]):
| | splits[] = SplitPointSample ($A_i$, D, infoGain[])
| End
End
$A_s$ = attribute(maxGain(infoGains[]))
bestSplit = splits[$A_s$]
Return $A_s$, bestSplit
```

```
SplitPointSample(A_i, D, infoGain[]):

Inputs: dataset $D_{sorted}$, attribute $A_i \in A$, array infoGains[]
Outputs: array of maximum info gains infoGain[],
  candidate split-point splitPoint

If $A_i$ is nominal:
| splitPoint = getNominalSplits($A_i$, D, infoGain[])
Elseif $A_i$ is numerical:
| hopStart = $D_{sorted}$[0].value($A_i$)
| valueRange = $D_{sorted}$[last].value($A_i$) – hopStart
| hopStep = valueRange/10
| hopPoint = hopStart + hopStep
| Foreach $R_j \in D_{sorted}$:
| | If $R_j$.value($A_i$) > hopPoint:
| | | $m_j$ = calculateInfoGain($D_{sorted}$, $A_i$)
| | | If $m_j$ > maxInfoGain:
| | | | maxInfoGain = $m_j$
| | | | splitPoint = $R_j$.value($A_i$) +$R_{j-1}$.value($A_i$) / 2
| | | End
| | | hopPoint = hopPoint + hopStep
| | End
| | currentSplitPoint = $R_j$.value($A_i$)
| End
| infoGains[$A_i$] = maxInfoGain
End
Return splitPoint
```

Fig. 3. Algorithms for NAS (NodeAttributeSample) and SPS (SplitPointSample)

If the tree depth level modulus *M* is not equal to one, each attribute of the *sortedAtts* subset with information gain greater than the median is tested for its split-point. In any case, the attribute with maximum information gain is returned as the split-attribute A_s, its split-point *bestSplit*, and the information gain values '*infoGains*'.

This *NodeAttributeSample* function also calls the '*SplitPointSample*' function, featuring the split-point sampling (SPS) component we shall now discuss in Sect. 3.2.

3.2 Split-Point Sampling (SPS)

As we have seen previously in Sect. 2, many decision tree algorithms test for the optimum split-point of a numerical attribute at a node using some measure of information gain. One example measure is the Gini index:

$$\mathrm{Gini}(R) = 1 - \sum\nolimits_{i=1}^{m} p_i^2 \tag{1}$$

where p_i is the probability that the record $\boldsymbol{R}$ in dataset $\boldsymbol{D}$ belongs to the class value $\boldsymbol{C_i}$. However, to obtain the split-point with maximum information gain, each adjacent pair of distinct attribute values must be tested, such that for $\boldsymbol{j}$ distinct values, there are generally $\boldsymbol{j-1}$ adjacent pairs tested. Our SPS component does not interfere with information gain measure itself, however, it dynamically reduces the number of possible split-points tested. This is done by dividing the range of distinct attribute values into equal-width intervals and using the adjacent pair of values at the edges of each interval as potential split points. Thus, if there are $\boldsymbol{k}$ intervals, only $\boldsymbol{k-1}$ test points are required. Through experimentation, a value of $\boldsymbol{k} = 20$ has been shown to provide good results. While the sampling of 20 interval points may not result in the selection of the split-point of maximum information gain, it can be shown that the point selected should never be more than a value distance of half the interval step from the ideal (Fig. 4).

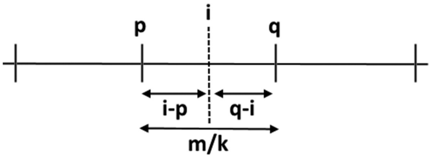


Fig. 4. Maximum distance the actual split-point i can be from a tested split-point is $m/2k$.

Theorem 1: *If* $\boldsymbol{m}$ = *range of numeric attribute values and* $\boldsymbol{k}$ = *the number of intervals, the actual split-point of maximum information gain cannot be more than* $\boldsymbol{m/2k}$ *away from an interval test point.*

Proof: Let $\boldsymbol{p}$ and $\boldsymbol{q}$ be two consecutive interval points, such that $\boldsymbol{q} - \boldsymbol{p} = \boldsymbol{m/k}$. Let $\boldsymbol{i}$ be the actual split point somewhere in the range sequence between points $\boldsymbol{p}$ and $\boldsymbol{q}$.

From this, there are three possibilities – (i) that $(\boldsymbol{i} - \boldsymbol{p}) < \boldsymbol{m/2k}$, (ii) that $(\boldsymbol{i} - \boldsymbol{p}) = \boldsymbol{m/2k}$, and (iii) that $(\boldsymbol{i} - \boldsymbol{p}) > \boldsymbol{m/2k}$. For possibilities (i) and (ii), the theorem is already proved, since neither is greater than $\boldsymbol{m/2k}$.

Now for possibility (iii), let $(\boldsymbol{i} - \boldsymbol{p}) > \boldsymbol{m/2k}$. Since from Fig. 3, $(\boldsymbol{i} - \boldsymbol{p}) + (\boldsymbol{q} - \boldsymbol{i}) = \boldsymbol{m/k}$, it must be that $(\boldsymbol{i} - \boldsymbol{p}) = \boldsymbol{m/k} - (\boldsymbol{q} - \boldsymbol{i})$. Substituting for $(\boldsymbol{i} - \boldsymbol{p})$, it must also follow that if $(\boldsymbol{i} - \boldsymbol{p}) > \boldsymbol{m/2k}$, then $\boldsymbol{m/k} - (\boldsymbol{q} - \boldsymbol{i}) > \boldsymbol{m/2k}$. It then follows that $\boldsymbol{m/k} - \boldsymbol{m/2k} > (\boldsymbol{q} - \boldsymbol{i})$ and $(\boldsymbol{q} - \boldsymbol{i}) < \boldsymbol{m/2k}$.

Choosing the optimum level of $\boldsymbol{k}$ is dependent upon the number of distinct values $\boldsymbol{j}$, for as $\boldsymbol{k}$ approaches $\boldsymbol{j}$, the difference in the number of calculations required, and thus the speed gain as a result, will be minimal. However, if $\boldsymbol{k}$ is too low, the distance between the optimal and selected split points will likely be greater, potentially negatively affecting the choice of attribute at each node and hence, overall accuracy.

The *SplitPointSample* pseudo-code in Fig. 3 details this function. It takes as parameters the subset of records for the current node, D_{sorted}, the current attribute, A_i. and the array of information gains *infoGains*. It returns '*infoGain*', plus the candidate split-point called '*splitPoint*'. The function first considers attribute A_i – if it is categorical, it is passed onto the tree algorithm's current function for finding categorical split-points. As SPS handles numerical attributes only, we skip this categorical task.

For numerical attributes, four values are initially calculated from the sorted range of record values for attribute A_i – *hopStart* is the A_i value of the first record, *valueRange* is the range of numeric values $\boldsymbol{m}$, *hopStep* is the range distance divided by the number of intervals, $\boldsymbol{k}$, set to 20 and *hopPoint* is the first interval, set to *hopStart* plus *hotStep*.

At this point, each record R_j in the current data subset D_{sorted} is considered for its value of attribute A_i, but unless this value is greater than the current interval value *hopPoint*, it is skipped. This skipping is where speed is gained. However, if this A_i value is greater, we calculate the information gain. We compare this information gain value, m_j, against the current maximum information gain value *maxInfoGain* – if value m_j is greater, it becomes the new maximum and the current best '*splitPoint*' is set between the A_i values of current record R_j and the previous record R_{j-1}. We increment the interval step *hopStep* to the next interval, proceed to the next record and continue until the last record is reached. On completion, the maximum information gain value for A_i is stored in array *infoGains* and its corresponding split-point is returned.

4 Experiments

To test the validity of SPAARC, experiments were carried out using 18 freely-available datasets from the University of California, Irvine (UCI) machine-learning repository [15]. Dataset details are shown in Table 1. These datasets were tested on the Weka version 3.8.2 data-mining core using the SimpleCART algorithm from the Weka Package Manager. Tests were conducted comparing the SimpleCART algorithm in its original form against the same algorithm augmented with our SPAARC method. Further tests involving the SPS and NAS components individually were conducted to identify the role each plays in the overall SPAARC result.

Table 1. Details of the 18 numeric, categorical and mixed datasets used in experiments

Dataset	Inst.	Attrs.	Type	Dataset	Inst.	Attrs.	Type
mfeat-fourier (FOU)	2000	77	Num.	KRKPA7 (KRK)	3196	37	Cat.
mfeat-zernike (ZER)	2000	48	Num.	Soybean (SOY)	683	36	Cat.
Waveform (WAV)	5000	41	Num.	SPECT (SPE)	287	23	Cat.
EEG eye state (EEG)	14980	15	Num.	CMC	1473	10	Mixed
Crowdsource map (CRO)	10545	29	Num.	CKD	400	25	Mixed
Shuttle (SHU)	43500	9	Num.	Abalone	4177	8	Mixed
Car Evaluation (CAR)	1728	7	Cat.	Annealing (ANN)	798	39	Mixed
Mushroom (MUS)	8124	23	Cat.	Hypothyroid (HYP)	3772	30	Mixed
Nursery	12960	9	Cat.	Arrhythmia (ARR)	452	280	Mixed

Classification accuracy was evaluated using ten-fold cross-validation, while tree induction or 'build' times were programmatically recorded to millisecond precision. Testing was done on an Intel Core i5-2300 PC with Windows 8.1 operating system.

Classification accuracy and model build times of our SPAARC algorithm compared with SimpleCART appear in Table 2 (leading results in all tests are shown in **bold**). Significantly, SPAARC produced faster tree build times than SimpleCART in 15 of the 18 datasets tested. The most successful results were achieved on the six numeric datasets where SPAARC matched or exceeded SimpleCART in all sets for classification accuracy and processing speed. The FOU dataset achieved a 69% time reduction.

Importantly, SPAARC achieved these faster speeds whilst still managing an 11/3/4 win/draw/loss record against SimpleCART for classification accuracy. This also led to SPAARC producing a marginally-higher overall classification accuracy average than SimpleCART (83.65 vs 83.47). The least successful group for SPAARC in terms of speed gains was the 'mixed' group, where three of the six datasets recorded slower build times, despite improving classification accuracy on four of those six. However, across all 18 datasets, SPAARC reduced the total build time by more than 35% (18.452 s vs 28.583).

Table 2. Classification accuracy and build times of SimpleCART and SPAARC algorithms.

Dataset	Type	Class'n accuracy (%)		Model build time (secs)	
		SimpleCART	SPAARC	SimpleCART	SPAARC
mfeat-fourier (FOU)	Num.	75.50	**75.65**	2.629	**0.811**
mfeat-zernike (ZER)	Num.	67.25	**69.95**	1.826	**0.689**
Waveform (WAV)	Num.	76.68	**77.04**	1.865	**1.150**
EEG eye state (EEG)	Num.	84.11	**84.38**	2.846	**2.472**
Crowdsource map (CRO)	Num.	90.09	90.09	4.505	**1.919**
Shuttle (SHU)	Num.	99.94	**99.96**	4.501	**2.808**
Car Evaluation (CAR)	Cat.	97.11	**97.69**	0.174	**0.146**
Mushroom (MUS)	Cat.	99.94	99.94	1.944	**1.584**
Nursery	Cat.	99.58	99.58	2.215	**1.798**
KRKPA7 (KRK)	Cat.	**99.37**	99.31	0.746	**0.632**
Soybean (SOY)	Cat.	**91.07**	90.63	0.545	**0.479**
SPECT Heart (SPE)	Cat.	80.90	**81.27**	0.039	**0.035**
CMC	Mixed	55.19	**55.53**	0.319	**0.280**
Chron. Kidney Dis. (CKD)	Mixed	97.50	**98.00**	**0.069**	0.072
Abalone	Mixed	26.07	**26.41**	2.344	**1.873**
Annealing (ANN)	Mixed	91.85	**92.11**	**0.198**	0.223
Hypothyroid (HYP)	Mixed	**99.55**	99.36	**0.743**	0.766
Arrhythmia (ARR)	Mixed	**70.80**	68.81	1.082	**0.715**
Average %/Total secs		83.47	**83.65**	28.583	**18.452**

To understand in more detail how each of the algorithm's two main components contribute to the overall result, SPAARC was modified to implement each component separately and compared in each case with the SimpleCART algorithm. The results in Table 3 show that SPS improved model build times in 13 of the 18 datasets tested. Worthy of note is the fact that the total build time achieved by SPS is greater than that recorded by SPAARC, indicating the SPAARC results are not due solely to SPS. SPS improved upon the SimpleCART classification accuracy results in nine of 18 datasets. However, the overall average classification accuracy for SPS was marginally better than SimpleCART largely due to one dataset gain (ZER). Nevertheless, SPS improved the cumulative model build time by just over eight seconds or 29.1%. Thus, SPS can accelerate tree induction and have minimal effect on classification accuracy.

Similarly, Table 4 shows the results of implementing the NAS component alone in comparison with SimpleCART. Here, model build times were improved by NAS in every datasets tested. While NAS achieved speed gains in more datasets than SPS, the gain in build time resulting from NAS was less than SPS (24.31 s vs SPS' 20.25). This indicates NAS with its treeDepth modulus setting of 2 contributes less to overall speed than SPS, but is more likely to show a speed gain on a dataset than SPS alone. The results also saw NAS improve classification accuracy in seven of 18 datasets, with a slightly higher average accuracy overall. More broadly, Tables 3 and 4 reveal the performance of SPAARC overall is due to both components combined.

5 Discussion and Further Research

Although every decision tree algorithm has its own structure, tree induction in general is a two-step process. First, the tree is grown until all branches are finished as leaves via a stopping criteria. Then, second, depending on the algorithm, the tree is reduced or

Table 3. SimpleCART classification accuracy and build times compared with SPS alone.

Dataset	Type	Class'n accuracy (%)		Model build time (secs)	
		SimpleCART	SPS	SimpleCART	SPS
mfeat-fourier (FOU)	Num.	75.50	**75.55**	2.623	**1.006**
mfeat-zernike (ZER)	Num.	67.25	**69.95**	1.826	**0.794**
Waveform (WAV)	Num.	76.68	**77.10**	1.865	**1.290**
EEG eye state (EEG)	Num.	84.11	84.05	2.846	**2.618**
Crowdsource map (CRO)	Num.	90.09	**90.25**	4.505	**2.144**
Shuttle (SHU)	Num.	99.94	**99.96**	4.501	**2.639**
Car Evaluation (CAR)	Cat.	97.11	97.11	0.174	0.174
Mushroom (MUS)	Cat.	99.94	99.94	1.944	**1.841**
Nursery	Cat.	99.58	99.58	2.215	**2.108**
KRKPA7 (KRK)	Cat.	99.37	99.37	0.746	**0.691**
Soybean (SOY)	Cat.	91.07	91.07	0.545	**0.541**
SPECT Heart (SPE)	Cat.	80.90	80.90	**0.039**	0.040
CMC	Mixed	55.19	**55.80**	0.319	**0.317**
Chron. Kidney Dis. (CKD)	Mixed	97.50	**98.00**	**0.069**	0.085
Abalone	Mixed	26.07	**27.17**	2.344	**1.994**
Annealing (ANN)	Mixed	91.85	**92.11**	**0.198**	0.231
Hypothyroid (HYP)	Mixed	**99.55**	99.36	**0.743**	0.908
Arrhythmia (ARR)	Mixed	**70.80**	68.81	1.082	**0.826**
Average %/Total secs		83.47	**83.65**	28.583	**20.247**

'pruned' to minimise the effects of noise in the training dataset and improve its generalisation ability on new unseen instances [16]. The SPAARC algorithm's two components only operate on the growth phase, leaving potential for further time savings through accelerating the pruning phase. This will be an area for future research.

Table 4. SimpleCART classification accuracy and build times compared with NAS alone.

Dataset	Type	Class'n accuracy (%)		Model build time (secs)	
		SimpleCART	NAS	SimpleCART	NAS
mfeat-fourier (FOU)	Num.	75.50	**75.90**	2.623	**2.108**
mfeat-zernike (ZER)	Num.	67.25	**68.35**	1.826	**1.499**
Waveform (WAV)	Num.	76.68	76.68	1.865	**1.560**
EEG eye state (EEG)	Num.	84.11	**84.41**	2.846	**2.586**
Crowdsource map (CRO)	Num.	90.09	**90.12**	4.505	**3.691**
Shuttle (SHU)	Num.	99.94	**99.95**	4.501	**3.876**
Car Evaluation (CAR)	Cat.	97.11	**97.69**	0.174	**0.169**
Mushroom (MUS)	Cat.	99.94	99.94	1.944	**1.630**
Nursery	Cat.	99.58	99.58	2.215	**1.815**
KRKPA7 (KRK)	Cat.	99.37	99.31	0.746	**0.643**
Soybean (SOY)	Cat.	**91.07**	90.63	0.545	**0.484**
SPECT Heart (SPE)	Cat.	80.90	**81.27**	0.039	**0.035**
CMC	Mixed	**55.19**	55.13	0.319	**0.281**
Chron. Kidney Dis. (CKD)	Mixed	97.50	97.50	0.069	**0.058**
Abalone	Mixed	**26.07**	25.69	2.344	**2.106**
Annealing (ANN)	Mixed	91.85	91.85	0.198	**0.190**
Hypothyroid (HYP)	Mixed	99.55	99.55	0.743	**0.621**
Arrhythmia (ARR)	Mixed	70.80	70.80	1.082	**0.961**
Average %/Total secs		83.47	**83.57**	28.583	**24.310**

Although research shows attempts to implement classification algorithms within low-power MCUs have been made, the lack of RAM in many of these devices is likely as much an impediment as their limited processing capability. This effectively limits the size of datasets that are 'mine-able', whether in terms of instances or attributes. Thus, reducing algorithm RAM needs is another area of further research relevant to local IoT data-mining. Nevertheless, the speed gains achieved by SPAARC with minimal loss of classification accuracy could contribute toward greater implementation of data-mining in IoT applications. This is another area we are keen to research.

6 Conclusion

In this paper, we have proposed a novel implementation of sampling methods we have called SPAARC to reduce the computational workload of decision tree induction. The first of these methods involves dynamically selecting attributes with above-median information gains, then using the current tree depth level to switch between all attributes and the selected attribute subspace for node testing. The second method samples possible attribute value split-points by hopping across the distinct value space at equal widths just prior to the tree algorithm's information gain calculations. The combination of these methods improved upon SimpleCART's classification accuracy in more than half of the 18 datasets tested, while reducing the model build time in 15 of those datasets by as much as 69%. These methods only apply to the tree growth phase and leave the pruning phase open for further research that we plan to pursue.

Acknowledgements. This research is supported by an Australian Government Research Training Program (RTP) scholarship.

References

1. Islam, M.Z., Furner, M., Siers, M.J.: WaterDM: a knowledge discovery and decision support tool for efficient dam management (2016)
2. Dangare, C.S., Apte, S.S.: Improved study of heart disease prediction system using data mining classification techniques. Int. J. Comput. Appl. **47**(10), 44–48 (2012)
3. Han, J., Pei, J., Kamber, M.: Data Mining: Concepts and Techniques. Elsevier, New York (2011)
4. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and Regression Trees. CRC Press, Boca Raton (1984)
5. Quinlan, J.R.: C4.5: Programs for Machine Learning. Elsevier, New York (2014)
6. Nath, S.: ACE: exploiting correlation for energy-efficient and continuous context sensing. In: Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services ACM (2012)
7. Srinivasan, V., Moghaddam, S., Mukherji, A., Rachuri, K.K., Xu, C., Tapia, E.M.: MobileMiner: mining your frequent patterns on your phone. In: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing ACM (2014)
8. Hinwood, A., Preston, P., Suaning, G., Lovell, N.: Bank note recognition for the vision impaired. Australas. Phys. Eng. Sci. Med. **29**(2), 229 (2006)
9. Maurer, U., Smailagic, A., Siewiorek, D.P., Deisher, M.: Activity recognition and monitoring using multiple sensors on different body positions. In: International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2006) IEEE (2006)
10. Darrow, B.: Amazon just made a huge change to its cloud pricing. http://fortune.com/2017/09/18/amazon-cloud-pricing-second/ Accessed 30 June 2018
11. Fayyad, U.M., Irani, K.B.: On the handling of continuous-valued attributes in decision tree generation. Mach. Learn. **8**(1), 87–102 (1992)
12. Ranka, S., Singh, V.: CLOUDS: a decision tree classifier for large datasets. In: Proceedings of the 4th Knowledge Discovery and Data Mining Conference (1998)
13. Chandrashekar, G., Sahin, F.: A survey on feature selection methods. Comput. Electr. Eng. **40**(1), 16–28 (2014)

14. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. **3**(Mar), 1157–1182 (2003)
15. Dheeru, D., Karra Taniskidou, E.: UCI machine learning repository. https://archive.ics.uci.edu/ml/datasets.html Accessed 12 Aug 2018
16. Buntine, W., Niblett, T.: A further comparison of splitting rules for decision-tree induction. Mach. Learn. **8**(1), 75–85 (1992)

LCS Based Diversity Maintenance in Adaptive Genetic Algorithms

Ryoma Ohira(✉), Md. Saiful Islam, Jun Jo, and Bela Stantic

School of Information and Communication Technology, Griffith University,
Southport, QLD 4215, Australia
{r.ohira,saiful.islam,j.jo,b.stantic}@griffith.edu.au

Abstract. A genetic algorithm (GA) experiences premature convergence when the diversity is lost in the population. Adaptive GAs aim to maintain diversity in the population by trading off a balance between exploring the problem space and exploiting known solutions. Existing metrics for population diversity measures only examine the similarity between individuals on a genetic level. However, similarities in the order of genes in individuals in ordered problems, such as the travelling salesman problem (TSP) can play an important role in effective diversity measures. By examining the similarities of individuals by the order of their genes, this paper proposes longest common subsequence (LCS) based metrics for measuring population diversity and its application in adaptive GAs for solving TSP. Extensive experimental results demonstrate the superiority of our proposal to existing approaches.

Keywords: Adaptive GA · LCS · Diversity maintenance · TSP

1 Introduction

Genetic Algorithms (GAs) find and develop solutions through exploring the global search space to look for good solutions and then exploiting found solutions to discover optimal solutions. By maintaining a balance between exploration and exploitation, a GA attempts to search and converge on the global optima. However, a simple GA has a tendency to converge to a local optima without sufficiently exploring the search space. This premature convergence is often attributed to a loss of diversity through selective pressure, mutation rates and crossover between similar individuals. Identifying optimal settings for these parameters for maintaining population diversity is a long-standing challenge for GAs [1]. While studies [2,3] have shown that a GA can have a significant performance increase when parameter controls are applied in an adaptive and on-line manner, the success of adaptive controls are reliant on performance and health measures. With GAs, the no free lunch (NFL) theorem is presented in how a given set of parameter control mechanisms may not be optimal, or even applicable, to another problem [1]. One sub-set of optimisation problems are ordered problems where the order of nodes in a solution is critical to the quality of the

R. Islam et al. (Eds.): AusDM 2018, CCIS 996, pp. 56–68, 2019.
https://doi.org/10.1007/978-981-13-6661-1_5

solution. An example of this is the travelling salesman problem (TSP) where a salesman must travel to n number of cities and the objective is to find the shortest route possible with the salesman visiting each city once.

1.1 Related Works

Early diversity metrics simply calculate the Hamming distance between individuals [4,5]. Ursem et al. [6] extend [5] and propose a Hamming distance based diversity metric considering the size of the population and problem. Mc Ginley et al. [7] propose a novel healthy population diversity (HPD) metric that examines the diversity of a population from a fitness perspective for their ACROMUSE GA. ACROMUSE utilises both a standard population diversity (SPD) measure similar to earlier works and the proposed HPD to demonstrate how the relationship between a healthy, or highly fit, population and a diverse population can contribute to maintaining a diverse population. By introducing the HPD, ACROMUSE outperforms other adaptive GAs that use diversity based adaptive parameter controls in benchmark tests. However, ACROMUSE and its competitors were all based on diversity metrics that only measured the gene-wise similarity between individuals and tested with general optimisation problems. Experiments demonstrate lower performance gains when ACROMUSE and its gene-wise approach to measuring population diversity is applied to the TSP.

Many works make novel use of diversity metrics based on the works [8–10], however, they do not offer new metrics for measuring population diversity. While general diversity measures and adaptive GAs are an improvement over traditional GAs [4–11] for ordered problems such as the TSP, they do not take into consideration the relationship between nodes in the problem space. As such, existing population diversity metrics for general optimisation problems are not efficient for the ordered problems, including the TSP.

1.2 Our Contributions

This paper proposes new metrics for measuring the SPD and HPD of a population for problems with ordered lists by considering the longest common subsequence (LCS) between the fittest individual and the population (Sect. 2). Furthermore, the application of these new SPD and HPD metrics are applied to an adaptive GA framework that applies crossover and mutation rates in on on-line manner and dynamically manages selective pressure on exploration and exploitation sub-populations (Sect. 3). An experimental evaluation is also presented to demonstrate the effectiveness of the proposed approach (Sect. 4).

2 Proposed LCS Based Diversity Metrics

In this section, we propose a metric for measuring population diversity with a sequence-wise approach. To motivate our approach consider different tours for Berlin52 (an instance of TSP problem) as illustrated in Fig. 1, where Fig. 1a is the optimal tour. A gene-wise approach to measuring population diversity will

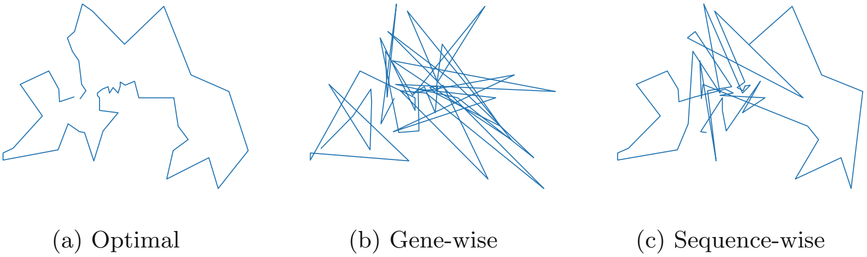

(a) Optimal (b) Gene-wise (c) Sequence-wise

Fig. 1. Comparison of sequence-wise similarity and gene-wise similarity to the optimal tour for Berlin52

find that Fig. 1b has a 50% genetic similarity to the optimal solution and the tour represented in Fig. 1c has no similarity to the optimal solution. However, a sequence based diversity metric will find a 50% similarity between Fig. 1a and c. With the TSP, a tour may start from a different city but if the sequence in which the cities are visited are the same, the total distance travelled will also be the same. The solution in Fig. 1b has a travel distance of 24,986 while the solution in Fig. 1c is 13,786. This example demonstrates the importance of considering the sequence of genes in an ordered problem and the need for it to be taken into consideration when measuring the diversity of a GA's population.

To develop sequence-based diversity metrics, we can adopt either a contiguous sequence based approach or a non-contiguous sequence based approach. A contiguous sequence is where each node in the sub-sequence must be sharing a border with the next node. Individuals with a high level of gene-wise similarity will have a high level of non-contiguous, sequence-wise similarity. However, they may not necessarily share a high level of contiguous sequence-wise similarity. Therefore, this paper adopts non-contiguous sequence based approach for determining the similarity between two individuals and exploits *longest common subsequence* (LCS) for developing diversity measures. The LCS identifies the longest common non-contiguous subsequence between two individuals [12]. The dynamic programming formulation of LCS is given in Eq. 1, where X and Y are the two sequences that are being compared.

$$LCS(X_l, Y_k) = \begin{cases} 0 & \text{if}\, l = 0\, \text{or}\, k = 0 \\ LCS(X_{l-1}, Y_{k-1}) + 1 & \text{if}\, X_l = Y_k \\ max(LCS(X_l, Y_{k-1}, LCS(X_{l-1}, Y_k)) & \text{if}\, X_l \neq Y_k \end{cases} \tag{1}$$

Standard Population Diversity (SPD). SPD describes the variation of individuals of a population regardless of its fitness [7]. It is generally used to control adaptive mutation and crossover parameters in order to maintain population diversity. In order to calculate the SPD, the average LCS length for the generation is calculated as G^{μ}. This can be expressed in Eq. 2 as the average LCS length between the fittest individual, F, and the remaining individuals in population P. While population

$$G^{\mu} = \frac{1}{P}\sum_{i=1}^{P} LCS(F, i) \tag{2}$$

diversity could be defined as a simple average, different problems and populations may result in large variances in SPD. To prevent this, we normalise the standard deviation against the mean. Equation 3 calculates the standard deviation of the LCS lengths between individuals of the population.

$$\sigma(G^{\mu}) = \sqrt{\frac{1}{P}\sum_{i=1}^{P}(LCS(F,i) - G^{\mu})^2} \tag{3}$$

The standard deviation $\sigma(G^{\mu})$ is then used to calculate the coefficient of variance in Eq. 4. Finally, the coefficient of variance is then used as the SPD to measure the variation of individuals in a population.

$$SPD = C_v(G^{\mu}) = \frac{\sigma(G^{\mu})}{G^{\mu}} \tag{4}$$

Healthy Population Diversity (HPD). HPD differs itself from SPD by considering the fitness of the individuals. A healthy population refers to a population with a high degree of fitness and diversity. The HPD is used as a control mechanism for applying selective pressure [7]. Equation 5 calculates the individual i's contribution to the overall fitness of population P. This is expressed as its fitness in proportion to the total fitness of the population. By applying the weights to LCS length of each individual, the weighted average is calculated as G^{μ}_{w}. This enables the HDP to take into consideration the sequence-wise contribution of each individual's fitness to the overall diversity and health of the population.

$$w_i = \frac{f_i}{\sum_{k=1}^{P} f_k} \tag{5}$$

Similar to the SPD measure, the HPD measure also needs to be normalised against the mean in order to prevent variances that may occur. Therefore, the standard deviation of the weighted LCS is calculated using the weighted average genotype, G^{μ}_{w}, which is given in Eq. 7.

$$G^{\mu}_{w} = \frac{1}{P}\sum_{i=1}^{P} w_i LCS(F,i) \tag{6}$$

$$\sigma(G^{\mu}_{w}) = \sqrt{\sum_{i=1}^{P} w_i(LCS(F,i) - G^{\mu}_{w})^2} \tag{7}$$

Finally, the HPD is calculated as being the coefficient of variance as in Eq. 8 and is considered as a fitness-based measure of population diversity.

$$HPD = C_v(G^{\mu}_{w}) = \left(\frac{\sigma(G^{\mu}_{w})}{G^{\mu}}\right) \tag{8}$$

3 LCS-Based Adaptive Genetic Algorithm

This section describes the details of *LCS based adaptive genetic algorithm* (LCSB-AGA) and its implementation of the SPD and the HPD. The LCSB-AGA uses a T_{size} tournament selection method [13] with a modified crossover operator

(MOX) [14] and partial shuffled mutation (PSM) [15] as the mutation operator. MOX and PSM were selected for their performance and ability to retain integrity constraints for ordered lists. The framework for LCSB-AGA is presented in Fig. 2 where components with adaptive mechanisms have been highlighted.

LCSB-AGA uses an adaptive crossover probability (P_c) to determine whether an offspring will be generated through crossover. P_c is determined by the SPD of the previous generation and is used to create an exploration and exploitation sub-populations. When an offspring is determined to be generated through crossover, two individuals from the previous generation are selected through tournament selection. The HPD is used to determine the T_{size} for the tournament in order to adaptively apply selective pressure according to the health and diversity of the population. Crossover and mutation are applied with the P_m rate being 0.01. This encourages the exploitation sub-population to refine existing solutions.

When an offspring is designated to be generated without crossover, it becomes part of the exploration sub-population. A single individual is selected from a tournament where the T_{size} is determined by the HPD. Adaptive mutation is applied where P_m is determined by the SPD. This enables the GA to explore new solutions where a highly converged population will result in a high P_m.

By creating the two sub-populations, LCSB-AGA is able to process each with different strategies, designed for either exploration or exploitation. This enables LCSB-AGA to dynamically balance exploration and exploitation, which is similar to the idea presented in ACROMUSE [7]. Furthermore, as the size of these sub-populations are determined by P_c, which is in turn determined by SPD, the sizes of these sub-populations adapt to a population's diversity. Thus the LCSB-AGA is able to adjust the bias between exploration and exploitation according to the degree to which it has converged. This enables LCSB-AGA to maintain a healthy population diversity.

Along with the adaptive mechanisms outlined in Fig. 2, LCSB-AGA implements elitism by selecting an elite individual from the population. This is done through a tournament selection where $T_{size} = P$. This ensures that the best solution for each generation is not lost and is carried to the new generation.

3.1 LCSB-AGA Adaptive Crossover

Probability of crossover (P_c) is determined by the degree of diversity of the population and determines whether crossover occurs for the offspring. This is expressed as in Eq. 9.

$$P_c = \left[\left(\frac{SPD}{SPD_{max}} \times (K_2 - K_1)\right) + K_1\right] \tag{9}$$

SPD_{max} is set to 4 as the coefficient of variance does not exceed 4 in practice. Two constants are used to manage limits with the P_c to ensure that a minimum proportion exists for the exploitation and exploration sub-populations. K_1 defines the minimum proportion of individuals that are protected from exploitation. With K_1 given a value of 0.4, in the situation where there is total convergence on a solution, 40% of the population is dedicated to exploration. K_2

defines the proportion of a population that is to be reserved for exploitation in a population with a maximum diversity attainable. K_2 is set to 0.2 to allow for 20% of the population to continue exploring with the majority focusing on exploitation.

3.2 LCSB-AGA Adaptive Selection

Adaptive selection is implemented through applying selective pressure according to the diversity of the population. This is done through adjusting the tournament size, T_{size}, for the tournament selection method as shown in Eq. 10. T_{max} is given a value of $\frac{P}{6}$ as results demonstrate that a tournament size larger than $\frac{1}{6}$ of the population size results in an instant loss of population diversity. This value ensures that a balance between selection pressure and diversity is maintained. HPD_{max} is the maximum attainable HPD value. Experiment results have shown that, in practice, the HPD does not exceed a value of 3.

$$T_{size} = T_{max} \frac{HPD}{HPD_{max}} \quad (10)$$

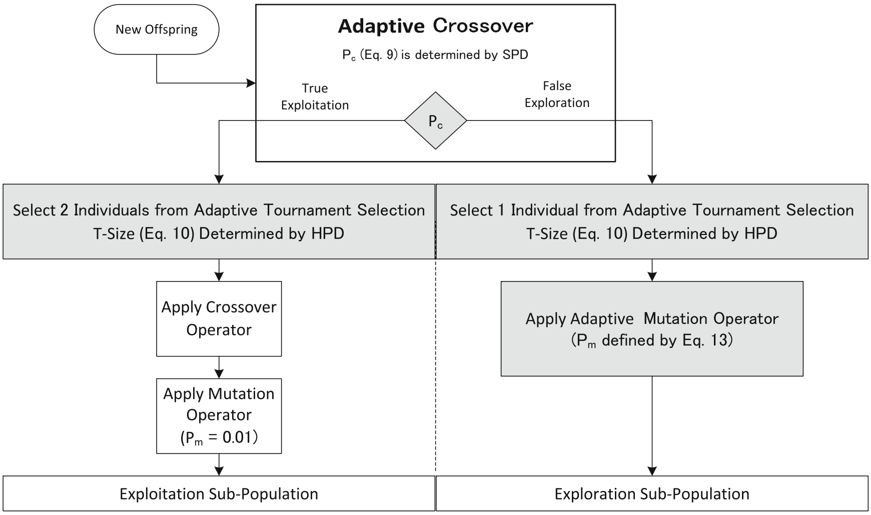

New Offspring
Adaptive Crossover
P_c (Eq. 9) is determined by SPD
True
Exploitation
False
Exploration
P_c
Select 2 Individuals from Adaptive Tournament Selection
T-Size (Eq. 10) Determined by HPD
Select 1 Individual from Adaptive Tournament Selection
T-Size (Eq. 10) Determined by HPD
Apply Crossover Operator
Apply Mutation Operator (P_m = 0.01)
Apply Adaptive Mutation Operator (P_m defined by Eq. 13)
Exploitation Sub-Population
Exploration Sub-Population

Fig. 2. Schematic diagram of our proposed LCSB-AGA framework

3.3 LCSB-AGA Adaptive Mutation

The probability for mutation (P_m) is the average between the diversity ratio of the population and the fitness ratio of the individual. This enables LCSB-AGA to adapt the mutation control parameters according to the population diversity and the relative fitness of individuals. Equation 11 shows the contribution of the SPD to the P_m where K is the upper bound of P_m and is set to 0.5 to ensure only a maximum mutation probability of 50%. SPD_{max} is the maximum attainable SPD value as described in Eq. 9. Equation 12 describes the contribution of an individual's fitness as a ratio of the overall fitness. As an individual's fitness is yet to be calculated at this point, the fittest parent's f value is used. f_{max} and f_{min} are respectively the best and worst fitness values from the previous population. K represents the upper limit and is given a value of 0.5. With both the $P_m^{Fitness}$ and $P_m^{Diversity}$ values, the average can be calculated as the P_m as expressed in Eq. 13. The K value for both Eqs. 11 and 12 ensures that a population that has the maximum diversity can not have a P_m value that exceeds the K limit. A P_m value of 0.5 will result in a very significant change in an individual's genotype.

$$P_m^{Diversity} = K\frac{SPD_{max} - SPD}{SPD_{max}} \quad (11)$$

$$P_m^{Fitness} = K\left(\frac{f_{max} - f}{f_{max} - f_{min}}\right) \quad (12)$$

4 Experiments

This section demonstrates the superiority of our LCSB-AGA in comparison to the best known static GA and ACROMUSE.

$$P_m = \frac{P_m^{Fitness} + P_m^{Diversity}}{2} \quad (13)$$

4.1 Setup

The experiments are conducted on an Intel Core i7-3770 CPU with 32 GB main memory. The algorithms are implemented in Python 2.7 and executed using the PyPy interpreter with Numpy. All three GAs use tournament selection [13], MOX [14] and PSM [15] as operators. The static GA has a static tournament size of 5 with ACROMUSE and LCSB-AGA implementing adaptive tournament sizes. The static GA has a p_c of 0.95 and p_m of 0.015. The p_c and p_m values for ACROMUSE and LCSB-AGA are dependent on their adaptive mechanisms. All GAs were run with a population size of 100. The convergence criteria is outlined in Eq. 14 where n is the problem size. When there are no improvements after c generations, the algorithm will complete. Instances from the Travelling Salesman Problem Library (TSPLIB) were selected according to the diversity in n size. Each test instance is run a total of 50 iterations for each GA. A two sample Z-Test is used to determine the significance of the difference in performance. The z value is calculated as shown in Eq. 15. The degree of freedom is 98 as $d_f = n_1 + n_2 - 2$. Both the d_f and t values are used in the z-distribution to determine the significance (p) in the difference between mean performance. $p \leq 0.05$ demonstrates a significant difference in performance where a $p \leq 0.01$ signifies a very significant difference. The Z-Test columns indicate the significance of the LCSB-AGA's performance improvements over the GA and ACROMUSE. A * indicates no significant improvement while a + indicates a significant improvement.

$$c = n + \sum_{k=1}^{n} k \quad (14)$$

$$z = \frac{\overline{x_1} - \overline{x_2}}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \quad (15)$$

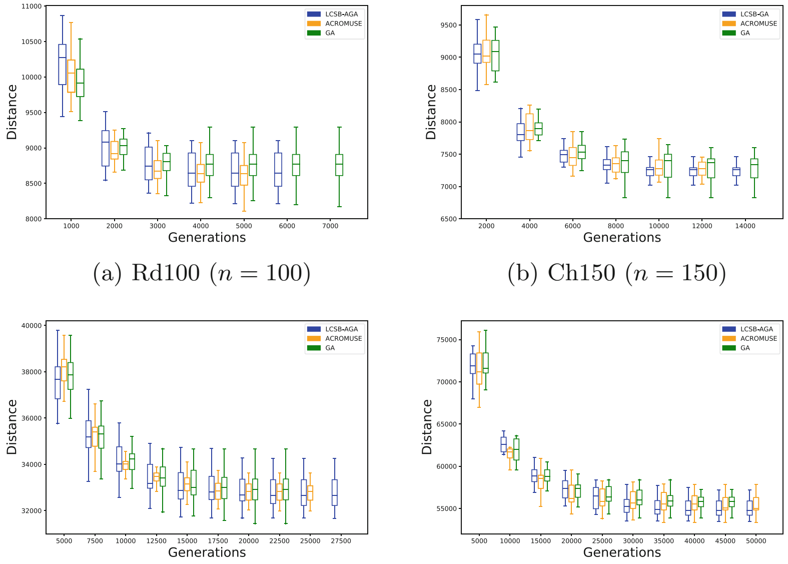
4.2 Results

The experimental results are outlined in Table 1. These results demonstrate that there are no significant gains for either ACROMUSE or LCSB-AGA in problems where $n < 100$ with all GAs achieving average distances close to the optimal distance for each test. However, with problems where $100 \leq n < 200$, the LCSB-AGA manages to significantly outperform both the GA and ACROMUSE on a majority of test instances. While LCSB-AGA has a minor improvement in Eil101, Lin105 and Pr144 test instances, problems where $n \geq 200$, the results demonstrate that LCSB-AGA is able to achieve more robust performance gains as the problem size, n, increases.

Table 1. Results of the best GA, the ACROMUSE and the proposed LCSB-AGA

Test	Optima	GA			ACROMUSE			LCSB-AGA			Z-Test	
		Best.	Avg	S. Dev	Best.	Avg	S. Dev	Best.	Avg	S. Dev	GA	ACR.
wi29	27603	27603	27883	306.3	27603	27915	388.7	27603	27933	377.4	*	*
dj38	6656	6659	6716	116.7	6659	6765	163.6	6659	6691	118.3	*	+
eil51	426	438	451	7.9	433	454	9.2	434	453	8.8	*	*
berlin52	7542	7544	8089	261.6	7544	8036	223.6	7544	7834	254	+	+
st70	675	694	722	14.9	691	722	21.3	688	717	18.6	*	*
eil76	538	566	593	12.6	562	592	13.8	564	587	10.8	+	+
pr76	108159	109258	115261	3069.9	110195	114771	2953.8	109456	114018	3242.2	+	*
rat99	1211	1298	1349	25.4	1270	1351	33.5	1274	1336	33.6	+	+
rd100	7910	8282	8664	201.7	8191	8673	225.2	8127	8569	263.9	+	+
eil101	629	667	700	12.9	667	696	12.4	678	694	11.4	+	*
lin105	14379	14682	15499	460.3	14713	15635	416.5	14565	15162	410.1	+	+
pr107	44303	44762	47431	1302.8	44939	47211	1373.1	44685	46825	1047.3	+	*
pr124	59030	59736	62363	1789.3	60322	62592	1561.8	59409	61652	1796.3	+	+
ch130	6110	6394	6696	133.3	6376	6685	178.1	6345	6616	170.8	+	+
pr136	96772	101478	105843	2514.6	100574	106173	2394.4	101289	104868	2272.6	+	+
pr144	58537	59034	61627	1831.8	59115	62613	2413.2	58831	61459	2144.1	*	+
ch150	6528	7035	7294	156.7	6899	7275	184.6	6714	7203	172.9	+	+
pr152	73682	75618	78098	1489.5	75514	78308	1776.3	75278	77130	1377.4	+	+
rat195	2323	2510	2630	44.1	2541	2641	52.5	2496	2512	55.9	+	+
d198	16169	16169	16660	197	16169	16708	193.4	16169	16516	191.6	+	+
kroB200	29437	31720	32747	538.5	31157	32775	680.9	30689	32124	754.7	+	+
pr226	80369	81711	85995	2574.6	81788	85992	3372.8	81824	84144	3548.8	+	+
pr264	49135	53562	55756	1289.8	53177	55843	1400.1	52409	54923	1316.4	+	+
pr299	48191	51669	54201	1172.6	51120	54210	1120.5	5215	53366	1170.7	+	+
lin318	42029	45589	46904	727.8	44762	47017	965.1	44571	46204	861.4	+	+
pr439	107217	116090	120522	2618.7	114446	121546	2584.1	117840	118917	2876.7	+	+

Fitness Analysis. By examining four instances from Table 1, Fig. 3 illustrates some of the characteristics of LCSB-AGA, ACROMUSE and GA as they solve given problems. In Fig. 3a, all three algorithms can be seen as having a similar rate of improvement in fitness. However, ACROMUSE can be seen as converging on a good solution the fastest with LCSB-AGA and the traditional GA following afterwards. If looking at this data alone, the traditional GA could either be the least efficient at exploiting known solutions or generating the most diverse populations. This can be clarified by examining the SPD and HPD later. However, with the small problem size, where $n = 100$, the final solutions found by the GAs are similar with no significant differences in performance. Figure 3b shows the performances of the algorithms in a slightly larger problem set where $n = 150$. Here, the LCSB-AGA can be seen as having a faster rate of convergence at the beginning and continues to explore longer than ACROMUSE. This suggests that the sequence-wise approach to maintaining diversity is capable of exploring and then exploiting solutions with greater efficacy than a gene-wise approach. The traditional GA can be seen as having a longer time than ACROMUSE to converge but also having lower efficacy than both ACROMUSE and LCSB-AGA. As the problem sizes get larger, the LCSB-AGA can be seen as having a greater and more significant advantage over ACROMUSE and the GA. In Fig. 3c LCSB-AGA maintains its population diversity longer than the other algorithms but also has a significant improvement in its average fitness. This is even more pronounced in Fig. 3d.



(a) Rd100 ($n = 100$)

(b) Ch150 ($n = 150$)

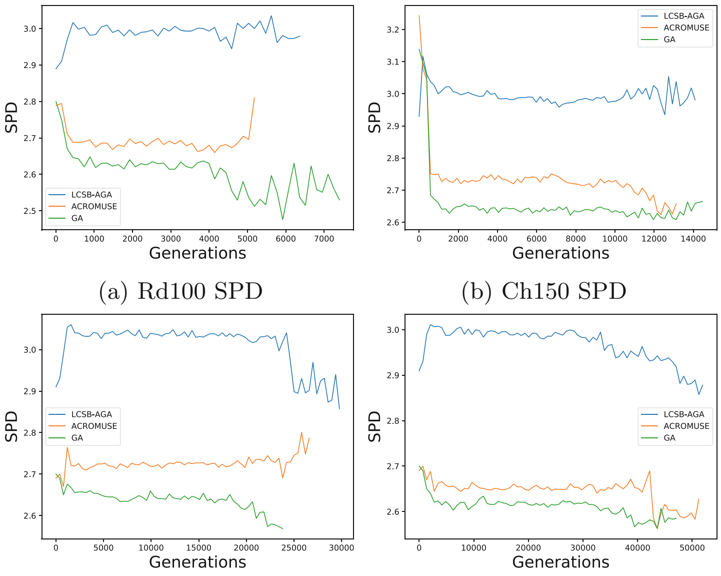
(c) KroB200 ($n = 200$)

(d) Pr264 ($n = 264$)

Fig. 3. Comparison of LCS, ACROMUSE and GA for best of fitness

Standard Population Diversity Analysis. SPD measures the population diversity without taking the population's overall fitness into consideration. This metric provides insight into the overall diversity of a population. Figure 4 illustrates the SPD trend for each GA when applied to the Rd100, Ch150, KroB200 and Pr264 TSPLIB problem instances. These instances were selected to demonstrate the characteristics of our sequence-wise population diversity metrics as the problem size increases. While ACROMUSE and LCSB-AGA both share similar strategies for maintaining diversity, it can be seen that LCSB-AGA manages to maintain a higher level of sequence-wise diversity throughout the generations. The results in Table 1 show the sequence-wise approach of the LCSB-AGA has significantly improved performance over the gene-wise approach of ACROMUSE, we can see here that LCSB-AGA is also able to achieve and maintain a higher level of population diversity. In fact, Fig. 4 demonstrates that there is little difference between ACROMUSE and the GA with sequence-wise population diversity.

One common characteristic between all methods is how the SPD appears to fluctuate prior to converging. The traditional GA experiences larger fluctuations while the LCSB-AGA can be seen as attempting to alleviate the lack of diversity. ACROMUSE appears to also respond to the converging population by activating its own diversity mechanisms. In comparison with the GA, both the LCSB-AGA and ACROMUSE can be seen as fluctuating less severely and less frequently. This is likely due to the algorithms' attempts at maintaining population diversity. ACROMUSE experiences a sudden loss in SPD in Fig. 4d and both GA and LCSB-AGA in Figs. 4a and c. All five cases demonstrate how a sudden loss in SPD results in an early convergence. When the adaptive GA is able to maintain diversity as the population converges, it is able to continue improving its solution. This can be seen in Fig. 4d where LCSB-AGA begins to lose diversity at generation 30,000 but continues until converging around generation 50,000.



(a) Rd100 SPD

(b) Ch150 SPD

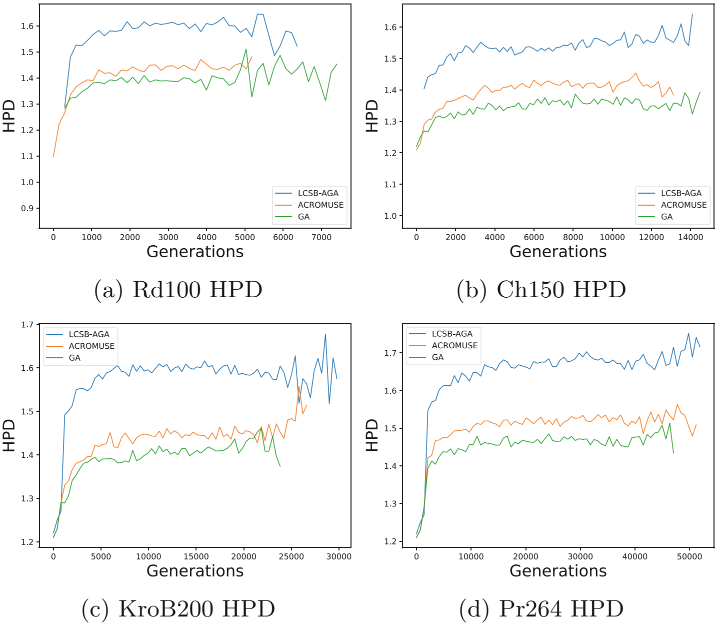
(c) KroB200 SPD

(d) Pr264 SPD

Fig. 4. SPD: Comparison between LCSB-AGA, ACROMUSE and GA

Another common characteristic from Fig. 4 is how a sudden and significant loss in diversity is soon followed by a premature convergence. This is a logical conclusion given the definition of convergence. In Fig. 4c, a number LCSB-AGA test runs where the test had run for a longer number of generations appear to have experienced lower diversity. LCSB-AGA can be seen as trying to recover its lost diversity by the fluctuations in its SPD. This can be attributed to LCSB-AGA attempting to re-balance its exploration and exploitation sub-populations. This loss in diversity lead to P_m values reaching 0.32 which would have resulted in very disruptive changes to many individuals.

Healthy Population Diversity Analysis. HPD is a fitness-weighted measure of the population diversity. A highly fit and diverse population is considered to be a healthy population. Figure 5 demonstrates the HPD trend for each GA when applied to the Rd100, Ch150, KroB200 and Pr264 TSPLIB problem instances. The graphs presents the average HPD value over 50 runs at regular intervals. A higher value represents a healthier population. Across the four instances, the HPD can be seen as starting at a low value and increasing in a manner similar to the inverse of the fitness data from Fig. 3 as should be expected. A number of interesting characteristics of the HPD are presented in the figures. As the HPD takes fitness and diversity into consideration, the difference between LCSB-AGA and the other two algorithms is more significant. In smaller problems, such as Rd100 in Fig. 5a, there is little difference between the GA and ACROMUSE



(a) Rd100 HPD

(b) Ch150 HPD

(c) KroB200 HPD

(d) Pr264 HPD

Fig. 5. HPD: comparison between LCS, ACROMUSE and GA

with both catching up to the HPD performance of LCSB-AGA towards the end. This is expected as all three methods performed strongly in the smaller problems with no significant difference in fitness between the three. However, the higher HPD of LCSB-AGA indicates that the greater population diversity contributed greatly. It can be seen that as the problem size increases, so do the differences between LCSB-AGA, ACROMUSE and GA.

One interesting observation is the KroB200 instance when comparing Fig. 4c and Fig. 5c. As stated previously, some runs for the KroB200 instance experienced lower diversity which resulted in longer run times and LCSB-AGA can be seen as trying to recover its diversity by introducing disruptive levels of mutation, as demonstrated in the fluctuations in SPD. However, when viewed in context with the HPD data, it can be seen that LCSB-AGA was successful in maintaining a reasonable level of healthy population diversity as the general trend maintained an upward projection.

5 Conclusion and Future Work

The TSP is a well known ordered optimisation problem with much theoretical and practical implications for which many variations of GAs have been proposed. While existing adaptive GAs have performed very well for general optimisation problems, they have not demonstrated the same success for ordered problems. In this paper, we present LCSB-AGA as an adaptive GA for ordered optimisation problems such as the TSP. A novel sequence-wise approach is also proposed for measuring the population diversity. The performance of the LCSB-AGA in terms of solution quality is demonstrated in the extensive experiments presented in this paper. In particular, the Z-Tests highlight the significance in the improvements as the problem size gets larger. Furthermore, by inspecting the SPD and HPD data, the sequence-wise approach is found to be having a greater degree of diversity and being better equipped to maintain it. The future work of this paper includes further investigation of the proposed sequence-based diversity metrics for other ordered optimisation problems such as the vehicle routing problem.

References

1. Eiben, Á.E., Hinterding, R., Michalewicz, Z.: Parameter control in evolutionary algorithms. IEEE Trans. Evol. Comput. **3**(2), 124–141 (1999)
2. Chaiyaratana, N., Piroonratana, T., Sangkawelert, N.: Effects of diversity control in single-objective and multi-objective genetic algorithms. J. Heuristics **13**(1), 1–34 (2007)
3. Pan, Q.K., Suganthan, P.N., Wang, L., Gao, L., Mallipeddi, R.: A differential evolution algorithm with self-adapting strategy and control parameters. Comput. Oper. Res. **38**(1), 394–408 (2011)
4. Zhu, K.Q.: A diversity-controlling adaptive genetic algorithm for the vehicle routing problem with time windows. In: Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence, pp. 176–183. IEEE (2003)

5. Shimodaira, H.: A diversity-control-oriented genetic algorithm (DCGA): performance in function optimization. In: Proceedings of the 2001 Congress on Evolutionary Computation, vol. 1, pp. 44–51. IEEE (2001)
6. Ursem, R.K.: Diversity-guided evolutionary algorithms. In: Guervós, J.J.M., Adamidis, P., Beyer, H.-G., Schwefel, H.-P., Fernández-Villacañas, J.-L. (eds.) PPSN 2002. LNCS, vol. 2439, pp. 462–471. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45712-7_45
7. Mc Ginley, B., Maher, J., O'Riordan, C., Morgan, F.: Maintaining healthy population diversity using adaptive crossover, mutation, and selection. IEEE Trans. Evol. Comput. **15**(5), 692–714 (2011)
8. Vidal, T., Crainic, T.G., Gendreau, M., Prins, C.: A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. Comput. Oper. Res. **40**, 475–489 (2013)
9. Segura, C., Hernandez, A., Luna, F., Alba, E.: Improving diversity in evolutionary algorithms: new best solutions for frequency assignment. IEEE Trans. Evol. Comput. **21**(4), 539–553 (2017)
10. Cruz-Salinas, A.F., Perdomo, J.G.: Self-adaptation of genetic operators through genetic programming techniques. In: GECCO, pp. 913–920. ACM (2017)
11. Adra, S.F., Fleming, P.J.: Diversity management in evolutionary many-objective optimization. IEEE Trans. Evol. Comput. **15**, 183–195 (2011)
12. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. MIT Press, Cambridge (2009)
13. Goldberg, D.E., Deb, K.: A comparative analysis of selection schemes used in genetic algorithms. Found. Genet. Algorithms **1**, 69–93 (1991)
14. Abdoun, O., Abouchabaka, J.: A comparative study of adaptive crossover operators for genetic algorithms to resolve the traveling salesman problem. Int. J. Comput. Appl. **31**(11), 49–57 (2011)
15. Abdoun, O., Abouchabaka, J., Tajani, C.: Analyzing the performance of mutation operators to solve the travelling salesman problem. Int. J. Emerg. Sci. **2**, 61–77 (2012)

Categorical Features Transformation with Compact One-Hot Encoder for Fraud Detection in Distributed Environment

Ikram Ul Haq[1(✉)], Iqbal Gondal[1],
Peter Vamplew[1], and Simon Brown[2]

[1] ICSL, School of Science, Engineering and Information Technology,
PO Box 663, Ballarat, VIC 3353, Australia
ikramulhaq@students.federation.edu.au,
{iqbal.gondal,p.vamplew}@federation.edu.au
[2] Westpac Bank, Melbourne, Australia
simonbrown@westpac.com.au

Abstract. Fraud detection for online banking is an important research area, but one of the challenges is the heterogeneous nature of transactions data i.e. a combination of numeric as well as mixed attributes. Usually, numeric format data gives better performance for classification, regression and clustering algorithms. However, many machine learning problems have categorical, or nominal features, rather than numeric features only. In addition, some machine learning platforms such as Apache Spark accept numeric data only. One-hot Encoding (OHE) is a widely used approach for transforming categorical features to numerical features in traditional data mining tasks. The one-hot approach has some challenges as well: the sparseness of the transformed data and that the distinct values of an attribute are not always known in advance. Other than the model accuracy, compactness of machine learning models is equally important due to growing memory and storage needs. This paper presents an innovative technique to transform categorical features to numeric features by compacting sparse data even if all the distinct values are not known. The transformed data can be used for the development of fraud detection systems. The accuracy of the results has been validated on synthetic and real bank fraud data and a publicly available anomaly detection (KDD-99) dataset on a multi-node data cluster.

Keywords: One-hot Encoder · Compactness · Categorical data · Distributed computing · Hadoop · HDFS · Spark · Machine learning · Sparse data

1 Introduction

Outlier detection techniques have been in use for many applications including Intrusion and Fraud Detection [1–5]. Most of the outlier detection methods use homogeneous datasets having the single type of attributes like numerical or categorical attributes, but real-world datasets often have a combination of these attribute types [6]. For example, Maruatona [4] explains that a typical bank transaction datasets have attributes which are a combination of numeric and categorical attributes.

R. Islam et al. (Eds.): AusDM 2018, CCIS 996, pp. 69–80, 2019.
https://doi.org/10.1007/978-981-13-6661-1_6

Numeric features give better performance in classification and regression algorithms. Similarly, clustering algorithms work effectively on the data where all attributes are either numeric or categorical data, as most of the algorithms perform poorly on mixed data types [7]. Huang [8] describes in his finding that clustering methods like k-means are efficient for processing large datasets, but these methods are often limited to numeric data. In addition, machine learning software may only support certain types of data. For example, Apache Spark [9–11] is a highly scalable platform to run machine learning algorithms in a distributed environment, but it accepts only numeric data for classification, regression and clustering algorithms. Therefore, there may be a need to convert categorical variables to a numerical encoding.

Categorical variables are commonly encoded using One-hot Encoding (OHE). Chen [12] indicates that in many traditional data mining tasks, OHE is widely used for converting categorical features to numerical features. OHE transforms a single variable with n observations and d distinct values, to d binary variables with n observations each. Each observation indicates the presence 1 or absence 0 of dth binary variable. However, data becomes sparse after this transformation.

Sparse datasets are common in the big data, where the sparsity comes from factors i.e. feature transformation (OHE), large feature space and missing data [13]. For a given attribute, OHE will increase the number of attributes from one to n distinct values in that attribute, which will not only make the datasets high dimensional but also increase datasets size. Chen [12] believes that other than the accuracy, due to growing memory and storage consumption, compactness of machine learning models will become equally important in the future.

We have presented a technique to transform categorical attributes to numeric attributes and compact the sparsity. The transformed data can be used for the experimental validation and development of fraud detection technique, especially for scalable and distributed data. This technique is tested on a fraud detection bank data and on an anomaly detection KDD-99 dataset, which is widely used as one of the few publicly available datasets for anomaly detection [14]. Multi-node Hadoop cluster is used for experiments, and the performance comparison of the technique has been presented with different classification techniques.

1.1 Contribution

Considering model accuracy and importance of growing memory and storage needs, we have developed a technique to transform categorical attributes to numeric attributes and compact the sparsity as well. An innovative technique is developed and presented in this paper to transform categorical features to numeric features by compacting sparse data even when all the distinct values are not known in advance. Two further models are also developed in One-hot Encoding Extended Compact technique and classification accuracy is evaluated with both models.

Our main contributions in this research are summarized as follows:

(a) Developing One-hot Encoded Extended (OHE-E) technique.

(b) Extending One-hot Encoded Extended with Compactness (OHE-EC).

(c) Develop two further models: First Come First Serve (FCFS) and High Distribution First (HDF) in One-hot Encoded Extended Compact (OHE-EC).

(d) Evaluating classification accuracy, the effect on data size and efficiency in terms of training model and prediction with well-known classification techniques.
(e) Empirical evaluation with a synthetic dataset generated from real bank transaction data and the well-known KDD 95 dataset.

2 Related Work

Several efforts have been made in the past to transform categorical attribute to numeric attributes. First attempt and one of the popular way to convert a categorical feature to a numerical is OHE, but this transformation results in high-dimensional sparse data. Jian et al. [15] have transformed categorical data with Coupled Data Embedding (CDE) technique by extending coupling learning methodology by obtaining hierarchical value-to-value cluster couplings. CDE is slower than other embedding methods, thus is not ideal for large data-sets. It is only applied to unsupervised clustering domain. Another categorical data-representation technique was proposed by Qian et al. [16] with an objective of solving the problem of the categorical data not having a clear space structure. They have not addressed the problem of clustering for a mixed dataset. A comparative evaluation of similarity measures for categorical data is done by Boriah et al. [17]. But the evaluation is performed in a specific context of outlier detection, and relative performance of similarity measures is not studied for classification and clustering. Boriah et al. [17] highlight that several books on cluster analysis [18–20] that discuss the problem of determining the similarity between categorical attributes, recommend binary transformation of data for similarity measures.

To overcome these limitations and for better accuracy, we have presented a technique to transform categorical attributes into numeric attributes and compact the sparsity. This data can be used for the experimental validation and development of fraud detection technique, to check scalability in a distributed environment.

3 Methodology

We have further extended Highly correlated rule-based uniformly distributed synthetic data (HCRUD) [21] to generate numeric synthetic data from mixed reference data. Multi-node Hadoop cluster is used for experiments in a distributed environment with a name-node, resource-manager and multiple workers and data-nodes. The complete process of loading data, filtering categorical features, distribution, transformation, and compactness is explained in the algorithm below.

3.1 Algorithm

```
# Load source data and do Feature selection with Singular Value Decomposition SVD using Eq.(1).
# Filter categorical features only. Distribute data rows on worker-nodes in distributed environment in multi-node Hadoop cluster using Eq.(4). Block size and replication factor is configurable. We have used 64-MB block size and three replication factor. Distributing data on worker-nodes gives efficiency with data locality. Process rows on worker-nodes in parallel and Process each Row.
  a. Process each Feature
  b. IF (Feature is Selected and Categorical)
    i. For each Feature transform with OHE-E adding extra feature using Eq.(5).
# Missing value imputation (MVI) is applied with majority value of a given attribute for selected attributes. The decision of taking extra attribute is configured in various contextual and model-based profiles. It is evaluated with different measures explained in 3.3.
    ii. Check sparsity of the vector created with the transformation step i using Eq.(2), Eq.(3)
    iii. Compact the sparse data values using Eq.(6)
    FOR Feature 1 to n LOOP
      IF feature NON-ZERO AND NOT NULL
        CompactFeature = featureIndex:feature
      ELSE
        SKIP VALUE
      NEXTVALUE
    ENDLOOP
  c. IF (more features in the row) Goto step-a
# Compact complete Row using compact values from Step a-c
  CompactRow = EMPTY
  FOR CompactFeature 1 to n LOOP
    CompactRow = CompactRow + SPACE + CompactFeature
    NEXTVALUE
  ENDLOOP
  CompactRow = ClassLabel + SPACE + CompactRow
# Map and reduce tasks are used for processing and resource manager manages the processing jobs.
# IF (more Row) from any worker-node Goto Step-4 ELSE FINISH
```

Source data can be represented in a two-dimensional matrix: $D_{S\,=}$ $[d_{ij}]$ where D_S is reference data and having i attributes from 1 to n and j are rows from 1 to m. Feature reduction is done using Singular Value Decomposition (SVD which is a well-known method used for dimensionality reduction). SVD factorizes a matrix into three matrices: U, Σ, and V.

$$A = U\Sigma V^T \tag{1}$$

where U is an orthonormal matrix, Σ is a diagonal matrix with non-negative diagonals in descending order, V is an orthonormal matrix and V^T is the conjugate transpose of V. Sparsity of a vector or matrix can be represented as:

$$V^S = \sum\nolimits_{1(k=0)}^{n} / \sum\nolimits_{1}^{n} \tag{2}$$

where sparsity is the ratio of the sum of attributes of a vector V from 1 to n having value k = 0 to the total attribute values. The sparsity can also be represented as (3), which is 1 minus, the sum of the number of attributes which are non-zero.

$$V^S = 1 - \sum\nolimits_{1(m \neq 0)}^{n} \tag{3}$$

where m are the attribute values, which are non-zero.

3.2 Data Blocks

When a file is stored in Hadoop [22] Distributed File System (HDFS), the system breaks it down into an individual blocks set and stores these blocks in multiple slave nodes (worker-nodes) in the Hadoop cluster. Rows division in each data block can be calculated with (4).

$$\mathrm{Rows}^{\mathrm{Block}} = \sum \mathrm{Rows}/\mathrm{WorkerNodes}/\mathrm{DataBlockSize}/\mathrm{RowDataSize} \tag{4}$$

3.3 Transformation with OHE-E

One-hot Encoding Extended (OHE-E) is a technique developed in this paper, which transforms categorical attributes to numeric attributes with an extra attribute. Missing value imputation (MVI) is applied with majority value of a given attribute for selected attributes. Transformation with One-hot Encoding Extended with an extra attribute is explained in (5).

$$E^{ohe-e} = \mathrm{fTrans}\left(A^d\right) \tag{5}$$

where E^{ohe-e} is One-hot Encoding Extended (OHE-E) format and A^d is attribute with d predefined distinct values and fTrans is transformation function of OHE-E. fTrans(A^n) function transforms a selected and categorical attribute A with n observations and d distinct attribute values, to d +1 binary attributes with n observations each. Each observation indicating the 1 as true or 0 as false of the dth + 1 binary variable. The dth + 1 variable will be true if an attribute value is not from the predefined attributes values. The extra attribute is only included if there is a possibility of new values from previously known values. The decision of taking extra attribute is configured in various contextual and model-based profiles. It is evaluated with different measures including; ratio of total d distinct values of an attribute with n observations. Threshold applied in bank dataset is 0.005. Another measure is time-bound attribute values. For example, in a banking application, the types of transactions can be enumerated in advance, but other attributes such as the device or browser being used may continue to exhibit novel values over time as technology changes.

3.4 Compactness with OHE-EC

Transformation with conventional OHE method makes the data sparse, so compactness of data is suggested and applied in this paper. Compactness on sparse data is applied by omitting all zero and empty attributes values in an instance and keeping the remaining attribute values along with the attribute index. Compactness is explained in (6).

$$\mathrm{C}^{\mathrm{ohe-ec}} = \mathrm{f}Compact \int_{i}^{{}_{1}^{n}Y} (X)\, m \neq 0 \tag{6}$$

Where X is $\mathrm{E}^{\mathrm{ohe-e}}$ format data from (5) and $\mathrm{C}^{\mathrm{ohe-ec}}$ is the OHE Extended Compact format and fCompact is a function to compact a row y with only selecting attributes from 1 to n on i^{th} index having m value which is non-zero. Empirical evaluation has shown that after compacting data with OHE-EC, size could be 3x smaller from OHE format.

3.5 Sample Datasets Formats

A sample of the mixed datasets is explained by [21], Table 1 shows sample data, in OHE format for categorical attributes; Transaction Type (BPay and PA), Account Type (Credit, Personal), Browser (Alt, Moz4, Browser New) and Country (AU, NZ, Country. New), while Table 2 shows compact OHE format for same data in Table 1. Compacting process is explained in (6).

Table 1. One-hot Encoding extended dataset.

Class	Bpay	PA	Amount	Credit	Personal	Login	Password	Alt	Moz 4	Moz 5	Brows. New	AU	NZ	Count. New
1	0	1	8210	0	1	5	1	0	0	1	0	1	0	0
0	0	1	5124	0	1	4	1	0	0	1	0	1	0	0
2	0	0	2035	0	1	8	2	0	0	0	0	0	0	1

Table 2. Compact data format.

Class	Attributes
1	2:1 3:8210 5:1 6:5 7:1 10:1 12:1
0	1:1 3:5124 4:1 6:4 7:1 9:1 13:1
2	2:1 3:2035 5:1 6:8 7:2 8:1 14:1

First Come First Serve (FCFS) and High Distribution First (HDF) are two models in this technique. (5) explains that OHE transforms a single variable with n observations and d distinct values, to d + 1 binary variables with n observations each. Each observation indicates the presence 1 or absence 0 of the binary variable. Distribution is calculated for a binary variable having the presence in n observations. In FCFS no sorting is done, but in HDF, the attributes are sorted based on the distribution (higher distribution first). FSFS is efficient in training and testing the model, but it has relatively lower classification accuracy. HDF has better classification accuracy but is little slower in training and testing due to the extra overhead of sorting higher distribution attribute values. Empirical evaluation has shown that if lower distribution attributes are excluded then accuracy with HDF further increases as compared with FCFS.

OHE-EC technique not only reduces dataset size, but gives better performance also in terms of classification accuracy and time (especially on hadoop multi-node cluster), and data can also be used in the Classification techniques which use numeric data only.

4 Results

4.1 Synthetic Bank Transaction Dataset

A synthetic dataset based off actual bank transaction data was generated using the HCRUD technique [21]. Comparison of classification accuracy with synthetic generated mixed data (generated by HCRUD), and numeric data (converted by OHE) is shown in Tables 3 and 4 for different classification algorithms. Training and test data split ratio is 70% and 30% respectively and average results are taken.

Table 3. Accuracy with mixed datasets.

Random forests	Decision tree	Naïve bayes	SVM	OneVsRest	Instances in dataset
96.02%	97.55%	63.59%	60.99%	62.79%	10,000
97.77%	98.85%	64.39%	61.01%	62.58%	100,000
97.90%	98.84%	64.07%	61.57%	62.96%	1,000,000

Table 4. Accuracy with numeric datasets with OHE.

Random forests	Decision tree	Naïve bayes	SVM	OneVsRest	Instances in dataset
97.93%	97.76%	64.86%	93.60%	94.12%	10,000
98.82%	98.85%	64.05%	93.04%	93.21%	100,000
98.88%	98.82%	63.95%	93.24%	93.66%	1,000,000

Classification accuracy results shown in Tables 3 and 4 depict that classification accuracy is better with numeric data (OHE) as compared with a mixed dataset. A T-TEST was performed to determine whether classification accuracy in Tables 3 and 4 are likely to have come from the same two underlying populations that have the same mean or those values have any significant difference. T-TEST, results prove that the classification accuracy results have significant differences.

First come first serve (FSFS) and High distributions first (HDF) are two further models developed in One-hot Encoding Extended Compact (OHE-EC) technique. Tables 5 and 6 show a comparison of classification accuracy with these two models.

Table 5. OHE-EC (FCFS).

Random forests	Decision tree	Naïve bayes	Instances in dataset
97.97%	97.67%	64.77%	10,000
98.84%	98.62%	63.98%	100,000
99.02%	98.95%	63.83%	1,000,000

Table 6. OHE-EC (HDF).

Random forests	Decision tree	Naïve bayes	Instances in dataset
98.16%	97.79%	63.29%	10,000
98.92%	98.76%	64.23%	100,000
99.07%	99.07%	63.84%	1,000,000

The classification accuracy results in Tables 5 and 6 suggest that classification accuracy with OHE-EC (HDF) is slightly better than OHE-EC (FSFS). To confirm this a T-TEST was performed on these results. T-TEST results for Random Forests, Decision Tree and Naïve Bayes are 0.6075, 0.5162 and 0.2113 respectively, indicating that the observed differences between OHE-EC (HDF) and OHE-EC (FCFS) with regards to classification accuracy are not statistically significant.

Other than the classification accuracy, one measure was to compare model's training and perdition time with OHE and OHE-EC. Figure 1 shows training and prediction improvement with OHE-EC in terms of the time.

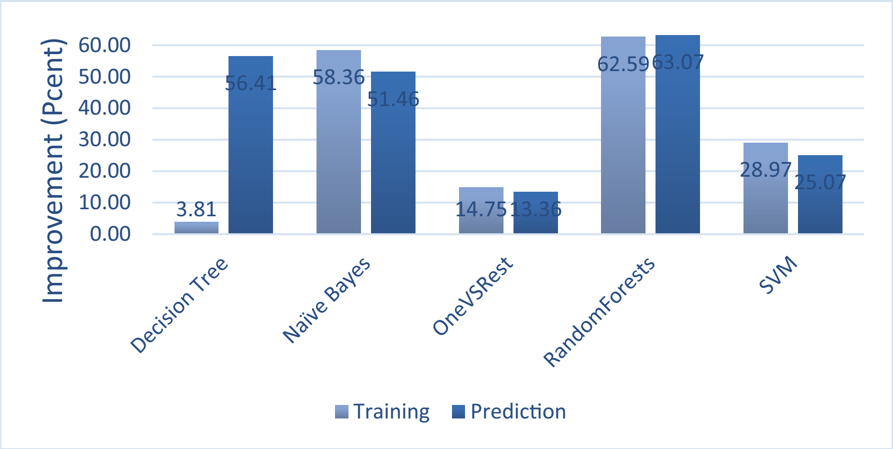


Fig. 1. Average train/prediction time improvement with OHE-EC.

X-axes in the above figure are the classifiers. Y-axes is the average improvement time for different dataset size ranging from very small to large datasets. Results show that there is significant improvement in training and prediction times of the models with OHE-EC. Another empirical evaluation was done with larger datasets only. Figure 2 shows that improvement in prediction time is higher than the training time with larger datasets in almost all classifiers other than Random Forests.

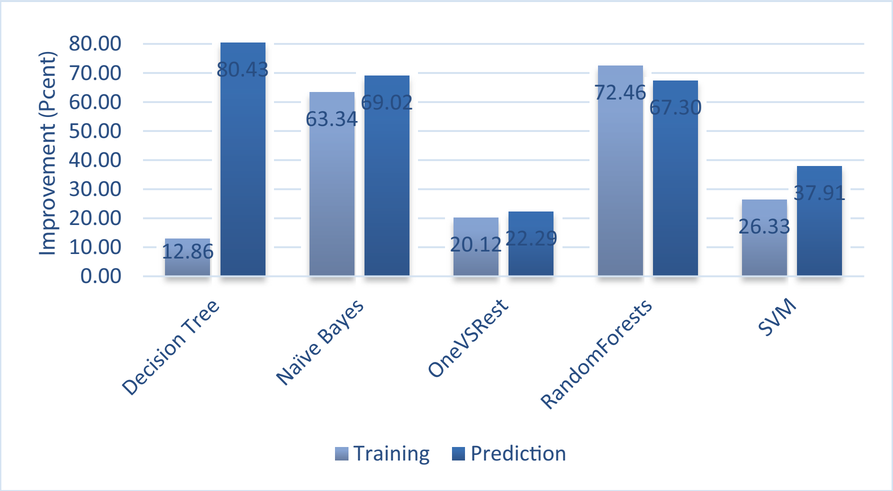


Fig. 2. Large data train/prediction time improvement with OHE-EC.

4.2 KDD Cup Data

The proposed technique was also tested on a KDD-99, a widely used publicly available datasets for anomaly detection [14]. The current datasets contain more than 65 distinct attributes values in service attribute. There is a high possibility that there is new service in the data. One-hot Encoding Extended can transform the row to OHE-E as it is using one extra attribute for new attribute values. Table 7 shows a comparison of classification accuracy with 10 million instances of KDD-99 datasets.

Table 7. Comparison of performance of various classifiers on the KDD-99 dataset.

Random forests	Decision tree	Naïve bayes	SVM	Format	Model
99.973%	99.920%	93.043%	99.991%	Mixed	
99.986%	99.997%	93.711%	99.990%	OHE	
99.99%	99.993%	93.265%	99.997%	OHE-EC	FCFS
99.993%	99.993%	93.463%	99.999%	OHE-EC	HDF

Datasets size of different formats including synthetic data of mixed data and data generated by OHE and OHE-EC were compared. It was observed that datasets size is smallest with OHE-EC, as an average the data in OHE-EC is 3x reduced from OHE. Classification accuracy with OHE-EC with HDF model is also slightly better as compared to the mixed dataset, OHE and OHE-EC (FCFS). Model training and prediction time is also improved with OHE-EC.

5 Conclusion

Fraud detection for online banking is an important area of research, but the heterogeneous nature of data (i.e. mixed data) is challenging. Numeric format data is known to give better performance with classification and some machine learning platforms such as Apache Spark by default only accept numeric data. One-hot Encoding (OHE) is a widely used approach for transforming categorical features to numerical features, but in various datasets, the distinct values of an attribute are not always known in advance. Also, the sparseness of the transformed data is another challenge. Due to growing memory and storage consumption needs; compactness of machine learning models has become much more critical. An innovative technique is presented in this paper to transform categorical features to numeric features by compacting sparse data even when all the distinct values are not known. Results produced by this technique are demonstrated on synthetic and real bank fraud data and anomaly detection KDD-99 datasets on multi-node hadoop cluster. The empirical results show that One-hot Encoding Extended (OHE-E) gives improvements over mixed datasets and One-hot Encoding Extended compact (OHE-EC) not only gives further improvement in reducing the size of datasets, but also an improvement in model's training and prediction time. Two further models OHE-EC (FCFS) and OHE-EC (HDF) are also developed in One-hot Encoding Extended Compact (OHE-EC) technique, where OHE-EC (HDF) gives slightly better classification accuracy as compared to OHE-EC (FCFS).

One of the recommended future work is to test this technique on high dimensional data having and datasets with categorical attributes having a higher number of distinct values.

References

1. Breunig, M.M., Kriegel, H.-P., Ng, R.T., Sander, J.: LOF: identifying density-based local outliers. In: ACM Sigmod Record (2000)
2. Hodge, V., Austin, J.: A survey of outlier detection methodologies. Artif. Intell. Rev. **22**(2), 85–126 (2004)
3. Jin, H., Chen, J., He, H., Kelman, C., McAullay, D., O'Keefe, C.M.: Signaling potential adverse drug reactions from administrative health databases. IEEE Trans. Knowl. Data Eng. **22**(6), 839–853 (2010)
4. Maruatona, O.: Internet Banking Fraud Detection Using Prudent Analysis. University of Ballarat, Ballarat (2013)
5. Zhang, Y., Meratnia, N., Havinga, P.: Outlier detection techniques for wireless sensor networks: a survey. IEEE Commun. Surv. Tutor. **12**(2), 159–170 (2010)
6. Zhang, K., Jin, H.: An effective pattern based outlier detection approach for mixed attribute data. In: Li, J. (ed.) AI 2010. LNCS (LNAI), vol. 6464, pp. 122–131. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17432-2_13
7. Shih, M.-Y., Jheng, J.-W., Lai, L.-F.: A two-step method for clustering mixed categroical. Tamkang J. Sci. Eng. **13**(1), 11–19 (2010)
8. Huang, Z.: Clustering large data sets with mixed numeric and categorical values. In: Proceedings of the 1st pacific-asia conference on knowledge discovery and data mining, (PAKDD) (1997)
9. Pentreath, N.: Machine Learning with Spark, p. 338. Packt Publishing, Birmingham (2015)
10. Meng, X., et al.: Mllib: machine learning in apache spark. J. Mach. Learn. Res. **17**(34), 1–7 (2016)
11. Shanahan, J., Dai, L.: Large scale distributed data science using apache spark. In: 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco (2015)
12. Chen, W.: Learning with Scalability and Compactness, p. 147, Washington (2016)
13. Meng, X.: Sparse data support in MLlib. Apache Spark Community, San Francisco (2014)
14. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the KDD CUP 99 data set. In: IEEE Symposium on Computational Intelligence for Security and Defense Applications 2009. CISDA 2009, Ottawa, Canada (2009)
15. Jian, S., Cao, L., Pang, G., Lu, K., Gao, H.: Embedding-based representation of categorical data by hierarchical value coupling learning. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence (2017)
16. Qian, Y., Li, F., Liang, J., Liu, B., Dang, C.: Space structure and clustering of categorical data. IEEE Trans. Neural Netw. Learn. Syst. **27**(10), 2047–2059 (2016)
17. Boriah, S., Chandola, V., Kumar, V.: Similarity measures for categorical data: a comparative evaluation. In: Proceedings of the 2008 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics (2008)

18. Anderberg, M.R.: Cluster Analysis for Applications. Academic Press, New York (1973)
19. Hartigan, J.A.: Cluster Algorithms, vol. 214, p. 1993. Wiley, New York (1975)
20. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice-Hall, NJ (1988)
21. Ul Haq, I., Gondal, I., Vamplew, P., Layton, R.: Generating synthetic datasets for experimental validation of fraud detection. In: Fourteenth Australasian Data Mining Conference, Canberra, Australia. Conferences in Research and Practice in Information Technology, vol. 170, Canberra (2016)
22. Apache Software Foundation: Apache Hadoop, 26 April 2015. http://hadoop.apache.org/

Transport, Environment and Energy

Combining Machine Learning and Statistical Disclosure Control to Promote Open Data

Nasca Peng

Statistics New Zealand, Wellington 6011, New Zealand
Nasca.Peng@stats.govt.nz

Abstract. We proposed machine learning-oriented statistical disclosure control as a novel solution for New Zealand Transport Agency to release more person-related variables in its open crash data for privacy-preserving data mining. Instead of making arbitrary decisions in variable aggregation and using perturbation to guard against reidentification attacks at the cost of data distortion, we creatively drew upon feature engineering and dimensionality reduction techniques such as correspondence analysis to make evidence-based data manipulation without distortion. In addition, we built random forest classifiers along the way to directly monitor the impact of our data manipulation on users' modelling, rather than relying on traditional utility metrics such as information loss and difference of eigenvalues that are less interpretable for users. The dataset produced using our method satisfied 10-anonymity based on 11 quasi-identifiers, with less than 3% suppression, compared with only 3-anonymity based on no more than 8 quasi-identifiers with far more than 3% suppression commonly reported in literature. Furthermore, our method enabled random forest classifier to achieve 0.996 for AUC and 0.895 for F-score in predicting crash severity.

Keywords: Statistical disclosure control · Privacy-preserving data mining · Confidentiality · Dimensionality reduction

1 Introduction

1.1 Statistical Disclosure Control

Agencies have long been faced with the dilemma of fulfilling open data obligations while protecting privacy of individuals. Statistical disclosure control (SDC) refers to techniques used to ensure that no individual is identifiable from the released microdata [1]. It is considered as an issue of optimisation between data safety and utility, which are quantifiable by metrics such as k-anonymity [2] and special unique detection algorithm (SUDA) scores [3] and information loss [4].

SDC techniques can be classified into aggregation, suppression and perturbation, each subject to some extent of criticism. Aggregation seeks to reduce the granularity of information and therefore reduce the likelihood of exposure at observation level, but it typically involves arbitrary decisions and can be time-consuming. Suppression tends to cause over protection with significant compromise of utility. Perturbation seeks to create uncertainty to confuse data intruders at the cost of data distortion.

R. Islam et al. (Eds.): AusDM 2018, CCIS 996, pp. 83–93, 2019.
https://doi.org/10.1007/978-981-13-6661-1_7

To avoid such caveats, we decided to abandon perturbation and instead use data mining techniques to assist the decision making in aggregation and reduce the burden of suppression. To evaluate our solution, k-anonymity was selected as the safety metric because of easy interpretability and implementation using sdcMicro developed by [5]. To gauge the utility, we not only assessed the number of person-related variables added and amount of suppression, but also used performance metrics of classifiers to monitor the impact of our data manipulation on users should they build models to predict likely target variables.

1.2 Data Description

Crash Analysis System (CAS) is New Zealand's primary tool for capturing information on where, when and how road crashes occur [6]. The current open dataset consists of 645,798 instances, representing crashes from the year 2000. The majority of 88 variables are environment-related, with minimal information about drivers. However, because of frequent requests from users, we think it is necessary to include some important person-related variables in the dataset for greater utility.

To explore such possibility while preserving privacy, an extended CAS dataset is used for this study. It consists of 172,459 instances and 162 attributes, covering crashes as reported to the New Zealand Transport Agency from the year 2017 to 2018. Not all crashes are reported, and the level of reporting increases with the severity of the crash. Most attributes are categorical. The few exceptions are locations, such as street names and drivers' home countries, and their numeric encodings, such as easting, northing and area unit IDs.

The features represent three types of measurements – non-personal features are generally descriptive of the environment conditions such as road curvature; behavioural features assess drivers' deeds such as sudden actions, alcohol or drug intake, etc.; quasi-identifiers [2] represent variables that might be used to reidentify individuals, such as demographics, locations and time. Since no data dictionary is available, the determination of quasi-identifiers is mainly based on variable names. Our main goal for the open data solution is to release more behavioural features and quasi-identifiers, which are barely present in the current public CAS data.

Our machine learning target variable is set as crash severity, which include four classes, F for fatal, S for serious, M for minor and N for non-injury. Other possible target variables include injury severity and injured party counts, but they are highly inter-related, so we only included crash severity in feature selection and modelling. Since it is not feasible to cover all use cases in a single open data file, the target selection is based on the most generic sense of machine learning application and has been confirmed in stakeholder consultation.

It is worth noting that this dataset is highly imbalanced – the number of fatal and serious crashes is less than one tenth of that of minor and non-injury crashes. Either oversampling or undersampling is required for modelling.

2 Experimental Design and Discussions

2.1 Workflow

Figure 1 depicts our workflow, which consists of four stages – feature selection, feature engineering, modelling and risk reduction. Efforts are made to strike the balance between safety and utility throughout all stages.

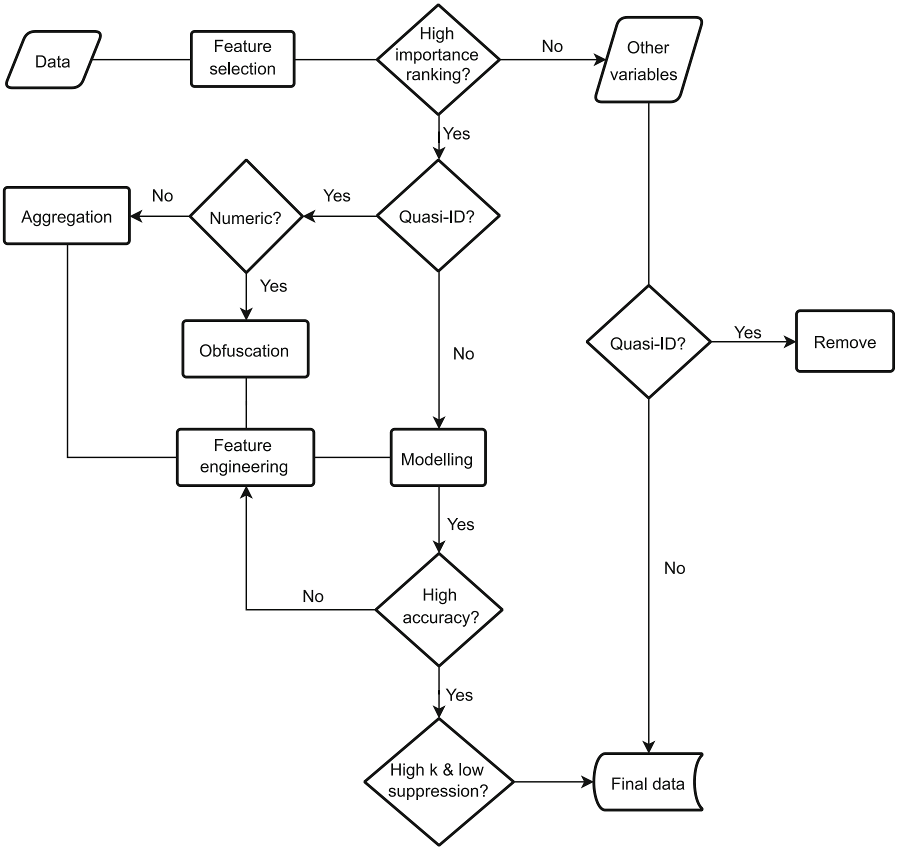


Fig. 1. Flowchart of the proposed solution.

Stage 1: Boruta [7] was used as a wrapper method as it could handle both categorical and numeric variables. In addition, SUDA was used to gauge variable contribution to record uniqueness. Quasi-identifiers with low Boruta importance scores and high SUDA risk scores were discarded. Unimportant non-personal variables were excluded from modelling but nevertheless kept in the final data as they pose minimal threat to reidentification. The remaining variables were important features with either high or low SUDA scores.

Stage 2: While feature engineering is generally intended to boost model performance, in this application it was conducted solely on quasi-identifiers to reduce their granularity or interpretability, since their contribution to reidentification risk was our major concern.

Stage 3: Random forest [8] was chosen as the classifier because of its inherent optimisation (bagging), fast computation and minimal requirement for data pre-processing. We used its performance to adjust feature engineering and the basket of quasi-identifiers. We would not progress further after a suitable threshold of classification accuracy was met, because our main goal was to produce a safe and useful microdata product rather than optimising the model.

Stage 4: We assessed k-anonymity based on the basket of quasi-identifiers. If the proportion of observations violating 2-anonymity was within 2%, it would take reasonably low suppression to achieve 5- or higher-anonymity. Otherwise, we would return to Stage 2 to seek further options of dimensionality reduction.

2.2 Feature Selection

Table 1 is a descriptive overview of some quasi-identifiers, alongside with a comparison between feature importance and risk contribution. According to Boruta output, top 50% features include 33 non-personal features, plus 15 quasi-identifiers and 18 behavioural attributes. We noticed that many top-ranking quasi-identifiers are granular location variables with high SUDA scores, indicating the necessity of feature engineering for privacy protection.

Table 1. Variable description metrics.

Variable name	Boruta score	SUDA score	Number of levels	Mean size
Crash location 2	58	37	15365	11
Crash location 1	34	20	7826	22
Area unit ID	34	7	1741	99
MB ID	33	38	16359	11
Licence	30	7	9	19162
Sex	23	5	4	43115
Territorial authority ID	21	1	67	2574
Age grouped	20	7	6	28743
CT party type	19	9	12	14372
Person party type	14	7	8	21557
Driver CULP	9	13	6	28743

When all features were fed into random forest without pre-processing, AUC was misleadingly high at 0.9 whereas F-score was 0.33, because the data was highly disbalanced. To establish a modelling baseline, we collapsed crash severity from four levels to two – class 0 (non-severe) and class 1 (severe). We used Synthetic Minority Over-Sampling Technique (SMOTE) [9] to oversample class 1 by 200% based on three nearest neighbours, with 33 top non-personal features as input. Random forest achieved 0.98 for AUC and 0.77 for F-score. We found that adding person-related variables, especially locations, in the raw form worsened F-score, hence the necessity of feature engineering for utility as well.

2.3 Feature Engineering

Correspondence Analysis. To reduce dimensions, we calculated eigenvalues to examine the proportion of variances retained by different axes. To unravel the correlation between different levels of categorical features and target variables, we created contingency tables and plotted columns (target variables) in standard coordinates and rows (features) in principal coordinates [10, 11]. We think that the resulting asymmetric biplot can make aggregation more evidence-based.

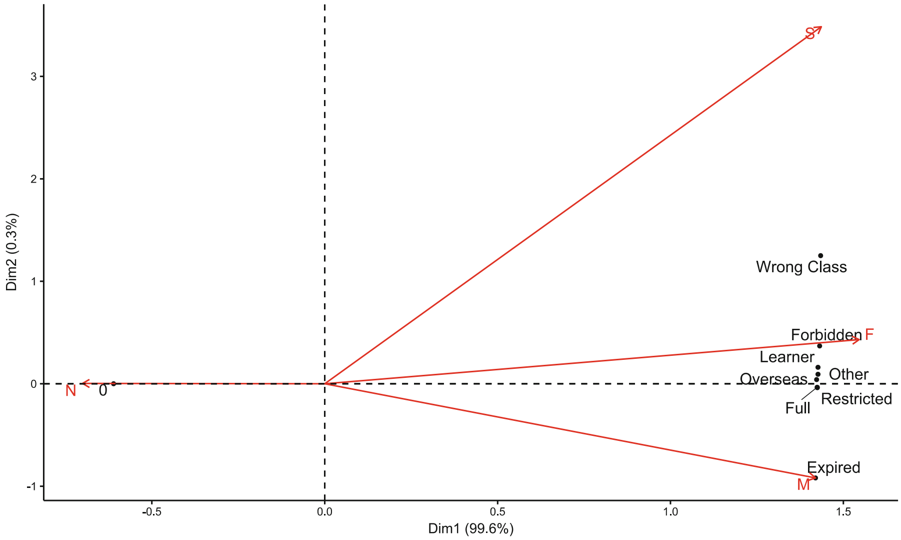


Fig. 2. Asymmetric biplot of cross tabulation between driver licence and crash severity (Color figure online)

Figure 2 helps us understand the correlation between driver licences (black dots) and crash severity (red arrow). The closer the dot is to the arrow, the stronger association there is between the relevant licence and crash severity. It shows that “0” (licence not reported) is highly correlated with non-injury crashes, echoing with the fact that the level of reporting decreases as the severity decreases. Meanwhile, both wrong class and forbidden licence are located between fatal (F) and serious (S) crashes. Therefore, the collapsing of these two licences can not only reduce variable granularity and contribution to reidentification risk, but also enable the classifier to be biased towards the minority class. In fact, it turned out to boost the F-score.

We can also see that Dimension 1 explains 99.6% of the total inertia retained by the dimensions. The higher the retention, the more subtlety in the original data is retained in the low-dimensional solution [11]. It suffices to keep only this dimension since others explain less than 1% of the total inertia.

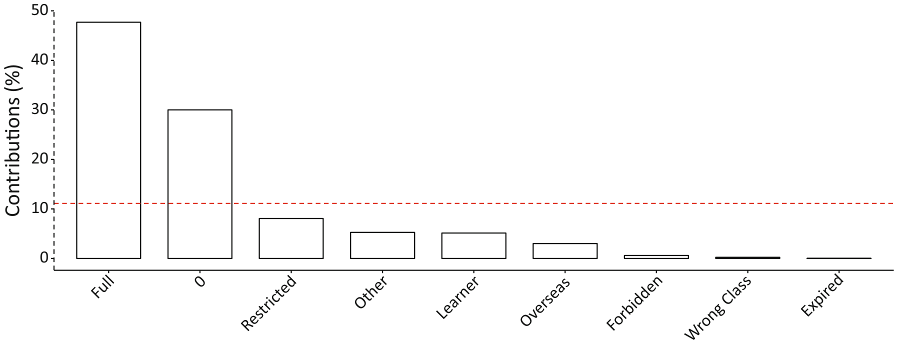


Fig. 3. Contribution of different driver licences to Dimension 1 (Color figure online)

The row contribution to the principal axis can be used to determine the cap of aggregation, as shown in Fig. 3. It indicates that full licence contributes most to Dimension 1, while the red dash indicates the expected average value if contributions are uniform [10]. Levels below the dash can be dropped or aggregated. In this case, full licence will be kept intact, whereas learner and overseas licences can be collapsed into "other" because of their proximity (see Fig. 2) and small collective contribution. This strategy also boosted F-score. All other categorical quasi-identifiers were processed based on the same criteria.

Meanwhile, some features are already in an aggregated format. Notably, the variable age has been collapsed into wide bands such as 25–54, rather than the common 5-or-10-year bands, indicating that the original aggregation was solely based on privacy protection. We can examine its impact on classification from the figure below.

Figure 4 shows that all age groups are cluttered in the same area and therefore will not help to discriminate crash severity. Since the inclusion of age did not boost model performance, the risk caused by its public release is likely to loom larger than gains. We think that it is necessary to redesign its original aggregation scheme.

String Clustering. Driver's country of origin is one of the text variables in need of aggregation because there are over 70 unique country names. String clustering algorithms normally group strings based on similarity. It is likely to work better for occupations where words sharing similar components such as "project manager" and "product manager" can be conveniently clustered, but here such a scheme can result in countries, say Samoa and Canada, that differ greatly in terms of socio-economic spectrums, to be grouped in the same cluster simply because of spelling. It turned out that this scheme did not lead to the improvement of F-score.

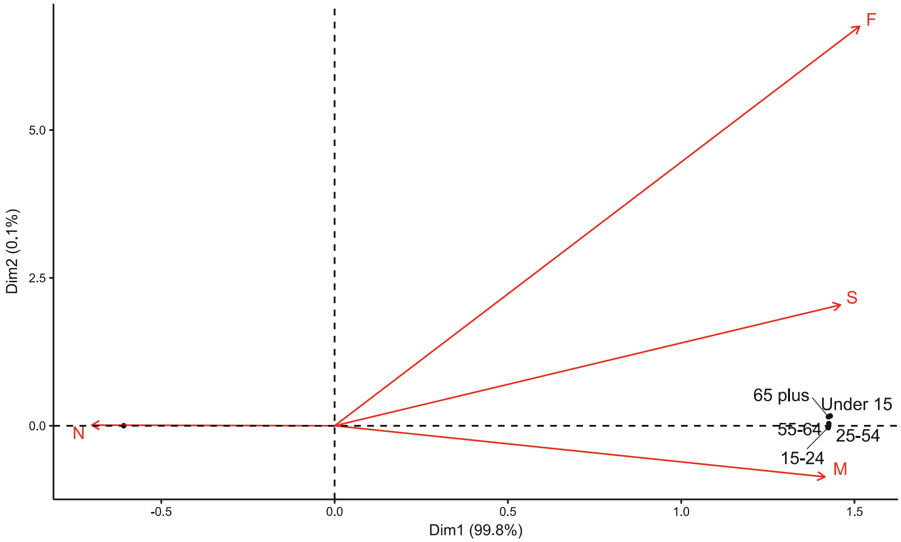


Fig. 4. Asymmetric biplot of cross tabulation between age group and crash severity

Another attempt was made to aggregate crash location 2 where places with matching street names are clustered based on Jaro-Winkler distance [12, 13]. However, no significant boost of the k value can be achieved without clustering beyond the street level because the number of crashes in specific streets is generally small. Yet more aggressive clustering will lead to the same dilemma as above, so we think this method originally intended for record linkage may not be suitable for our scenario.

Principal Component Analysis (PCA). We noticed that most of top person-related features are location-specific. It not only makes observations easier to be reidentified but is likely to bring redundant information to the classifier and cause overfitting. Therefore, we applied label encoding and PCA to such variables as territorial authority ID, area unit ID, MB ID, crash location 1, crash location 2 and driver's country of origin. They are reduced to two dimensions, resulting in less interpretability but greater safety and information efficiency. Since the retention of the original form will cause far greater suppression and more compromise of utility, we think it is an acceptable trade-off. Besides, aggregated territorial authority ID is still included in the final data for user interpretation apart from the PCA dimensions.

Latent Dirichlet Allocation (LDA). LDA assumes that each document can be characterised by a set of topics. It uses a generative approach to identify topics based on the likelihood of word co-occurrence [14]. We used LDA to model crash locations 1 and 2 with the following parameter settings: number of topics: 5; N-grams: 2; normalisation: true; dictionary of N-grams: true; and maximum size of dictionary: 20,000. It turned out to boost F-score. We think the reason why LDA worked better than string clustering here is because it factors into the context rather than mere literal similarity. Like PCA, LDA will result in less interpretability but can nevertheless serve as an artificial proxy of original variables whose retention will cause over suppression.

Other Transformation. We used regularised incomplete gamma function to process numeric encodings of locations because gamma functions are commonly used in LDA [14]. We squared the output columns of PCA to strengthen the location information. Many other methods such as summary statistics and special functions have not been explored, so it is left to data users to apply novel ways for greater model performance.

2.4 Modelling

The Table 2 above shows the performance of typical modelling sessions. We used stratified split and included 75% of the data in the training set. Top 33 non-personal features were included across all sessions. We tried to add and engineer high-ranking person-related variables one by one to raise F-score from 0.77 to 0.895, an increase of over 10%, indicating significantly greater robustness in crash severity classification compared with the baseline.

Table 2. Model performance in different sessions.

SMOTE KNN	Feature engineering	AUC	F-score
None	None	0.90	0.33
3	None	0.98	0.77
3	Licence	0.98	0.77
3	Licence (aggregated)	0.982	0.783
4	Licence (aggregated)	0.982	0.795
4	Above + alcohol/drug	0.988	0.814
3	Above + area unit ID (aggregated)	0.988	0.843
3	Above + disability/illness	0.988	0.84
3	Above + sudden action	0.991	0.846
3	Above + location (PCA)	0.991	0.86
3	Above + location (Gamma regularised P)	0.992	0.871
4	Above + CT party role	0.992	0.879
4	Above + location (squaring PCA dimensions)	0.993	0.88
4	Above + location (LDA)	0.995	0.889
4	Above + vehicle manufacture year (aggregated)	0.996	0.895

2.5 Risk Reduction

Table 3 helps us monitor re-identification risks throughout the process. Almost all initial records violated 2-anonymity. Feature engineering not only improved model performance but resulted in significant risk reduction – only 1% of records violating 2-anonymity, making the burden of suppression much lower, as seen below.

Table 3. Number of records violating k-anonymity during different phases of workflow.

Number of observations violating	Original data	After feature engineering	After suppression
2-anonymity	165,497 (96%)	1,807 (1%)	0 (0%)
3-anonymity	170,745 (99%)	3,673 (2%)	0 (0%)
5-anonymity	172,204 (99.9%)	6,165 (4%)	0 (0%)

Table 4 shows that only around 2–3% suppression is needed for 10-anonymity of as many as 11 quasi-identifiers, indicating compliance of stringent safety and utility standards.

Table 4. Amount of suppression to achieve 5-anonymity after dimensionality reduction.

Quasi-identifiers	Amount of suppression for 5-anonymity	Amount of suppression for 10-anonymity
Disability/illness	6 (0.003%)	4 (0.002%)
CT party role	0 (0%)	0 (0%)
Territorial authority ID	2031 (1.178%)	3762 (2.181%)
Licence	73 (0.042%)	124 (0.072%)
Driver CULP	20 (0.012%)	21 (0.012%)
Person PROT	435 (0.252%)	841 (0.488%)
Person role	4 (0.002%)	8 (0.005%)
Person sex	133 (0.077%)	319 (0.185%)
Person type	7 (0.004%)	21 (0.012%)
Person party role	285 (0.165%)	554 (0.321%)
Crash year	1 (0.001%)	1 (0.001%)

3 Conclusion

Based on the research above, we make the following recommendations to create an alternative open dataset:

- A total of 11 quasi-identifiers in engineered forms can be included with 5- or higher- anonymity and 2–3% suppression.
- Since a stringent safety standard is satisfied, 18 behavioural variables can be added to the public domain.
- Non-personal variables can be kept as is.

- Considering a trade-off between location granularity and addition of person-related variables, the proposed dataset enables 25 more variables to be released compared with the existing open data.
- The proposed dataset allows users to capitalise on a new modelling baseline of 0.996 for AUC and 0.895 for F-score and conduct more in-depth research on the interactions of new variables to inform policy changes.

In terms of the method:

- We used mathematical obfuscation such as PCA and LDA to turn location-related variables into artificial proxies to balance anonymity with utility. Meanwhile, an aggregated form of territorial authority ID was kept for the retention of interpretability.
- Correspondence analysis was applied to make the aggregation of categorical quasi-identifiers more evidence-based and less time-consuming.
- Data mining techniques proved to be useful in guide the decision making of SDC applications so that the reduction of granularity and interpretability boost the modelling performance and k-anonymity simultaneously while keeping the level of suppression to the minimum.

4 Limitation and Future Work

We did not consider l-diversity [15] beyond k-anonymity but can integrate it in future work. K-anonymity is dependent on the setting of quasi-identifiers. In this case, we tried to be inclusive in the initial determination of quasi-identifiers. Also, a significantly higher k than commonly seen in literature also helps guard against potential limitation of k-anonymity.

This method can also be biased on our perception about the general purpose of the dataset. While it is justifiable here to choose crash severity from possible target variables that are both few and inter-related, it can sacrifice some utility when users' needs are specific. For example, because location variables are less interpretable or more aggregated in exchange for a larger number of person-related features, it might not serve a researcher's need to monitor whether a change of alcohol test policy has resulted in reduced crashes at a specific street.

In this case, we propose a hierarchical approach. The final dataset produced in this research can serve as a generic dataset for unrestricted public access. Meanwhile, this method can still be used to make a customised de-identified product quickly for streamlined researcher access and less restrictive usage, i.e., allowing users to manipulate data on their own devices rather than in a controlled lab environment. It will help accelerating the data lifecycle for policy advocacy and promote privacy-preserving information transparency.

References

1. Skinner, C.: Statistical disclosure control for survey data. In: Handbook of Statistics, pp. 381–396. Elsevier, Amsterdam (2009)
2. Sweeney, L., Samarati, P.: Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression, Computer Science Laboratory, SRI International (1998)
3. Elliot, M.J., Manning, A., Mayes, K., Gurd, J., Bane, M.: SUDA: a program for detecting special uniques. In: Joint UNECE/Eurostat Work Session on Statistical Data Confidentiality, Geneva (2005)
4. Mateo-Sanz, J.M., Domingo-Ferrer, J., Sebé, F.: Probabilistic information loss measures in confidentiality protection of continuous microdata. Data Min. Knowl. Discov. **11**(2) 181–193 (2005)
5. Templ, M., Kowarik, A., Meindl, B.: Statistical disclosure control for micro-data using the R package sdcMicro. J. Stat. Softw. **67**(1), 1–36 (2015)
6. NZTA, Crash Analysis System (CAS). https://www.nzta.govt.nz/safety/safety-resources/crash-analysis-system
7. Kursa, M.B., Rudnicki, W.R.: Boruta. https://cran.r-project.org/web/packages/Boruta/index.html
8. Ho, T.K.: Random decision forests. In: Proceedings of 3rd International Conference on Document Analysis and Recognition (1995)
9. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357(2002)
10. Kassambara, A.: Practical Guide to Cluster Analysis in R: Unsupervised Machine Learning, STHDA (2017)
11. Bendixen, M.: A practical guide to the use of correspondence analysis in marketing research. Mark. Bull. **14**, 16–38 (2003)
12. Jaro, M.A.: Advances in record linkage methodology as applied to the 1985 census of Tampa Florida. Stat. Med. **14**, 491–498 (1989)
13. Winkler, W.E.: String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. In: Proceedings of the Section on Survey Research Methods (1990)
14. Blei, D.M., Ng, A.Y., Jordan, M.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022(2003)
15. Machanavajjhala, A., Gehrke, J., Kifer, D., Venkitasubramaniam, M.: L-diversity: privacy beyond k-anonymity. In: 22nd International Conference on Data Engineering (ICDE 2006) (2006)

Predicting Air Quality from Low-Cost Sensor Measurements

Hamish Huggard[1], Yun Sing Koh[1](✉), Patricia Riddle[1],
and Gustavo Olivares[2]

[1] The University of Auckland, Auckland, New Zealand
hamishhuggard@gmail.com, {ykoh,pat}@cs.auckland.ac.nz
[2] National Institute of Water and Atmospheric Research, Auckland, New Zealand
Gustavo.Olivares@niwa.co.nz

Abstract. Urban air pollution poses a significant global health risk, but due to the high expense of measuring air quality, the amount of available data on pollutant exposure has generally been wanting. In recent years this has motivated the development of several cheap, portable air quality monitoring instruments. However, these instruments also tend to be unreliable, and thus the raw measurements require preprocessing to make accurate predictions of actual air quality conditions, making them an apt target for machine learning techniques. In this paper we use a dataset of measurements from a low cost air-quality instrument—the ODIN-SD—to examine which techniques are most appropriate, and the limitations of such an approach. From theoretical and experimental considerations, we conclude that a robust linear regression over measurements of air quality metrics, as well as relative humidity and temperature measurements produces the model with greatest accuracy. We also discuss issues of concept drift which occur in this context, and quantify how much training data is required to strike the right balance between predictive accuracy and efficient data collection.

Keywords: Air quality · Polynomial regression · Concept drift

1 Introduction

The term particulate matter (PM) describes the aerosols (solid or liquid) suspended in the air, and is a key pollutant in urban areas. PM has been linked to a range of health issues and the World Health Organization has estimated that PM causes 6.4 million years of life lost every year globally [3]. Understanding and quantifying exposure to air pollutants are essential in many human health applications, including risk assessment and accountability evaluations [9].

Due to the high cost of installing and operating accurate air pollution sensors [10], measurements of PM concentrations tend to be sparse in space, and often sparse in time. More precise knowledge of spatiotemporal PM distributions would therefore be of benefit to public health. In recent years, considerable effort has been

R. Islam et al. (Eds.): AusDM 2018, CCIS 996, pp. 94–106, 2019.
https://doi.org/10.1007/978-981-13-6661-1_8

put in the development of portable, low-cost air pollution monitoring instruments [12]. This reduction in cost allows for more measurements to be collected over a given region of space, but also means that each individual measurement is less accurate. Determining how to extract useful measurements and uncertainties from the raw measurements of these low cost sensors would therefore be a useful application of machine learning.

To illustrate this type of application, we examine a dataset generated from a pilot study by the National Institute of Water and Atmospheric Research (NIWA). For this study, a fleet of 18 low-cost air pollutant monitoring instruments called Outdoor Dust Information Nodes - Size Distribution (abbreviated ODIN-SD or ODIN) were deployed throughout Rangiora, a town in the Canterbury plains, New Zealand during the Winter of 2016 (illustrated in Fig. 1). Due to the unknown accuracy of the ODINs, for stretches of the experiment they were colocated with a TEOM-FDMS (a regulatory grade instrument used for official measurements) at Environment Canterbury's (ECan's) air quality monitoring site in central Rangiora so that the two sets of measurements could be compared. Figure 2 illustrates when each ODIN was at the ECan site. These periods provide us with labeled datasets from which we can build models to predict actual PM concentrations from raw ODIN measurements.

The key contributions of this research are:

- We argue that situations such as this are highly adversarial to machine learning techniques: the data are noisy, multi-dimensional, and subject to both real and virtual drift. It is therefore unreasonable to expect to do better than robust linear regression models, and indeed we demonstrate that such models outperform several more sophisticated models.
- By considering feature selection, seasonal variation, and concept drift detection we are able to recommend a modelling technique for the ODIN-SD instrument. The ODIN has since been deployed to collect data in Idaho, Otago, and Gisborne. Having an appropriate modelling technique at hand will therefore assist with future PM modelling work.
- We estimate the error in PM predictions based on when and how much training data is collected. This should allow more strategic deployment of ODINs in the future for more efficient data collection.

2 The Dataset

The ODIN is equipped with a Plantower PMS3003 dust sensor and a DHT22 temperature and relative humidity sensor. The dust sensor measures three quantities: PM_1, $PM_{2.5}$ and PM_{10}. These denote collections of particles which can pass through a size-selective inlet with a 50% efficiency cut-off at 1, 2.5 and $10\,\mu$m aerodynamic diameter, respectively. ODINs sample PM_1, $PM_{2.5}$, PM_{10}, temperature and relative humidity at one minute intervals.

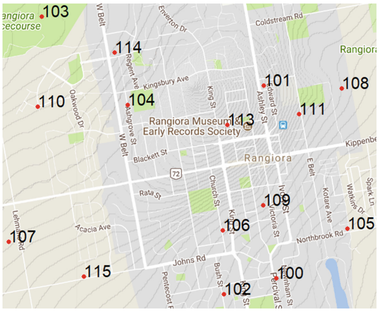


Fig. 1. ODIN deployments in Rangiora overlayed on a Google Map. Locations are labeled by ODIN serial number.

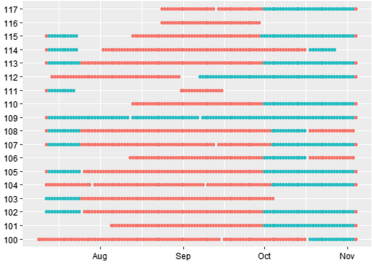


Fig. 2. Timeline of ODIN locations. Blue denotes being located at the ECan site and red denotes deployment. (Color figure online)

The TEOM-FDMS at the ECan site measures $PM_{2.5}$ and PM_{10} concentrations according to New Zealand's regulatory standard (AS/NZS 3580.9.16:2016[1]). We will refer to these measurements as $PM_{2.5,\mathrm{ECan}}$ and $PM_{10,\mathrm{ECan}}$, respectively. ECan data was obtained through ECan's Data Catalogue[2]. These measurements are given as hour averages, whereas ODIN measurements are taken at ten minute intervals, so we needed to apply a sliding window average to the ODIN data for them to be comparable. We investigated whether the ODIN and ECan site clocks were well synchronized and found that displacing the timestamps in either direction tended to reduce the correlation between measurements[3]. A policy of not offsetting ODIN time stamps was therefore chosen.

From Fig. 2 it is apparent that the data requires sanitisation before useful comparisons may be made between ODINs. It will be useful to extract two sub-datasets from the complete set of measurements. ODIN-109 (that is, the ODIN with serial number 109) is at the ECan site throughout the experiment. We therefore call the set of measurements from ODIN-109 the *Single ODIN Dataset*, which will be useful for evaluating ODIN performance over medium-length time horizons. Several of the ODINs were concurrently at the ODIN site for the first 12 days and the last two weeks of the experiment, so we will call this set of ODINs the *Multi-ODIN dataset*. Because these datasets both deal with reasonably small time spans and quantities of data, we are unable to draw any strong or general conclusions about PM prediction policies. Instead we make some practical recommendations for effective use of the ODIN-SD and similar instruments in future PM monitoring work. The main results of this work are in Sect. 5, on model selection, and Sect. 6 on concept drift.

[1] https://shop.standards.govt.nz/catalog/3580.9.16%3A2016%28AS%7CNZS%29/view.

[2] http://data.ecan.govt.nz/Catalogue/Method?MethodId=94.

[3] Our code is available at https://github.com/lajesticvantrashell/ODINs.

3 Related Work

The importance of modelling air quality has attracted much attention from the machine learning community in recent years, and sophisticated air quality models have been developed. These have generally been designed for the scale of large cities, with spatial resolution at the order of $1\,\mathrm{km}^2$ pixels. These have often also included real-time forecasts [8] and even web interfaces [15].

The philosophy of these approaches has been to use sophisticated models to interpolate between the sparse measurements from high-cost air quality sensors. That is not the goal of this paper. Instead, we are concerned with the case where air quality measurements are spatially dense, but from low-cost, inaccurate sensors. For this application, more rudimentary interpolation techniques may be used, so our concern is with the problem of predicting local PM conditions based on the unreliable raw measurements of the sensors.

The Purple Air PA-I, and its successor the PA-II are low-cost air-quality instruments, which, like the ODIN, use a Plantower PMS dust sensors from earlier and later in the design cycle, respectively. These instruments have undergone both lab and field tests [2]. Like many low-cost PM sensors, these tests demonstrated that the Purple Air instruments can measure PM_1 and $\mathrm{PM}_{2.5}$ concentrations much more accurately than PM_{10} concentrations. This is extremely likely to also apply to the ODIN, so this paper focuses on predicting $\mathrm{PM}_{2.5,\mathrm{ECan}}$ (rather than $\mathrm{PM}_{10,\mathrm{ECan}}$) from ODIN measurements, which will probably be more successful. These tests also revealed that at the conjunction of low temperature, high humidity, high PM concentration, the dust sensor accuracy is worst.

4 Accuracy

In this section we explore two questions relating to the accuracy of PM predictions. The first is how much training data do you need to collect to achieve a given level of PM prediction accuracy. The second is whether it matters when in the season the training data is collected.

How Much Training Data Must be Collected? For each ODIN a separate set of training data is required. We therefore have a trade off: the more training data collected per ODIN, the more accurate the model will be. However, more time collecting training data means less time collecting new and useful measurements. In this section we empirically quantify this trade off.

To estimate the error of a model trained on n days worth of data (where $n \in \{1, 2, \ldots, 40\}$, a plausible range of colocation durations), we divided the Single ODIN Dataset into noon-to-noon 24-hour periods $(D_1, \ldots, D_n)$. Then for each $i \in \{1, 2, \ldots, 30\}$, we created a training dataset of n days data, starting with D_i: $Train_{i,n} := \{D_i, D_{i+1}, \ldots, D_{i+n-1}\}$, and a test set of the following month's data: $Test_{i,n} := \{D_{i+n}, \ldots, D_{i+n+29}\}$. We could then build a model from $Train_{i,n}$, and the MSE calculated using $Test_{i,n}$ (this value is then called $\mathrm{Err}_{i,n}$). We then estimated the error of a model trained on n days data as

$$\mathrm{Err}_n = \frac{1}{30}\sum_{i=1}^{30} \mathrm{Err}_{i,n} \tag{1}$$

The purpose of averaging over a range of i values is to account for the fact that model accuracy may depend on the dates of when training data is collected. By starting the training data period on different dates we may achieve an error estimation that applies to a broad range of training start-dates.

This is plotted in Fig. 3. We see that for up to 40 days of training data, the MSE of a model decreases approximately linearly. Performing a linear regression allows us to estimate the MSE for a model trained on $n \in \{1, \ldots, 40\}$ days' colocation data as $\mathrm{Err}_n = (44.48 \pm 0.53) - (0.23 \pm 0.02) \times n$. After 35 days the MSE actually increases when more days' training data are added. This may be because only training windows with at least 35 days will include the anomalous period seen in Fig. 4.

Seasonal Dependencies of Accuracy. In addition to understanding model sensitivities to the amount of training data available, it is also useful to check for sensitivities to when in the season the training data is collected. The distribution of PM concentrations changes over the course of the winter, and different distributions of training data may yield better models than others. To explore this, we again tested the performance of linear regression models on the Single ODIN Dataset. We iterated through every noon-to-noon week-long and month-long window, used the data from this window as a training set and the rest of the data as a test set. The MSE of these models are illustrated in Fig. 4.

As before, we performed a simple linear regression over these error estimates, this time using the date on which the training data was collected as the regressor. This revealed positive trend between the MSE and how late in the season the training data is collected. Each day's delay in the collection of training data added an average of 0.35 ± 0.20 (95% CI) to the MSE of a model trained on one week's worth of data, and added an average of 0.35 ± 0.06 (95% CI) to the MSE of a model trained on one month's worth of data. In either case, there is a statistically significant positive relationship between how late in the season the training data is taken and the accuracy of the model ($p < 0.05$). By examining Fig. 4, we see that there is a period near the end of September when the error is anomalously high. This could have something to do with daylight saving time starting (this occurred on the 25th of September), when human and natural daily cycles deviate from one another. Whatever the cause, it is not at all clear that the positive trend between lateness of training data and accuracy of resulting models exists beyond the fact that when training data is taken later in the season, it is more likely to coincide with the anomalous period. The only practical policy to be gleaned from this result seems therefore to be to avoid collecting training data during the end of September. Whether better results are obtained by taking training data from early in Winter is unclear.

5 Model Selection

In this section we attempt to find a suitable model for predicting $\mathrm{PM}_{2.5,\mathrm{ECan}}$ from ODIN measurements.

Bias Variance Trade-Off. Choosing a model involves a trade-off between bias and variance. Bias, or approximation bias, is error resulting from poor choice of model, and variance, or estimation variance, is error resulting from the model being trained on some particular training set, rather than on the actual underlying distribution.

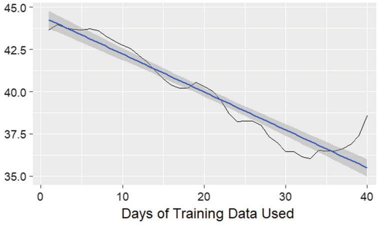


Fig. 3. Estimation of the MSE of a model dependent on the number of days training data used.

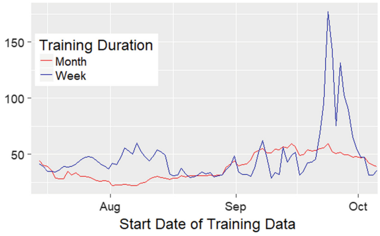


Fig. 4. Estimation of the absolute MSE of a model versus when in the season the training data is collected (x-axis).

There are two main classes of models available: parametric and nonparametric. For nonparametric models – including kernel regression, k-nearest neighbour, splines, and local polynomials – the MSE of the model follows

$$MSE_{nonpara} = \sigma^2 + O(n^{-4/(p+4)}) \tag{2}$$

where the σ^2 term is the intrinsic noise term, $O(n^{-4/(p+4)})$ is the estimation variance term, n is the size of the training set and p is the dimensionality of the feature vector [11]. Here $p = 3$, so the variance term becomes $O(n^{-4/7})$.

Compare this with the MSE for a (nonparametric) linear model:

$$MSE_{linear} = \sigma^2 + a_{linear} + O(n^{-1}). \tag{3}$$

The estimation variance term shrinks faster with more training data than in the nonparametric case. However, the additional term a_{linear} represents approximation bias resulting from the "true" relationship deviating from linearity.

Figure 5 shows that the relationship between $\mathrm{PM}_{2.5,\mathrm{Ecan}}$ and $\mathrm{PM}_{2.5,\mathrm{ODIN}}$ is approximately linear, although it is unlikely that it is exactly linear. Previous attempts to apply a linear model to Plantower PMS dust sensor measurements have been fairly successful [2], indicating that the approximation bias of a linear model will be small. On the other hand, Fig. 5 also reveals that the data is very

Table 1. Results of 2-fold cross-validation of models using data from initial collocation of several ODINs.

Serial	Linear MSE	Quadratic MSE	Cubic MSE	Reg. Tree MSE
102	52.14	56.47	59.9	98.21
103	39.59	40.37	42.93	82.59
105	100.23	131.95	134.80	129.66
107	74.30	77.51	79.93	94.29
108	52.99	60.93	68.17	98.06
109	50.71	48.45	43.99	74.58
111	64.70	72.98	80.04	86.23
113	88.07	89.81	85.50	108.09
114	71.71	63.21	75.95	81.59
115	38.31	39.25	40.79	78.37
Average	63.27	68.09	71.2	93.17
Std.	20.40	27.61	27.94	16.54

Table 2. Results of 3-fold cross validation of models using ODIN-109 data.

Model	MSE
Linear	35.31
Quadratic	37.80
Cubic	155.65
Reg. Tree	51.61

noisy, so estimation variance will probably be large. From these considerations, it seems very likely that models with fewer degrees of freedom will outperform more complicated models, even if these other models have the capacity to more closely capture the true relationship between the variables.

In particular, it seems very likely that parametric models will perform better than nonparametric ones (a_{linear} will be small compared to the difference between the $O(n^{-1})$ and $O(n^{-4/7})$ terms from Eqs. 2 and 3). Nonparametric models generally require fine tuning of hyperparameters (for example, kernel regression requires bandwidth selection for each dimension), so they take considerably more effort to implement than parametric models. We therefore tried several parametric models before trying any nonparametric ones, to determine whether increasing the degrees of freedom yields greater accuracy (Table 1).

From the above considerations, we chose to test additive polynomials of up to order three. That is, a linear, quadratic and cubic model. In addition, we also tested a regression tree model, to make sure there were no large gains which could be easily achieved from a nonparametric model, even if it is not well-suited to the modeling task.

Multi-ODIN Dataset. The standard method of error validation in situations such as these is k-fold cross validation. Because this data is a time sequence, data points close together in time will be correlated, and so we compose folds of contiguous blocks of data to avoid overfitting.

The results of performing a 2-fold cross validation on the Multi-ODIN Dataset with several models are given in Table 2, along with the inter-ODIN mean and sample standard deviation. We see that no model significantly outperforms linear regression, supporting the notion that estimation variance dominates over approximation bias, so out of parsimonious considerations we should use a linear model.

Single ODIN Dataset. The above experiment validated the use of a linear model across ODINs over a short time horizon. We next performed a 3-fold cross validation on the Single ODIN Dataset to verify that the model holds up over a medium time horizon. The results are given in Table 2. Once again, the linear model is not significantly outperformed. From these two experiments, we can have fairly high confidence that the linear model is the best choice, and especially that investing more effort into developing a more complex model will not pay off with a significant improvement in accuracy. Interestingly, consistent across all models, the mean error from using the third fold as the training data is significantly larger those obtained with the other two folds. Figure 4 sheds some light on this, by demonstrating that there is an anomalous region during the latter third of the experiment.

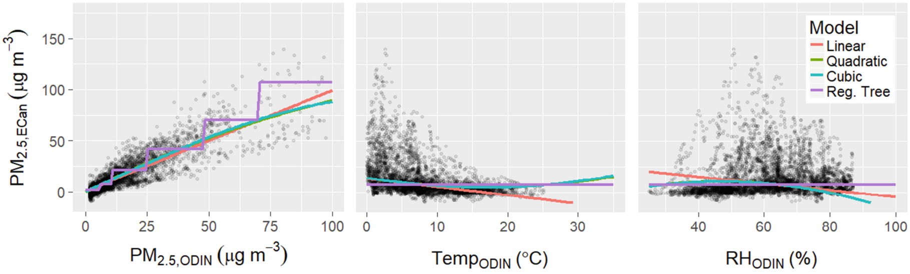


Fig. 5. On the x-axes are the measurements made by the ODIN, compared to the actual $PM_{2.5}$ values (as measured by the ECan TEOM-FDMS) on the y-axis. These data points come from the first third of the Single ODIN dataset, and the lines designate the results of different modelling techniques.

Robustness. To extend the above result, we would also like to test whether a robust linear regression model performs better than a simple linear regression model. Given that the noise is high enough for estimation variance to dominate over approximation bias, it seems likely that greater robustness against outliers would improve performance. We performed a 3-fold cross validation using an M estimator on both the Single and Multi-ODIN Datasets. For the Multi ODIN Dataset, the mean square error estimation dropped from 63.27 ± 20.40 to 59.06 ± 18.13 when switching from linear to robust linear (these are the inter-ODIN averages, with standard deviations) and in the Single ODIN Dataset the mean square error estimation dropped from 35.31 to 34.86. Although these reductions are not significant, adding robustness *a priori* seems sensible for noisy data such as these. We therefore believe that the best model in this context is produced by robust linear regression.

Feature Selection. Work on a previous iteration of an ODIN found that the dust sensor's performance varied with temperature and relative humidity conditions [10]. This dataset is small enough that we were able to apply a brute-force feature selection technique to verify that using a feature vector of $(\mathrm{PM}_{2.5,\mathrm{ODIN}}, \mathrm{Temp}_{\mathrm{ODIN}}, \mathrm{RH}_{2.5,\mathrm{ODIN}})$ did indeed yield a more accurate model than using $\mathrm{PM}_{2.5,\mathrm{ODIN}}$ as a sole regressors.

6 Concept Drift

In this section we attempt to determine whether the ODIN measurements are stable over time, or if sensor degradation is occurring. In the study on the earlier ODIN, it was discovered that significant baseline drift occurs over the course of several weeks, rendering a baseline correction necessary [10]. We do not know if such a correction is required in this case. If sensor degradation *is* occurring, then this would be concept drift, *i.e.*, a non-stationary learning environment.

Virtual and Real Concept Drift. It will be useful here to draw on the distinction sometimes made between real and virtual concept drift: real drift is a physical change in the actual learning environment, whereas virtual drift does not occur in reality, but only in the computer model [14]. This may be a result of a changing context altering the distribution of the data, and thus affecting the model error in such a way that requires the model to adapt. The distribution of $\mathrm{PM}_{2.5}$ is changing over time: earlier in the season, there are many more instances of high $\mathrm{PM}_{2.5}$ concentrations. This makes it seem likely that virtual drift will be occurring, so it will be difficult to differentiate this from real drift.

There are plenty of techniques for detecting concept drift, either real or virtual. For example, one can monitor the raw data stream [1], the error rate of the learner [5] or the parameters of the model itself [13]. However, most of these techniques are designed for classification, rather than regression problems. Furthermore, the problem of detecting real drift apart from, but in the presence of, virtual drift has not been previously explored in any depth.

Evolving models which adapt to concept drift - whether via adaptive ensembles [6], instance weighting [7] or adaptive sampling [4] may perform well under these conditions. However, because these techniques require the receipt of correct labels after they have made their predictions, they are useless for our purposes: once an ODIN is deployed, we will not have access to "correct" labels.

Detection of Real Drift. Fortunately, in this particular case we *do* have a way of detecting real drift in the presence of virtual drift. Many of the ODINs were at the ECan site at both the beginning and the end of the experiment.

For each of these ODINs we can therefore train models on both the initial and final ECan collocations. If the parameters change from the first case to the second, we cannot determine whether this is due to real or due to virtual drift. However, if we have data from two ODINs from the same time intervals, then because these data are collected from identical PM conditions any virtual drift which occurs should result in parameters moving in the same direction for both ODINs. Thus if the model parameters drift apart, then we have evidence of real drift (although we will not be able to tell which of the ODINs are degrading, nor in which direction).

Table 3. The absolute changes of each of the model parameters between being trained on the initial and final collocations with the ECan site. "Cnst." denotes the constant term of the model, and "ΔCoef." denotes the change in coefficient for that measurement between in initial and final collocation. The averages include the standard deviation across ODINs.

Serial	ΔCnst.	$\text{PM}_{2.5,\text{ODIN}}$	$\text{Temp}_{\text{ODIN}}$	RH_{ODIN}
		ΔCoef.	ΔCoef.	ΔCoef.
102	−6.53	−0.01	0.44	0.05
105	0.59	−0.23	0.21	−0.01
107	−6.39	0.29	0.53	0.05
108	−7.35	−0.17	0.33	0.08
109	−1.54	−0.21	0.34	−0.01
113	−14.25	0.15	0.45	0.15
115	−8.21	−0.20	0.31	0.10
Average	-6.24 ± 4.80	-0.06 ± 0.21	0.37 ± 0.11	0.06 ± 0.06

To implement this idea, we used data from ODINs 102, 105, 107, 108, 109, 113 and 115, which have a reasonably large intersection of colocation times. We then performed linear regressions over the initial and final colocation datasets and compared the parameters for the initial and final colocation datasets. Random noise could still cause some inter-ODIN differences in model changes, and to mitigate this effect, we again used `rlm` for robustness against outliers. The changes in the $\text{PM}_{2.5,\text{ODIN}}$ coefficients are illustrated in Fig. 6 (including 95% confidence intervals, estimated as twice the standard deviation given in the `rlm` output), and the changes in all the model parameters are given in Table 3.

We see that the ODIN model parameters are indeed drifting apart from one another, providing good evidence that real drift (*i.e.*, sensor degradation) is occurring. This is especially evident in the case of $\text{PM}_{2.5,\text{ODIN}}$, which is the feature we are most concerned with. The $\text{PM}_{2.5,\text{ODIN}}$ coefficients of ODIN-107, ODIN-108, ODIN-109 and ODIN-115 all decrease from the initial to the final colocation by a fairly consistent amount, whereas that of ODIN-105 *increases* by a similar amount and those of ODIN-102 and ODIN-113 do not vary much.

Because the $\mathrm{PM}_{2.5,\mathrm{ODIN}}$ coefficients which do change, change by several confidence intervals, we can be fairly certain that different ODINs are genuinely degrading, and thus that real drift is in fact occurring.

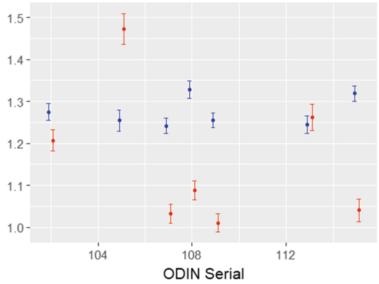


Fig. 6. $\mathrm{PM}_{2.5,\mathrm{ODIN}}$ Coefficient of robust linear regression model built using initial (blue) and final (red) collocation data. (Color figure online)

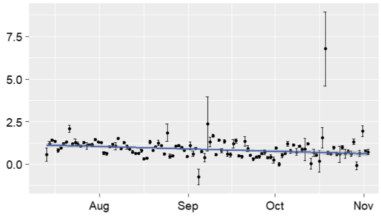


Fig. 7. The $\mathrm{PM}_{2.5,\mathrm{ODIN}}$ coefficients of linear regression models trained on successive noon-to-noon 24 h periods of ODIN-109 data.

Model Evolution Over Time. Having established that some kind of sensor degradation (real drift) is occurring, we now look at the entire ODIN-109 dataset, to try and discern trends in how the model changes over time. By dividing the dataset into 24 h period (noon to noon) blocks and training a robust linear model on each block, the movements of each model parameter can be plotted over time, as in Fig. 7. We notice that the $\mathrm{PM}_{2.5,\mathrm{ODIN}}$ coefficient is steadily decreasing over time. However, we do not know if this represents real or virtual drift. We can attempt to differentiate the two types of drift by making a correction for the PM conditions of each day.

This can be achieved by performing a simple linear regression over the mean $\mathrm{PM}_{2.5,\mathrm{ECan}}$ value and the date of each 24-hour period, using the $\mathrm{PM}_{2.5,\mathrm{ODIN}}$ coefficient as the dependent variable. The resulting time-dependent slope indicated that the $\mathrm{PM}_{2.5,\mathrm{ECan}}$ coefficient on average decreases by $(1.08 \pm 5.46) \times 10^{-3}$ (95% CI) per day, regardless of actual PM conditions for that day. This indicates that real drift is occurring in ODIN-109, and that the sensor is degrading in a steady, linear manner over time. Further research is required to characterise this drift more precisely so that drift corrections can be incorporated into the model.

7 Conclusion

We examined the question of how best to extract useful information about PM concentrations from an ODIN instrument. We produced a formula relating the amount of training data collected (*i.e.*, the length of the colocation with the

ECan site) to the error of the resulting model. This allows more informed decisions about trade offs between the quantity and the quality of the data which can be collected. Whether model accuracy has a dependency on when in the season training data is collected was ambiguous. Because the measurements taken by ODINs are so noisy, estimation variance dominates over approximation bias on the time scales we are interested in. This means that linear regression models yield optimal performance. From there, removing outliers with M estimators likely further improves performance. In particular, feature vectors of $(\mathrm{PM}_{2.5,\mathrm{ODIN}}, \mathrm{Temp}_{\mathrm{ODIN}}, \mathrm{RH}_{2.5,\mathrm{ODIN}})$ seem to work best for linear regression models. We also established that some form of sensor degradation was occurring over the course of the experiment, and attempted to quantify this. However, more work will need to be done on this before useful corrections can be made.

References

1. Bifet, A., Gavalda, R.: Learning from time-changing data with adaptive windowing. In: Proceedings of the 2007 SIAM International Conference on Data Mining, pp. 443–448. SIAM (2007)
2. Air Quality Sensor Performance Evaluation Center. Purpleair PA-ii - summary report. http://www.aqmd.gov/docs/default-source/aq-spec/summary/purpleair-pa-ii---summary-report.pdf?sfvrsn=4. Accessed 20 Feb 2018
3. Cohen, A.J., et al.: The global burden of disease due to outdoor air pollution. J. Toxicol. Environ. Health Part A **68**(13–14), 1301–1307 (2005)
4. Delany, S.J., Cunningham, P., Tsymbal, A., Coyle, L.: A case-based technique for tracking concept drift in spam filtering. Knowl. Based Syst. **18**(4–5), 187–195 (2005)
5. Gama, J., Medas, P., Castillo, G., Rodrigues, P.: Learning with drift detection. In: Bazzan, A.L.C., Labidi, S. (eds.) SBIA 2004. LNCS (LNAI), vol. 3171, pp. 286–295. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28645-5_29
6. Zico Kolter, J., Maloof, M.A.: Dynamic weighted majority: an ensemble method for drifting concepts. J. Mach. Learn. Res. **8**, 2755–2790 (2007)
7. Koychev, I.: Gradual forgetting for adaptation to concept drift. In: Proceedings of ECAI 2000 Workshop on Current Issues in Spatio-Temporal Reasoning (2000)
8. Lu, X., Wang, Y., Huang, L., Yang, W., Shen, Y.: Temporal-spatial aggregated urban air quality inference with heterogeneous big data. In: Yang, Q., Yu, W., Challal, Y. (eds.) WASA 2016. LNCS, vol. 9798, pp. 414–426. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42836-9_37
9. McKone, T.E., Barry Ryan, P., Özkaynak, H.: Exposure information in environmental health research: current opportunities and future directions for particulate matter, ozone, and toxic air pollutants. J. Expo. Sci. Environ. Epidemiol. **19**(1), 30 (2009)
10. Olivares, G., Edwards, S.: The outdoor dust information node (ODIN)-development and performance assessment of a low cost ambient dust sensor. Atmos. Meas. Tech. Discuss. **8**, 7511–7533 (2015)

11. Shalizi, C.: Advanced Data Analysis from an Elementary Point of View. Cambridge University Press, Cambridge (2013)
12. Snyder, E.G., et al.: The changing paradigm of air pollution monitoring (2013)
13. Su, B., Shen, Y.-D., Xu, W.: Modeling concept drift from the perspective of classifiers. In: 2008 IEEE Conference on Cybernetics and Intelligent Systems, pp. 1055–1060. IEEE (2008)
14. Widmer, G., Kubat, M.: Effective learning in dynamic environments by explicit context tracking. In: Brazdil, P.B. (ed.) ECML 1993. LNCS, vol. 667, pp. 227–243. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-56602-3_139
15. Zheng, Y., et al.: Forecasting fine-grained air quality based on big data. In: Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 2267–2276. ACM (2015)

An Approach to Compress and Represents Time Series Data and Its Application in Electric Power Utilities

Chee Keong Wee(✉) and Richi Nayak

Science and Engineering Faculty, Queensland University of Technology,
Brisbane, Australia
{Chee.wee, r.nayak}@qut.edu.au

Abstract. This paper proposes a novel method that can reduce the volume of time series data adaptively and provide an alternate means in handling time series symbolically which can be used for time series' classification or anomaly detection. The proposed method is tested using the time series data obtained from utility companies' substations by comparing the compressed outputs to the original forms. The result is a new discretized set that is lower in volume and can represent the time series succinctly with a minimal loss that can be managed.

Keywords: Time series compression · Time series representation

1 Introduction

Time series data is ubiquitous amongst several domains of industries. For example, in the domain of electricity utility, time series is prevalent across the utility assets such as substations, transformers, metering and control systems and a high volume of data is generated routinely [3]. The time series values are related to the energy outputs such as electrical current, voltage, frequency as well as the values derived from various electricity control systems [2]. The amount of data produced by modern equipment is a constant challenge for analysts to maintain, store and analyze. The frequencies of time series are usually produced in very fine-grained intervals such as in minutes or seconds or even in milliseconds [1] but there is few requirements for such fine granularity.

While the new bespoke IT solutions and technology such as big data and time series databases are available that can handle the volume and velocity of time-series data, most companies including the modern electricity utility rely on relational database management systems for their enterprise business data storage [4]. Storing them in databases at their existing state of granularity is not practical. There is no benefit in storing trillions of rows in databases that cannot be queried efficiently without massive parallelized hardware. While algorithms such as Piecewise Aggregate Approximation (PAA) [5] or Adaptive Piecewise Aggregate Approximation (APAA) [6] can effectively compress or reduce the volume of data, there is a need to control the level of compression on the time series and is simultaneously capable of using the compressed forms for other purposes like comparison and grouping processes which tend to take more time to process if the raw time series are used. The PAA and APAA algorithms

R. Islam et al. (Eds.): AusDM 2018, CCIS 996, pp. 107–120, 2019.
https://doi.org/10.1007/978-981-13-6661-1_9

produce results that are specific to that domain. However, a challenge is faced to use the time series in a reduced form for other purposes such as comparison and classification. Usually, these compression techniques are topical and requires additional processes to revert to the original time series interval. A more generalized approach is required which can be applied to a wide range of time series classes, each class with different magnitude in values and intervals.

In this paper, we propose an alternative approach that can be applied to a wide range of time series' classes and compress them using a parameter to throttle the depth of the compression and the maximum size of the dynamic window. The compressed form is represented symbolically, using a series of paired value (the window size and the mean value), in a format that corresponds to a reference matrix. The proposed method of compression is applied to a large set of time series that had been acquired from an Australian's electricity utility company. The results are examined to validate the amount of loosely compression that can be achieved versus the loss in information granularity. The feasibility of using the symbolical representation of compressed data for general function such as comparison is also tested.

2 Related Work

Dimensionality reduction is a popular data compression approach to reduce the complexity and volume of the high-frequency time series data. It is a form of lossy compression that represents time series by the shape characteristics while managing the noise level. A time series, x, of length, n, is compressed with a dimensionality of p where p < < n such that it closely approximates to x [1]. There are two common known approaches of time series dimensionality reduction [2]. The non-adaptive methods such as Piecewise aggregate approximation (PAA) [3], Discrete Fourier Transform (DFT) [4] and Discrete Wavelet transform (DTW) [4] and Symbolic Aggregate Approximation (SAX) [5] compress the time series with parameters that are constant through the entire time series, ignoring their trend and shape. The adaptive methods such as Singular Value Decomposition (SVD) [6] and Adaptive PAA (APAA) adapt to the shape of the time series, making the parameters to compress varied.

A non-adaptive compression method represents time series in a reduced format by aggregating it into small segments. For a time series X of length n, the PAA changes it into a smaller vector of the desired period m, transforming it into $\bar{X} = (\bar{x}_1, \bar{x}_2, \ldots\ldots, \bar{x}_m)$, where m $\leq$ n. The reduced time series $\bar{X}$. is calculated as in Eq. (1) [7];

$$\bar{X}_i = \frac{m}{n} \sum\nolimits_{j=\frac{n}{m(i-1)}+1}^{\frac{n}{m}i} x_j \tag{1}$$

An adaptive method generates segments as per activities within the data [8]. A segment will be longer if that area has low activity and vice versa for those with high activity [3, 7]. The reduced vector is presented as $\bar{X} = (\{\bar{x}_1, r_1\}, \{\bar{x}_2, r_2\}\ldots\ldots, \{\bar{x}_m, r_m\})$. where, $x_i = mean(x_{r_{i-1}+1} \ldots x_{ri})$, r is the length of the segment, i is the sequence of the segment, and m is the desired vector size. SAX [5] is a two-step process that uses PAA to reduce the raw times series of length, n, into segments of size, w, each with its own mean values. This is followed by discretizing it with a series of alphabets of length, k, where k < n. SAX assumes that the PAA represented results follow the Gaussian distribution, so each segment is tagged to an alphabet starting with A for the lowest band of values to the top of F if there are six bins in the distribution [5]. The result of a SAX's represented time series is a list of alphabets and while this can be used for clustering tasks, there are two shortcomings. Firstly, it uses the non-adaptive PAA method that is less sensitive to take the morphological characteristics of the time series into consideration. Secondly, the use of alphabets makes the calculation of Euclidean distances between the SAX represented time series' data points more complicated for comparison such as clustering and classification.

The APAA-KShape method [9] combines APAA together with the k-shape algorithm to cluster the compressed results. K-shape is a partitional clustering algorithm that first allocates centroids across a group of compressed times series that had been processed by APAA and then finds the distance between the represented point to the centroid via shape-based distance [10]. The distance measurement is adjustable on its scale, shift and noise [10]. However, this approach has the following shortcomings: firstly, the algorithm requires the participating groups of represented time series to have similar start time and values' magnitude.

In general, the time series produced by electricity industry has greater variation values and the accuracy of the comparison among the represented time series will be affected. In this paper, we propose a method that does not need the load profile's shape to be on the same temporal plane. The participating time series can start from any time, and there is no exclusion or boundary set on the time series' data point values or their temporal reference. It should be magnitude neutral and less sensitive to the frequency of their intervals. A time series of <10 KW with 5 min interval can be compared with another time series that has >100 KW and 20 min interval with minimum difficulty. The degree of similarity among the represented time series is percentage score.

Some sophisticated methods that apply function approximation, such as regression, to compress the data and use the function for comparison and clustering purposes [11–13]. However, the application is topical and confine to a specific group of time series with boundary on the value and intervals. These methods are attuned to retain the morphological characters of the load profile in the time series and the new representation has given significant reduction in term of information size. But it has not been addressed on how these adaptive methods of compression can be utilized for other tasks such as clustering and similarity comparison.

3 The Proposed Time Series Data Compression Method

The proposed method, Piecewise Dynamic Aggregate Approximation with Matrix Symbolic Representation (PDAAMSR), has been designed to meet three requirements. Firstly, it must be adaptive, and the degree of compression can be controlled. Secondly, the discretization of the compressed format can be easily used for clustering and comparison tasks. Thirdly, it eliminates the needs for the time series to have the same start points or similar temporal to enable the clustering process. PDAAMSR has been developed to perform dimensionality reduction as a form of compression that is general purpose enough to be used to compress a large volume and variety of time series data of different range, intervals, frequency, and magnitude. The PDAAMSR aims to achieve higher dimensionality reduction with parameters that can be adjusted to control the level of difference in the reduction process; the higher the difference value, the greater the compression will be. However, it increases the loss value too. The compressed format of the time series is represented symbolically using the proposed matrix coordinates. An example is shown in Table 1. The proposed method includes three stages: normalize the data, apply the reduction algorithm, and replace the compressed time series with reference to a matrix that will be built from the compression results.

3.1 Normalization of Time Series

The time series data is normalized to the range of 0 to 1, and their identification as well as attributes are recorded in a reference repository. The time series' attributes are any details such as reading types, intervals, spatial-temporal information as well as its maximum and minimum values range. The normalization enables the time series data to present in a common standard format and to manipulate across the multitudes of various time series data. In addition, dealing with a big number of time series data from different sources and entities is categorically challenging, the preparation can set them on an equal base that is easier for the mining algorithm to process.

3.2 Aggregate Approximation Based on Difference and Window Sizes

The next step is to compress the normalized time series into a summarized version that is comprised of a series of paired representation; window's size and mean value. Two parameters are introduced to manage the compression capabilities: the differencing factor - *d*, and the window size - *l*, limit. The differencing factor - *d*, defines the allowable difference between the mean value - a_i, in the window - w_i, and the next data point that is adjacent to the window's boundary. The window size limit - *l*, serves two purposes; the first is to prevent over-representation of a time series that has extensive constant readings, and the second is to set a boundary so that the index matrix's dimension can be limit to a manageable size. Algorithm 1 detailed the logic of looping through each time series, comparing each subsequent data point and decide whether to assimilate them into existing window or spurn a new one.

Algorithm 1. – representing the time series against pre-determined window and differencing limit

```
Input: $\bar{Y}$=normalized time series, $l$=windows max limit, $d$= differencing factor,
        $h$=substations_total
Output: $Y`$=represented time series
For $g$ =1:$h$
  Initialize ($Y`_g$)
  For $b$ = 1: length($\bar{Y}$)
    if $b$ = 1
         initialize($w_s$,$a_s$):
    else
    if ($\bar{Y}_i$ - $a_s$) < $d$ and $w_s$ < $l$:
         $w_s$ +=1
          $a_s$ = mean($a_{s-1}$ + $\bar{Y}_i$)
    else
       $Y`_g$ : append($w_s$,$a_s$)
       initialize($w_s$,$a_s$)
     End For
 End For
```

3.3 Matrix Representation of Compressed Time Series

The use of both time intervals and point values derived from the compression is not practical due to their large variety and ranges. Our idea is to transpose the asset information in a matrix and lays the baseline in which the comparison of assets can be performed easily with item counts instead of the usual point-distance calculation like dynamic time warping (DTW) [1]. We propose to use a unique type of index matrix. This matrix has dual axis for both X and Y to facilitate the discretization, of the complex represented paired values that comprise of time intervals windows and data values, with simple matrix coordinates. The elements in the matrix serves as a storage list to keep track of the time series' ID should our algorithm choose that elements for discretizing as shown in Table 1. The discretized output is a list of paired values that is related to the index matrix as shown in Table 2. Once the whole batch of substations' time series have been compressed and represented, the next step is to build this index matrix. All the represented time series' values are split into two groups, intervals and values, in accordance to the axis and sorted. Duplicates are removed so that the group has a list of distinct entries. They are then arranged to form two axes in this index matrix; the X-axis for the occurrence of all the distinct intervals versus the Y-axis of all the distinct values as shown in Table 1 below mapped to the bottom and right. The matrix coordinates that will be used for the discretization process is displayed on the left and top of the matrix and each of them is mapped to a distinct entity of the consolidated represented values. Algorithm 2 depicts the process in index matrix's construction using the sorted distinct represented values, discretization and population of the matrix's elements concurrently.

Referring to Table 1, those elements in the matrix has a list of the substation names which shared the element's coordinates for the discretization. We suggest that for any two given substations, their similarity to one another depend on the number of the index matrix's elements that they shared in the discretization process; the higher the count, the greater the similarity these two substations share. This forms the basis for the comparison effort for represented time series with heterogenous window time scale. A two-loop simple query can acquire the similarity degree for all the substation in a many-to-many relationship. For instance, based on the matrix in Table 2, the example of element (2, 2) represented (120 s, 0.02). This element includes two substations, Amamoor and Algester, that shared same coordinates in the discretization of their represented paired-values. Likewise, element (97, 98) represented (480 s, 0.98) with two substations, Arundel and Arana Hills, shared same coordinates.

Algorithm 2. – discretize time series and populate index matrix

```
Input: $Y`$=represented time series, $h$=substations_total
Output: $Y``$=discretized time series, $M$=index matrix
$e$ : list(sort{distinct{$Y`[w]$}})
$i$ = count($e$)
$f$: list(sort{distinct{$Y`[a]$}})
$j$ = count($f$)
$M$ = initialize_matrix($i,j$)
For $g$ =1:$h$
   for $d$ = 1:length($Y`_g$)
        $w`_d$ = discretize {$Y`_g[w]_d$, $e$:$i$}
         $a`_d$ = discretize {$Y`_g[a]_d$, $f$:$j$}
        $Y``$: append(list[$w`_d$, $a`_d$])
        $M_{ij}$ : register_substation($g$)
```

Table 1. The proposed APAAMSR's index matrix

Matrix (i, j)	1	2	3	4	…	…	97	98	99	100	Norm value	Substations
1		AB, AC, AM, AB	AB, AR, AG	AR, AG, AR	AH, AF, AB		AC, AN	AC, AH, AF,AR		AM, AF, AR	**0.01**	AR - Acacia Ridge
2	AC, AB, AN	AM, AG	AB, AS	AB, AS, AG		AC, AG, AN	AB, AM	AC, AR	AN, AC, AH, AG	AB, AM, AS	**0.02**	AC - Albany Creek
3		AC, AM, AS	AS, AB, AH		AB, AN, AG	AC, AH, AH, AS	AB, AH	AG	AH, AR	AM, AS	**0.03**	AH - Alexandra Headland
4	AH, AR, AF		AC, AG, AB	AC, AG	AF, AH	AC, AB, AN	AM	AB, AM	AM, AR, AH		**0.04**	AG - Algester
..	AM, AN	AB, AB	AC, AR, AH, AB	AB, AH	AH		AM, AR	A, AR, AB	AB, AM	AB, AM, AF	.	AM - Amamoor
96	AB, AG	AC, AG	AM, AS	AM, AF	AS, AN, AH		AR	AF, AR, AG	AH, AR, AF	AB, AR	**0.96**	AS - Ann St
97	AC, AB	AB, AM, AB	AC, AB, AS	AB, AN, AG	AG	AB, AS	AC, AM, AG, AF	AF, AM, AR	AC, AH	AR, AF, AG	**0.97**	AN - Annerley
98	AC, AB, AM, AB			AM, AF	AS, AN, AR		AR, AH	AB, AC, AG	AN, AG	AB, AS	**0.98**	AH - Arana Hills
99		AB, AG	AE, AF, AR, AG	AN, AG	AF, AB, AR	AB, AC, AH	AC, AB, AN		AB, AF, AR		**0.99**	AF - Archerfield
100					AN, AB, AR, AG			AF, AG, AS		AM, AB, AN	**1.00**	AR - Arundel
Interval	60 s	120 s	180 s	240 s	…	…	420 s	480 s	540 s	600 s		AG - Ashgrove

4 Experimental Results

The purpose of the experiments is to determine the effectiveness of the algorithm in compressing time series and presenting the results using matrix references. The effectiveness of the compression is validated by comparing the amount of information loss to the difference value set for the compression. We assess the feasibility of the index matrix use in consolidating the discretized compressed values by measuring the degree of similarity. The result shows the relationship between the amounts of information loss versus the level of compression specified, as well as using the information stored in the matrix for the asset's comparison. The datasets used for the tests comprised of two sets. The first set is comprised of 528 substations' time series that belong to an Australian electricity utility company. The data has granularity of 15 min interval, 17528 data points per year and each substation has 10 years' worth of historical data. The second set of times are non-power utility related that cover stock prices and air passengers travel.

4.1 Information Loss vs Compression Level Results

We tested the PDAAMSR method with PAA and APAA on the substation plus non-electricity related time series. For the PDAAMSR, we use a variety of differencing limits and the MAE scores are shown in Table 2. The amount of information loss is related to the level of compression. Based on the result, PAA has the highest amount of information loss where APAA has significantly less loss as it uses a differencing limit which we set at 10% but there is no limit on the window size, so there is a chance of cumulative aggregation without any limit. For the PDAAMSR, there is a finite size that each window size can reach, which mitigate the potential runaway cumulative aggregation. A lesser differencing limit of 5% incurs lesser information loss.

Figure 1 shows the result when PDAAMSR is applied to all the substations using a series of difference limit's values at an increment of 5 for a range from 0 to 20%. It can be observed that the inverse correlation exists between the differencing limit setting and the average percentage of information loss that was encountered. The loss started off slow initially, but it starts to degrade fast as the limit value increases. The two parameters show a balanced trade-off and a compression value of 10% is the optimum value.

Table 2. Comparison of PAA, APAA and PDAAMSR's MAE scores on different timeseries

Datasets	PAA	APAA	PDAAMSR (diff = 10%)	PDAAMSR (diff = 5%)
QLD substations	0.1188	0.0674	0.0613	0.0564
Air passengers	0.0826	0.0094	0.0669	0.0647
Stock prices	0.0872	0.0100	0.0557	0.0442

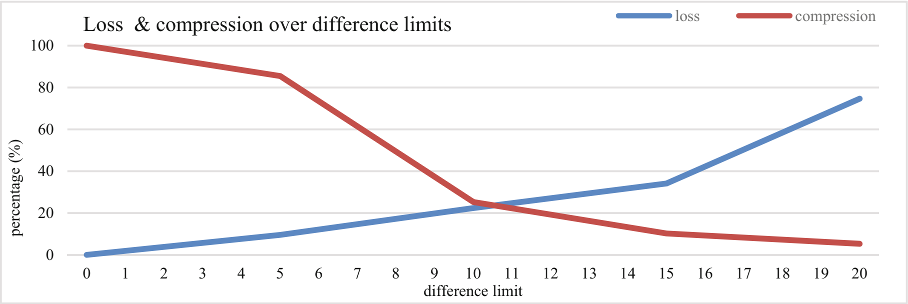


Fig. 1. Comparison of loss and compression versus difference limits results using PDAAMSR on substations' time series

Next, we present the compression result with the PDAAMSR algorithm in Fig. 2. A difference value of 10% is used and the maximum window size is 20. The PDAAMSR compressed result is tuned to the time series' shapes, aggregating to the load profile with a small degree of compromise to the details. The compressed result, a series of represented pair-values (the window size and the mean normalized value), is represented as matrix coordinates that correspond to the index matrix. The time series has 336 data points in its raw form and there are 29 paired values after compression. There is a reduction of 90% in

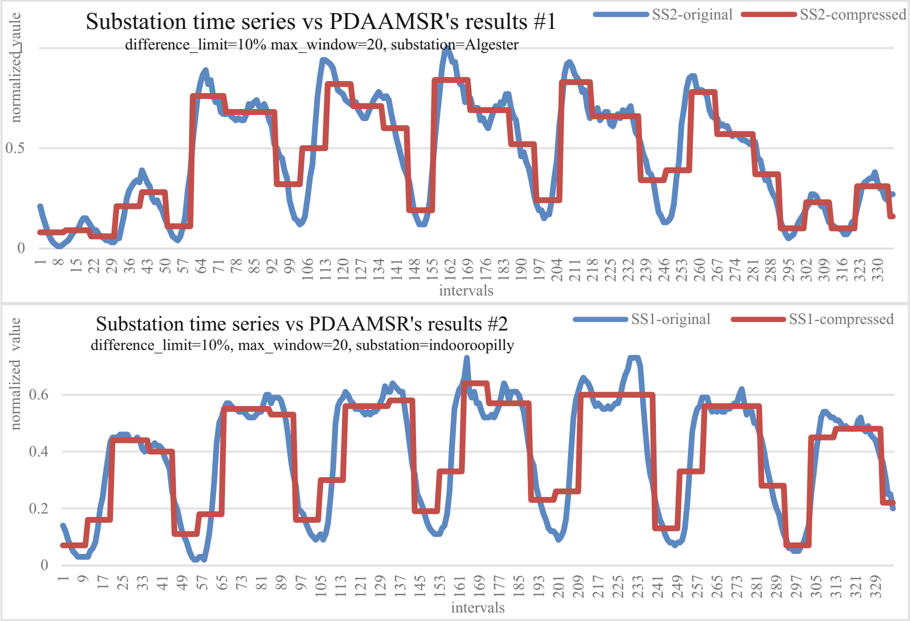


Fig. 2. Comparison of 2 substations' time series vs PDAAMSR represented results

term of data points in the compressed format. It is evident that the aggregation results given by PAA are less attuned to the trend in the original time series. This forces those extreme data points within the segment windows to conform to the average values, thus losing important features information. For the result that was compressed by PDAAMSR, they are more sensitive to the trend and adhere to the contour of the raw time series, thus preserving the features.

We run another set of tests by applying the method to non-power utility related time series - air passengers travel, and stock prices movement as shown in Fig. 3. The test involved using two difference parameter settings; 5% and 10%. As expected, the difference with 5% setting has less information loss and it can express the time series' shape in greater details particularly for the stock pricings. It can be observed that time series with higher number of sharp fluctuation tend to lose more information as compared to those with more stable trends.

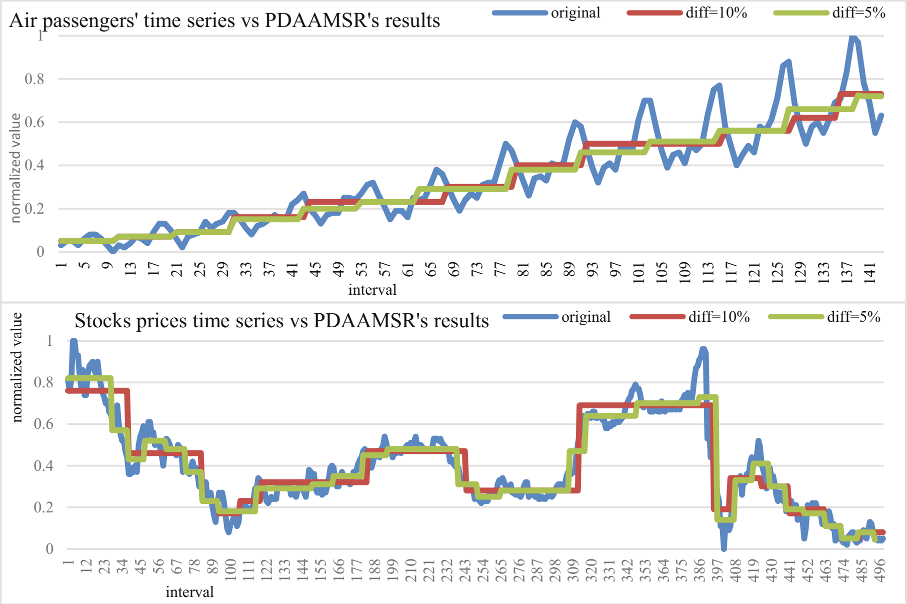


Fig. 3. Comparison of Stocks and air passenger time series vs PDAAMSR's represented results

In comparison to our method, we process the time series using PAA [14] and the result is shown in Fig. 4. The results showed that the compressed data aggregated at each fixed interval with significant information loss especially for data points with extreme values within the window. The aggregating values are confined within the window segments and don't follow the load profile's morphological characteristics close enough, ignoring off important details which are not desirable in electricity load analysis [15]. This shortcoming can be minimized by reducing the window segment size to increase the granularity, but the level of compression will have to be compromised.

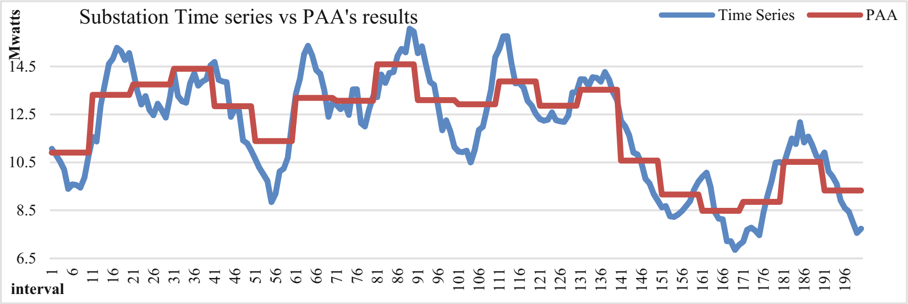


Fig. 4. Comparison of substation's times series vs compressed results from PAA

4.2 Matrix Representation Results

Table 3 showed the outcomes of applying our algorithm to the substations' time series. Using the parameter of difference at 10% and the max window size at 20, the compressed results are then represented to matrix coordinates. The amount of information loss by the time series when it is subjected to PDAAMSR is measured by reversing the converted values into a time series with a similar time interval and finds its difference against the original time series. The loss is estimated to be in the range of 8 to 10% using the combination of 10% difference and window limit of 20 but the amount of loss can be controlled by adjusting the parameter.

Table 3. Results of Substations' time series processed by PDAAMSR

Time series	Action	Results
Ss1(diff = 10%, Window = 20)	Compressed	(10 0.07), (10 0.09), (10 0.06), (10 0.21), (10 0.27), (10 0.1), (13 0.76), (20 0.67), (10 0.31), (10 0.49), (10 0.82), (12 0.7), (10 0.6), (10 0.18), (14 0.84), (16 0.68), (10 0.51), (10 0.23), (12 0.83), (19 0.66), (10 0.34), (10 0.39), (10 0.77), (15 0.57), (10 0.37), (10 0.09), (10 0.22), (10 0.09), (13 0.3)
	Represented	[10][7], [10][9], [10][6], [10][21], [10][27], [10][1], [13][76], [20][67], [10][31], [10][49], [10][82], [12][7], [10][6], [10][18], [14][84], [16][68], [10][51], [10][23], [12][83], [19][66], [10][34], [10][39], [10][77], [15][57], [10][37], [10][9], [10][22], [10][9], [13][3]
	MAE	0.089387

(*continued*)

Table 3. (*continued*)

Time series	Action	Results
Ss2(diff = 10%, Window = 20)	Compressed	(10 0.1), (10 0.17), (14 0.18), (13 0.48), (10 0.12), (18 0.3), (10 0.36), (10 0.74), (10 0.2), (16 0.36), (10 0.28), (12 0.78), (10 0.22), (14 0.39), (11 0.25), (15 0.75), (12 0.2), (10 0.47), (11 0.24), (15 0.64), (12 0.15), (10 0.39), (12 0.28), (14 0.61), (10 0.1), (24 0.24)
	Represented	[10][1], [10][17], [14][18], [13][48], [10][12], [18][3], [10][36], [10][74], [10][2], [16][36], [10][28], [12][78], [10][22], [14][39], [11][25], [15][75], [12][2], [10][47], [11][24], [15][64], [12][15], [10][39], [12][28], [14][61], [10][1], [24][24]
	MAE	0.095666

4.3 Computing Similarities Results

The next test is to assess the capability of our method in measuring the similarity among the substations' represented time series. A total of 91 substations time series were used for this but due to a large amount of information, only a subsection is showed in this report. The discretization process performed by the PDAAMSR populated the index matrix with the substation ID based on the corresponding coordinates that are used for the represented data points like in Table 1. Once PDAAMSR completes its process, the index matrix's individual elements would have a list of those substations that use the element's coordinates to discretize their represented paired-values. Using a two-dimensional loop, we can query against the index matrix to find the number of elements that are shared between the substations and thus, deduce the degree of their similarity. The approach here is to measure the occurrence of the substations' similar use of the index matrix's element and their sequence of occurrence are not taken into consideration. As the relationship matrix showed, a high value equates to a higher level of homogeneity between the substations' time series. The algorithm of comparison loops through the reference matrix and sum up the number of similarly represented bits. The chart in Fig. 5 plots the number of substations versus the number of similarly represented bits in substation's time series. The chart showed that there are many substations that have very few similarities with one another. Only 7 substations have similarities that are greater than 45% as shown in Table 5. Table 4 showed the correlation among the substations' discretized time series queried from the index matrix.

Table 4. Index matrix of the substation's discretized time series processed by PDAAMSR

	acaciaridge	albanycreek	alexandraheadlnd	algester	amamoor	amberley	annerley	annst	aranahills	archerfield	arundel	ashgrove	astorterrace	baldhillsbus1	baldhillsbus2	beaudesert	beenleigh	beenleighnorth	beerwah	belmont	bethania	birkdale	blackmountain	boonah	booval	brendale	Bribieisland	Brighton	broadbeach
acaciaridge		5	1	1	1		2		2	3	1	1		1	3	5	1	2	2	3	2	2	4	3	3	1	4	4	1
albanycreek	5		3	3	5	2	5	2	2	2	5	4		3	2	3	4	1	5		5	12	11	7	4	2	12	10	
alexandraheadland	1	3		3	1		1	3	4	2	1	3	3		2			2	5	1	2	5	6	3	1	2	10	4	
algester	1	3	3			1	8	3	1		2	4	3	4	2	2	3	1	5	5	4	2	3	5	6	5	5	2	2
amamoor	1	5	1					3	2	1	1	1		1	2	1			3			4	2	1	1	1	3	3	
amberley		2		1			2	2		1		1	1				1					1		1			1	1	1
annerley	2	5	1	8		2		4	3			6		5	3	3	5	1	2	3	1	2	3	3	2	4	2	3	1
annst		2	3	3	3	2	4		1	3		3	10	1	3	1	2	1	1		2	3		3		2	1	2	
aranahills	2	2	4	1	2		3	1			1	6	1	4	7	3	2	1	3	3	2	4	8	2	2		6	7	
archerfield	3	2	2		1	1		3				1	3	1	1	1		2	1		2	3	2	4	1	1	3	2	
arundel	1	5	1	2	1				1			2	1	1	1		2		3	2	1	4	4	1	4	2	5	4	
ashgrove	1	4	3	4	1	1	6	3	6	1	2		2	4	3	1	2	1	5	4	2	7	5	1	6	3	3	5	1
astorterrace			3	3		1		10	1	3	1	2		1	2	2	1	3			3		2	2	3	1	1	1	
baldhillsbus1	1	3		4	1		5	1	4	1	1	4	1		8	4	6	1	2	3	5	4	2	4	3	4	1	1	1
baldhillsbus2	3	2	2	2	2		3	3	7	1	1	3	2	8		4	2	1	4	2	1	1	6	1		3	6	4	
beaudesert	5	3		2	1		3	1	3	1		1	2	4	4		4	2	2	4	6	1	5	4	2	1	4	4	
beenleigh	1	4		3		1	5	2	2		2	2	1	6	2	4		1			4	1	2	3		3	2	3	
beenleighnorth	2	1	2	1			1	1	1	2		1	3	1	1	2	1				2		3	2	1	1		1	
beerwah	2	5	5	5	3		2	1	3	1	3	5		2	4	2				5	3	8	6	3	6	1	11	7	1
belmont	3		1	5			3		3		2	4		3	2	4			5			1			6	4	2	2	2
bethania	2	5	2	4			1	2	2	2	1	2	3	5	1	6	4	2	3			8	7	8	2	2	6	5	
birkdale	2	12	5	2	4	1	2	3	4	3	4	7		4	1	1	1		8	1	8		9	6	8	2	14	16	
blackmountain	4	11	6	3	2		3		8	2	4	5	2	2	6	5	2	3	6		7	9		4	5	1	14	12	
boonah	3	7	3	5	1	1	3	3	2	4	1	1	2	4	1	4	3	2	3		8	6	4		3	4	7	5	
booval	3	4	1	6	1		2		2	1	4	6	3	3		2		1	6	6	2	8	5	3		5	6	8	2
brendale	1	2	2	5	1		4	2		1	2	3	1	4	3	1	3	1	1	4	2	2	1	4	5		2		1
bribieisland	4	12	10	5	3	1	2	1	6	3	5	3	1	1	6	4	2		11	2	6	14	14	7	6	2		14	
brighton	4	10	4	2	3	1	3	2	7	2	4	5	1	1	4	4	3	1	7	2	5	16	12	5	8		14		
broadbeach	1			2		1	1					1		1					1	2					2	1			

Table 5. List of substations with > 45% similarities using PDAAMSR's comparison

Substation1	Substation2	Similarity (%)
birkdale	cooroy	57.14
birkdale	brighton	57.14
birkdale	bribieisland	50.00
blackmountain	bribieisland	50.00
bribieisland	burpengary	46.43
bribieisland	brighton	50.00
bribieisland	carpendale	46.43
bribieisland	cooroy	46.43
brighton	birkdale	57.14
brighton	bribieisland	48.28
burpengary	bribieisland	46.43
calamvale	hollandpark	50.00

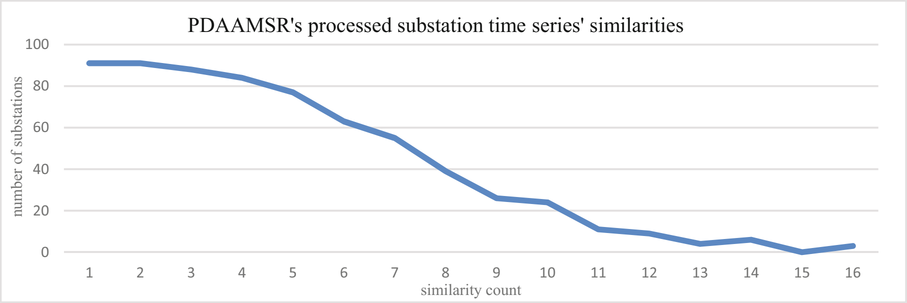


Fig. 5. Index Matrix's elements containing similar discretized substations' timeseries data points.

5 Discussion and Conclusion

From the experiments, it is noted that the MAE value can be controlled in conjunction with the amount of compression for time series using the differencing parameters. A higher difference limit value makes the compression even greater, saving more space and reduces the granularity but the loss is higher. This gives the user a greater flexibility to tune the algorithm to suit the situation. However, the differencing value must be fixed for the algorithm to be applicable generically across a group of time series, especially for substations' readings. This is to enable commonality and balance for all the converted time series' represented bits which are used subsequently for other purposes such as similarity comparison and group classification. Another one of our algorithm features compared to APAA is the requirement to normalize all the time series prior to the conversion process. The aim is to make it applicable to large volume and variety of time series that comprised of different measurement and intervals, whereas APAA is topical to a specific time series and rely on its real values.

The use of the index matrix for comparison process is viable. The degree of similarity among the substations' time series can be queried from the matrix with a two loop select, and it is faster to perform this task using SQL. However, this query can only be performed when all the times series have been represented and discretized, making this a batch process. New time series can be added to this system, but they must match the common characteristics of substation's time series. We still need to segregate the groups of the time series based on their characteristics for our method to function without running into the excessive higher dimensional scales for both window size and mean values, both of which must adhere to a reasonable range.

Time series are growing at an astronomical rate within the electricity utility domain and their growth will increase exponentially with the introduction of smart meters. While the utility companies spent a vast amount of money on hardware and software to support predictive analytics, storage consideration is always a keen area that requires substantial investment [1]. The proposed approach laid an alternate method of compressing large time series into smaller volumes that can represent the characteristics.

It also reshapes them by representing the compressed values into matrix coordinates that standardize them as well as lay the basis for future procedures such as comparison. Another application of our method is on text mining where observed text and their occurrence are expressed in rhythmic frequency time series term [16].

References

1. Esling, P., Agon, C.: Time-series data mining. ACM Comput. Surv. (CSUR) **45**(1), 12 (2012)
2. Bagnall, A., et al.: A bit level representation for time series data mining with shape-based similarity. Data Min. Knowl. Discov. **13**(1), 11–40 (2006)
3. Zhu, Z., et al.: Time series mining based on multilayer piecewise aggregate approximation, pp. 174–179. IEEE (2016)
4. Wu, Y.-L., Agrawal, D., El Abbadi, A.: A comparison of DFT and DWT based similarity search in time-series databases. In: Proceedings of the Ninth International Conference on Information and Knowledge Management. ACM (2000)
5. Notaristefano, A., Chicco, G., Piglione, F.: Data size reduction with symbolic aggregate approximation for electrical load pattern grouping. IET Gener. Transm. Distrib. **7**(2), 108–117 (2013)
6. Toshniwal, D., Joshi, R.C.: Finding similarity in time series data by method of time weighted moments. In: Proceedings of the 16th Australasian Database Conference, vol. 39. Australian Computer Society, Inc. (2005)
7. Keogh, E.: A decade of progress in indexing and mining large time series databases. In: Proceedings of the 32nd International Conference on Very Large Data Bases. VLDB Endowment (2006)
8. Keogh, E., et al.: Locally adaptive dimensionality reduction for indexing large time series databases. ACM Sigmod Rec. **30**(2), 151–162 (2001)
9. Tang, Q., et al.: Typical power load profiles shape clustering analysis based on adaptive piecewise aggregate approximation. In: IOP Conference Series: Materials Science and Engineering. IOP Publishing (2018)
10. Paparrizos, J., Gravano, L.: k-shape: efficient and accurate clustering of time series. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data. ACM (2015)
11. Eichinger, F., et al.: A time-series compression technique and its application to the smart grid. VLDB J. **24**(2), 193–218 (2015)
12. Burtini, G., Fazackerley, S., Lawrence, R.: Time series compression for adaptive chart generation. In: 2013 26th Annual IEEE Canadian Conference on Electrical and Computer Engineering (CCECE). IEEE (2013)
13. Ning, J., et al.: A wavelet-based data compression technique for smart grid. IEEE Trans. Smart Grid **2**(1), 212–218 (2011)
14. Guo, C., Li, H., Pan, D.: An improved piecewise aggregate approximation based on statistical features for time series mining. In: Bi, Y., Williams, M.-A. (eds.) KSEM 2010. LNCS (LNAI), vol. 6291, pp. 234–244. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15280-1_23
15. McLoughlin, F., Duffy, A., Conlon, M.: Evaluation of time series techniques to characterise domestic electricity demand. Energy **50**, 120–130 (2013)
16. Badea, I., Trausan-Matu, S.: Text analysis based on time series, pp. 37–41. IEEE (2013)

Applied Data Mining

Applying Data Mining Methods to Generate Formative Feedback in Team Projects

Rimmal Nadeem[1], X. Rosalind Wang[2], and Caslon Chua[3](✉)

[1] School of Computer Science and Engineering, UNSW, Sydney, Australia
rimmal.nadeem@student.unsw.edu.au
[2] Data61, CSIRO, Sydney, Australia
rosalind.wang@csiro.au
[3] School of Software and Electrical Engineering,
Swinburne University of Technology, Melbourne, Australia
cchua@swin.edu.au

Abstract. It is a challenging task to evaluate the performance of individual team members of a project, more so when there is unequal contributions by the team members. The project supervisor often provide formative feedback through assessment of weekly work logs against the agreed project plan. Automating this evaluation mechanism has the obvious advantage of providing timely feedback in an efficient manner otherwise not possible due to the workload on supervisors especially in the academic settings. We explore the design space of an automated formative assessment solution based on text data mining. We evaluate the performance of various machine learning techniques (both classification and regression versions), by tweaking their control parameters, based on the achieved accuracy of prediction. We showed that the KNN regression models, despite their simplicity, produce the most accurate result across different similarity methods. We verify these research findings by employing various data visualisation techniques such as learning curves, residual plots and goodness of fit.

Keywords: Educational data mining · Text mining · Visualisation

1 Introduction

The design and performance evaluation of data mining methods is a challenging task. Performance results are often presented in tabular forms, such as error measurements of a machine learning (ML) model's prediction, do not easily reveal whether the model was overfitting, had high bias/variance or how well it generalises to the data. Analysis is often limited to identifying the highest or the lowest value. Visualisation of a model's behaviour and its results simplify the decision making and result analysis process [1,6,11]. Visualisation offers a platform to help researchers think systematically about choices [8]. Vast information is more

R. Islam et al. (Eds.): AusDM 2018, CCIS 996, pp. 123–134, 2019.
https://doi.org/10.1007/978-981-13-6661-1_10

easily comprehended when they are presented in visually pleasing formats rather than large spreadsheets. Conclusions are much easily drawn and trends as well as relationships are easily identified when presented in charts and graphs [11].

In this study, we looked at improving previous works on an formative assessment data mining system that provides a formative mark for students based on their participation in a software project team [7,10], and utilise visualisation techniques to evaluate and fine tune the performance of ML models involved.

The main contributions of this study are two fold. First we address the limitations of the previous system. Second, we expand the previous work to increase its assessment accuracy, use additional prediction models for assessment and visualise the results to evaluate the prediction models.

2 Formative Assessment Data Mining System

Teamwork is inevitable when it comes to working on any sort of group project, regardless of which profession an individual is from.

In a university software engineering project, students aim to complete the project in a span of one semester, typically working in a team of four or five members. Students are often required to submit a work log in which they describe their contributions and progress for each week. An academic supervisor then assesses these logs each week against the project plan in order to provide formative performance feedback, identify issues and address problems that may be hindering team progress.

Although helpful, the team only receives feedback from their supervisors during weekly meetings. Small or minor issues that arise during the week may become large if not addressed early. Furthermore, the academic supervisors may supervise a large number of teams. Therefore, having a data mining system to assist in log analysis can be quite helpful.

We looked at the formative assessment data mining system [10], and extended the initial implementation [7]. The initial implementation was based on three machine learning methods using three features in the data set. In this study, we utilises visualisation to assist us in improving the implementation by expanding the number of features and machine learning methods used.

2.1 Experimental Data

The dataset used in this study is the same as previous work [7]. The data comprised of weekly work logs of the students and the project plan from 20 project teams. Each team is composed of four to five members working on software development, network, information systems or related projects. The data mining system analyses the project plan from each team, and work logs that each team member submitted over one semester. The work logs are compared to the project plan to assess each team member's performance.

2.2 Experiment Set Up

We follow the same experimental set up described in the previous work [7] for mining the work log data and generating the formative feedback. The dataset is pre-processed with natural language processing algorithms before inputting them to the machine learning method.

Similarity Techniques and Text Cleaning Methods. Following the experimental set up in the previous work [7], the logs are first normalised and cleaned through natural language processing algorithms. Normalisation involved punctuation removal, case-sensitivity removal, and tokenisation of sentences into words. Cleaning involved stop-words removal, where words do not carry any significant meaning on their own are removed, and stemming, where specific words are transformed back to its canonical form. Multiple text similarity algorithms (both lexical and semantic) are then employed to compare the work logs against the project plans to determine the student's current progress in relation to the plan. The text similarity algorithms implemented include Dice's coefficient, Cosine similarity, Jaccard similarity, N-gram similarity and Latent Semantic Analysis (LSA). We used these methods with no modifications.

Machine Learning Models. In previous implementation, three ML models were evaluated, namely Support Vector Machines (SVM), Linear Regression and K-Nearest Neighbours (KNN). The SVM classification and regression models were evaluated using the default kernel of radial basis function (Rbf). This study extended the evaluation to include linear and polynomial (poly) kernel functions. The KNN models were evaluated using the default parameter of uniform weights (UW), and is extended to include distance weights (DW).

We further added three ML models for evaluation. These include Naïve Bayes (NB), Random Forests (RF) and Neural Networks (NN). The additional algorithms are chosen due to their proven popularity and efficiency in various forecasting applications [5,13,14].

Additionally, in order to fine tune our predictive models we utilised cross validation with parameter tuning using grid search [4]. To find the optimal number of trees for random forests, we employed the Out-of-Bag error [2]. Neural networks were evaluated with 2–4 different numbers of hidden layers. These measures made our analysis more comprehensive.

Both regression and classification versions of SVM, RF, NN and KNN were implemented. Henceforth, we will use '-R' and '-C' tags for each ML methods to indicate regression and classification respectively.

Cross-Validation. Five-fold cross validation (CV) was employed to generalise the result and avoid as much bias as possible. The original data set is randomly divided into 5 equal sized partitions. One partition is used for test, and the rest are used for training. The selected partition for test is then rotated repeatedly such that all partitions are selected one by one. Cross validation was used for all of the ML methods and their different parameters.

2.3 Fine Tuning Methods

Features for Learning. In previous work [7], all ML models were trained using three features from the work log, namely work week, work duration and similarity. In this study, we added a fourth feature, previous mark, to evaluate its effect on prediction accuracy.

Week-Wise Comparison. Evaluating the performance of the data mining system and the ML models utilised, we found that errors in prediction were made for the case when the student's submitted progress is not perfectly aligned with the project plan. The previous similarity comparison was only done between the progress and project plan of a single week. This produces high errors as the model predicts low marks for not matching the current week's projected plan. This is reasonable for the case when the student is falling behind. However, it is unfair and inaccurate when the student is progressing ahead. The academic supervisor awards high marks for students who are progressing ahead.

To overcome the limitation of students being marked unfairly when working ahead of project plan, a week-wise comparison (WWC) similarity check algorithm is applied. This algorithm discovers the best matching week and registers whether the best match is behind or ahead of what is required for the current week. If it is behind the similarity feature, the current week's similarity value stays the same as before, however if it is ahead the similarity feature value is increased to the best matched value.

2.4 Evaluation Methods

As in previous work [7], different permutations of similarity measures were implemented in order to identify which technique yields the best result. Before returning the similarity score, the WWC algorithm is used in order to cater for unfair grading. The resulting similarity score is then normalised accordingly to match with the formative score range of 0 to 10 that is provided to the student.

Both regression and classification versions of each ML method are evaluated to determine the best performing model. For classification, classes are defined as the integer scores from 0 to 10; while for regression approach, the model gave an estimation of student's score as a real number within the range of 0 to 10 and were evaluated as it is.

Root Mean Squared Error (RMSE) was used to evaluate the accuracy of the final prediction results for both regression and classification techniques. This helps identify how close the predicted value is to the real value for any model.

Adjusted R-squared ($\mathbf{R^2}$) Score was analysed to determine how well the regression models fit our data (goodness of fit). The score is a statistical measure of how close the data approximates to the fitted regression line. The scores normally range between 0 and 1, however it can also produce negative values. A score of 1 indicates that the regression fits perfectly with the data. Generally, the closer the score is to 1 the better is the fit.

Data Visualisation Techniques were utilised to ascertain the validity of our prediction and learning models. The visualisation techniques used include:

- **Heat maps:** used to perform analysis on both R^2 and RMSE results. Heat map allows us to visually identify the best and worst methods, as well as possible trends by observing the colour changes.
- **Learning Curves** [12]**:** show how well a learning model can generalise to new data. Learning curves allow us to verify when a model has learned as much as it can about the data. This is signalled when the performances on the training and test sets reach a plateau or when there is a consistent gap between the two error rates with the increase in training samples.
- **Residual plots and Histograms:** help understand and improve regression models [3]. A model is good if there is a strong correlation between the model's predictions and its actual results. Furthermore, the corresponding histogram for the residuals should have a close to normal distribution with a mean of approximately zero.

3 Implementing Data Visualisation on the Machine Learning Model Results

With the data visualisation identified, we applied these on the results to analyse and fine tune the models that we are currently evaluating. In this section, we present an analysis of the top four models and worst model in Table 1.

3.1 Fine Tuning Using Heat Maps

Normally, RMSE and R^2 values are presented in a table for analysis and evaluation. Analysis involves looking for the lowest or highest values respectively to determine which model generate it. However, this tasks can be quite tedious and identifying if a trend exists can be difficult. In this study, we applied the heat map visualisation in performing the ML fine tuning process.

We visualised the RMSE values of the models that uses three and four features. The heat map of four features (Fig. 1(a)) showed the best models – KNN-R UW, KNN-R DW, RF-R and NN-R (4) – had slightly improved performance compared to that of the three features. Moreover, the worst performer – SVM-R (poly) – had very little improvement.

Finally, the models were further fine-tuned to incorporate WWC along with four features. The heat map of the RMSE shown in Fig. 1(b) continue to have KNN-R UW, KNN-R DW and RF-R as the best performers. The NN-R (4) resulted to a slightly poorer performance, however SVM-R (poly) resulted to a substantial improvement for N-gram (n-2) with stop words removed.

In this study, visualising the RMSE enables us to quickly analyse the outcome of fine tuning the models. By looking at the changes of colour towards darker blue from Fig. 1(a) to (b), we can quickly identify which model had improved in performance. WWC also improved the scale of the heatmap. In terms of

Table 1. A summary of experimental results using 4 features and week-wise comparison. This table contains the overall observations made through the results collected. It provides a basic description of the top four performing models and the worst performing model (SVM-R (Poly))

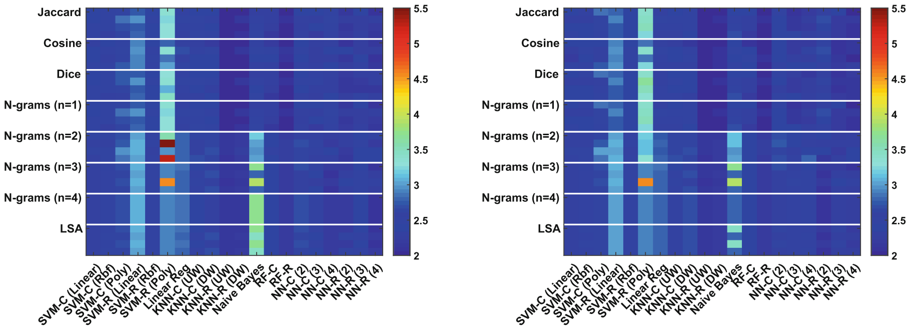
ML method	RMSE	R^2	Learning curves	Residual plots	Histogram
KNN-R (DW)	1.924	0.49	Contains variance, no bias	Good correlation between actual and prediction	Close to Normal Distr.
KNN-R (UW)	1.926	0.47	Biased with high training errors	Over estimation in prediction	Skewed to the left
RF-R	1.932	0.51	Contains variance, no bias	Good correlation between actual and prediction	Close to Normal Distr.
NN-R (4)	2.045	0.54	Biased with low training errors	Avg correlation, little over est. errors in prediction	Close to Normal Distr.
SVM-R (Poly)	4.561	−4.25	Biased with high errors for CV	Over estimation in prediction	Skewed to the left

analysing the cleaning methods applied to each similarity techniques, the heat map shows that none of the cleaning methods affect the performance of each similarity technique. Except for SVM-R (Poly) model, where each similarity techniques shows that each cleaning method resulted to different colour shades.

To measure the goodness of fit of the regression models using four features with week-wise similarity comparison, heat map is again selected as the visualisation technique. This time, we look for darker red to indicate goodness of fit. Looking at the heat map in Fig. 2, SVM-R (Rbf) has the best fit followed by KNN-R (UW) and RF-R. Once again, SVM-R (Poly) performed the worst showing patches of colours other than red.

3.2 Detecting Bias and Variance Using Learning Curve

A ML model produces an ideal learning curve when it can generalise to new data, and the test and training curves converge at similar low error values i.e., the gap between the test and training curves should be small. A learning curve shows sub-optimal performance of the underlying learning model when it produces either high bias or high variance. A high bias is shown when the training and test errors have converged but both show high error values. No matter how much data we feed to the learning model showing bias, the model cannot truly represent the underlying relationship and produces high systematic errors. The model thus requires additional or more complex features to truly represent the actual behaviour. On the other hand, a learning model that exhibits high variance



(a) Original 4 features

(b) Week-wise Comparison 4 features

Fig. 1. RMSE results for all ML methods. The y-axis shows all similarity algorithms tested. For each similarity algorithm, four text cleaning methods were used (from top): none, stop-word removal, stemming, or both. The x-axis shows the ML models with the corresponding parameters they were evaluated with. Note that in (a) the highest value is 7.95, we limited the maximum to 5.5 for better comparison between the two figures. (Colour figure online)

through a large gap between both errors in learning curves requires additional training samples to properly converge. Another option to reduce variance is to reduce the number of features or employing less complex features.

To further analyse the top performing models, learning curve visualisation is implemented. Using the results from the models that uses four features and week-wise comparison, we visualise the learning curves for KNN-R (DW), KNN-R (UW), RF-R and NN-R (Fig. 3). For comparison purposes we also visualise the learning curves for SVM-R (Poly), which is the worst performing model (Fig. 4).

Analysing the learning curve, KNN-R (DW) and RF-R show good fit. However, both do exhibit a bit of variance, which can be improved by providing additional training data or trying smaller set of features. To validate this, learning curve was generated for KNN-R (DW) using only three features and week-wise comparison (Fig. 5). Comparing the learning curves (Figs. 3(a) and 5), a noticeable trend is that KNN-R (DW) with four features and week-wise comparison converges faster and produces a lower error compared to only using three features. This visualisation showed that using an additional feature has in fact improved the results, thus the next best option to lower the variance and further improve the prediction results is to add more training data.

The learning curve for KNN-R (UW), however, shows that although the model has good RMSE (Fig. 1), it is biased, flat towards end and cannot be improved by simply adding more training samples as both the training and test error are high. For the case of SVM-R (Poly), the model had the highest RMSE that is validated by the very high training and test error shown in its learning curve visualisation (Fig. 4). Additionally, we can see in Fig. 3, a useful demonstration of the use of learning curves for two methods (KNN-R (UW) and

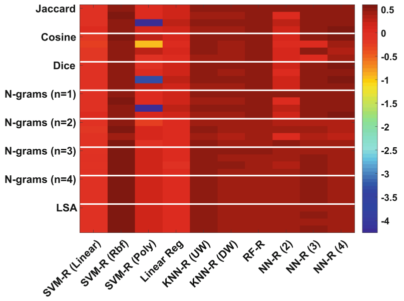


Fig. 2. Adjusted R^2 results for all regression models. The x and y-axes are the same as Fig. 1. (Colour figure online)

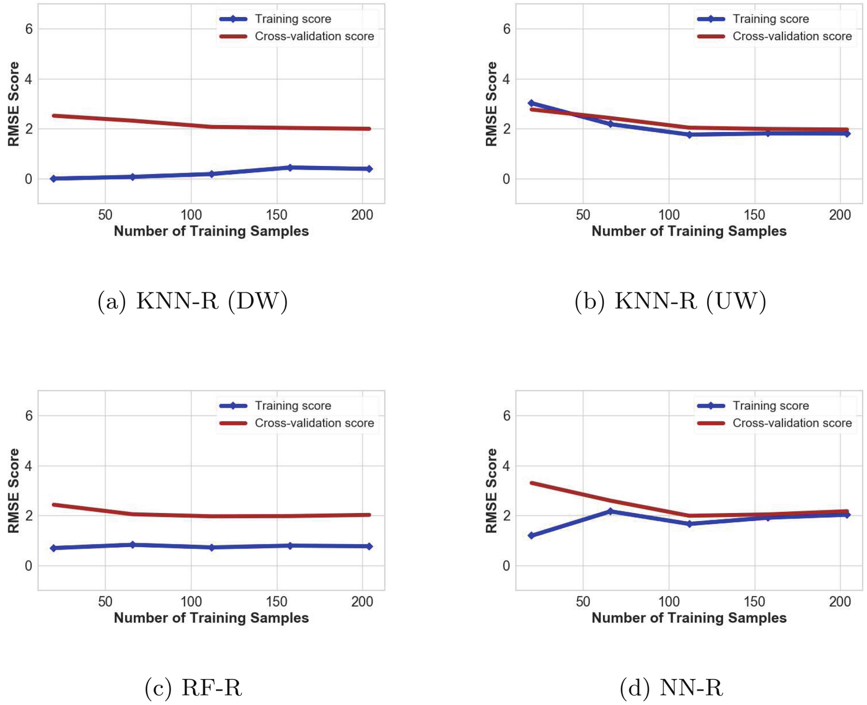
KNN-R (DW)) with very similar RMSE and R^2 values. Despite having similar values for RMSE and R^2, KNN-R (UW) (Fig. 3(b)), is highly biased compared to KNN-R (DW) (Fig. 3(a)).

3.3 Identifying the Best Models Using Residual Plot and Histogram

We next analysed residual plots and their corresponding histograms for ML models and text analysis methods with the best R^2 results. Ideally the plot should be symmetrically distributed around residual of 0, tending to cluster at a lower value towards 0. Moreover, it should not present any clear pattern or trend. The errors should be normally distributed with a mean close to zero for each value of outcome. If a model is of good fit, the corresponding histogram for residual plots is close to a normal distribution with zero mean.

We found the best models are: KNN-R (DW), RF-R and NN-R(4) (Fig. 6). This is because there was a good correlation shown between the model's predictions and its actual results through the minimum scattering and high clustering towards residual of 0. SVM-R (Poly) remained as the worst model (Fig. 7), highly overestimating and also producing high errors with the training data. We also observe from the residual plots that all models seem to predict the marks well between 5–10 while in most cases the results produce relatively more errors in the range 0–4. This observation was made through observing the fact that most residuals are negative which implies most of the models were overestimating. This also indicated the academic supervisor's marks tend to be more lenient in the range of 0–4 than our prediction models. The observation is also supported through the histograms being skewed to the left. Finally, there is no clear pattern/outstanding trend evident in the residual plots.

The histograms show that all best performing learning models including KNN-R(DW) produce almost symmetrical normal distribution with a mean close to zero (showing -1), indicating a good fit. This suggests that addition of more training data would help reduce the residual errors, and reinforcing the finding from the learning curves.



(a) KNN-R (DW)

(b) KNN-R (UW)

(c) RF-R

(d) NN-R

Fig. 3. Learning curves for the top four performing models using 4 features and week-wise comparison.

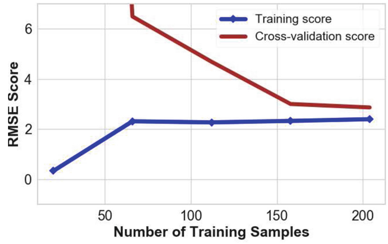


Fig. 4. Learning curve for worst performing model using 4 features and week-wise comparison-SVM-R (Poly).

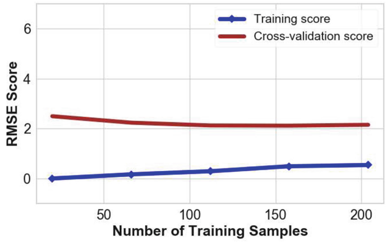
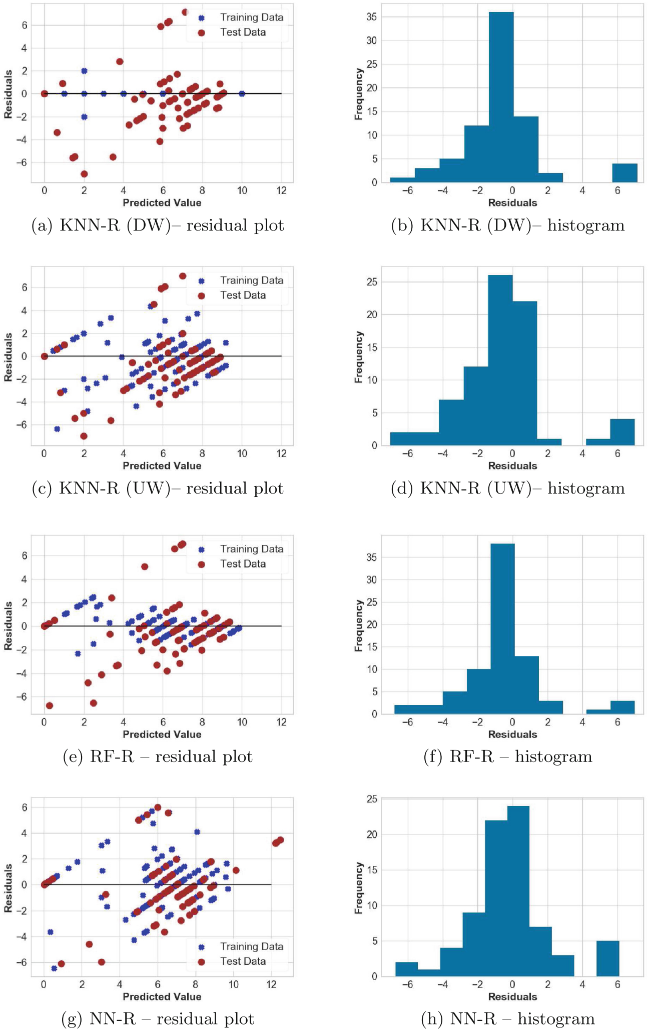


Fig. 5. Learning curve for KNN-R (DW) using 3 features and week-wise comparison.



(a) KNN-R (DW)– residual plot

(b) KNN-R (DW)– histogram

(c) KNN-R (UW)– residual plot

(d) KNN-R (UW)– histogram

(e) RF-R – residual plot

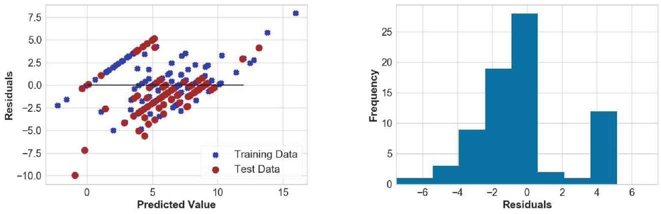
(f) RF-R – histogram

(g) NN-R – residual plot

(h) NN-R – histogram

Fig. 6. Residual plots and histograms for the top four performing models according to the residuals using 4 features and week-wise comparison.

From our learning curves and residual plot results, we were able to determine that SVM-R (Rbf) kernel model and KNN-R (UW) had high bias. From the RMSE results (Fig. 1) it was already seen that SVM-R (Rbf) produced the worst results however had a high R^2 value (Fig. 2). This highlights that even though the R^2 results for these models were high, they are not a good fit. For KNN-R (UW) (Fig. 6(c) and (d)), we observe that it is mostly overestimating (skewed to the left with most residuals in the negative) and it does not follow a normal distribution. It can be improved by introducing additional or more complex features that can capture the true behaviour.



(a) SVM-R (Poly)– residual plot (b) SVM-R (Poly)– histogram

Fig. 7. Residual plots and histograms for the worst performing model using 4 features and week-wise comparison.

From the visualisation results, we can conclude that KNN-Regression (DW) (Fig. 6(a) and (b)) is indeed the best model with a good fit. It is closely followed by RF-R and NN-R (4) (Fig. 6(e)–(h)). There is, however, still room for improvement as these models can be made more accurate by providing more training data.

4 Conclusion and Recommendation

In this case study, we showed that ML methods can be used to generate formative feedback on team projects. In addition, visualisation allowed analysis that enabled deeper understanding of the learning model behaviour.

In evaluating how well the fine-tuning through visualisation worked, the marks generated by the fine-tuned system and the marks awarded by the academic supervisors were manually compared. This was done through comparing the predicted marks from the ML methods to the given scores in the dataset. This manual analysis, in addition to analysing the ML performance metrics and visualisation results, allowed us to determine if our fine-tuned implementation is a suitable method to automatically generate formative feedback marks for students. The fine-tuned system was successful in producing better results than previous work [7] and generated formative feedback marks that matched the marks provided by the academics. In actual use, it is to be noted that the generated feedback marks will still be moderated by the supervisors prior to releasing the marks to the students.

Future work will aim to design an interactive data visualisation tool [9], which will enable supervisors to view and modify marks produced by our system and allows students to view their performance based on formative assessments. In addition, an interactive visualisation also allows supervisors to identify students who are struggling and can offer support quickly and efficiently.

Future work will also look at personalised prediction, comparable to work on personalised student performance forecast [14], through mining other student information.

References

1. Almasoud, A.M., Al-Khalifa, H.S., Al-Salman, A.: Recent developments in data mining applications and techniques. In: 2015 Tenth International Conference on Digital Information Management (ICDIM), pp. 36–42, October 2015
2. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
3. Deo, S.: R tutorial: residual analysis for regression, March 2016
4. Hsu, C.W., Chang, C.C., Lin, C.J., et al.: A practical guide to support vector classification (2003)
5. Huebner, R.A.: A survey of educational data-mining research. Res. High. Educ. J. **19**, 1–13 (2013)
6. Jin, H., Liu, H.: Research on visualization techniques in data mining. In: 2009 International Conference on Computational Intelligence and Software Engineering, pp. 1–3, December 2009
7. Le, K., Chua, C., Wang, R.: Mining software engineering team project work logs to generate formative assessment. In: International Workshop on Software Driven Big Data Analytics (SoftBDA 2017) (2017)
8. Munzner, T.: Visualization Analysis and Design. CRC Press, Boca Raton (2014)
9. Murray, S.: Interactive Data Visualization for the Web. O'Reilly Media, Inc., Sebastopol (2013)
10. Nguyen, T., Chua, C.: Predictive tool for software team performance. In: 2016 23rd Asia-Pacific Software Engineering Conference, pp. 373–376, December 2016
11. Obie, H.O.: Data—driven visualisations that make sense. In: 2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), pp. 313–314, October 2017
12. Perlich, C.: Learning curves in machine learning, January 2011
13. Shahiri, A.M., Husain, W., Rashid, N.A.: A review on predicting student's performance using data mining techniques. Procedia Comput. Sci. **72**, 414–422 (2015)
14. Thai-Nghe, N., Horvth, T., Schmidt-Thieme, L.: Personalized forecasting student performance. In: 2011 IEEE 11th International Conference on Advanced Learning Technologies, pp. 412–414, July 2011

A Hybrid Missing Data Imputation Method for Constructing City Mobility Indices

Sanaz Nikfalazar[1](✉), Chung-Hsing Yeh[1], Susan Bedingfield[1], and Hadi Akbarzadeh Khorshidi[2]

[1] Monash University, Clayton, VIC 3800, Australia
Sanaz.nikfalazar@monash.edu
[2] The University of Melbourne, Melbourne, VIC 3010, Australia

Abstract. An effective missing data imputation method is essential for data mining and knowledge discovery from a comprehensive database with missing values. This paper proposes a new hybrid imputation method to effectively deal with the missing data issue of the Mobility in Cities Database (MCD) to construct city mobility indices. The hybrid method integrates the advantages of decision trees and fuzzy clustering into an iterative algorithm for missing data imputation. Extensive experiments conducted on the MCD and three commonly used datasets demonstrate that the hybrid method outperforms other existing effective imputation methods. With the MCD's missing values imputed by the hybrid method, and using factor analysis and principal component analysis, this paper constructs city mobility indices for 63 cities in the MCD based on the novel concept of city mobility supply and demand. The city mobility indices constructed under a hierarchical structure of mobility supply and demand indicators represent substantial city mobility knowledge discovered from mining the MCD. The proposed hybrid method represents a significant contribution to missing data imputation research.

Keywords: Missing data imputation · City mobility index · Factor analysis · Principal component analysis · Decision tree · Iterative fuzzy clustering

1 Introduction

Transport plays a significant role in the sustainable development of cities due to various aspects such as social, economic and environmental [1, 2]. Mobility is a principal element of sustainable development and is known to be one of the most important requirements for achieving improved standards of living. Mobility is defined as the rapid, easy and inexpensive movement of people and goods to their destinations [3].

The Mobility in Cities Database (MCD) is a comprehensive database provided by the International Association of Public Transport (UITP). This database has been released in the years 1995, 2001 and 2015. The MCD has been used in several studies as the main source of data. The main reason for using the data was to develop a city mobility index with the aim of comparing and ranking cities with distinctive characteristics. Haghshenas and Vaziri [4] proposed a city mobility index using a combination of urban transport sustainability indicators using MCD 1995 and compared various

R. Islam et al. (Eds.): AusDM 2018, CCIS 996, pp. 135–148, 2019.
https://doi.org/10.1007/978-981-13-6661-1_11

cities based on the proposed index. Moeinaddini et al. [5] developed an urban mobility index for evaluating effective indicators for transport in cities using the MCD 2001. However, the MCD database has a significant amount of missing values that could affect the precision of the calculation. An effective missing data imputation method could improve data mining outcomes from a database such as the MCD.

In this study, we use the latest version of the MCD which contains mobility information for 63 cities around the world. To use the MCD (with many missing values) to examine the mobility of 63 cities with all the relevant indicators, we propose a new hybrid imputation method by integrating decision trees and fuzzy clustering to impute the missing values. To better understand the city mobility issues from different perspectives, we develop a new concept of city mobility supply and demand. New mobility indices are constructed based on the supply and demand aspects of city mobility using factor analysis and principal component analysis. The contribution of this paper is two-fold. First, this study makes a methodological contribution by developing a hybrid imputation method which can be used to effectively impute missing values for a comprehensive dataset. Second, this study provides new insights into city mobility comparisons under different contexts by developing a new concept of city mobility supply and demand.

In subsequent sections, we first discuss how we select the indicators from the MCD to represent the supply and demand dimensions of city mobility in Sect. 2. We then explain how we develop a new hybrid imputation method to address the missing data issue of the MCD in Sect. 3. In Sect. 4, we conduct extensive experiments to examine the performance of the proposed hybrid imputation method against other effective imputation methods. In Sect. 5, we use factor analysis to discover the hierarchical structure for mobility's supply and demand indicators with the imputed MCD dataset. In Sect. 6, we apply principal component analysis to construct city mobility indices for comparing mobility supply and demand of 63 cities. Finally, we summarise the key findings and contributions of this study in Sect. 7.

2 City Mobility Indicators

The concept of mobility in cities has been mainly focused on a framework of sustainability. However, city mobility is a multifaceted concept which includes accessibility, welfare, social and economic equity. Mobility can affect the meaning of equity, freedom of choice for activities and living conditions within a given cultural context. In this study, to have a universal interpretation of the issue, mobility is defined as a basic need of individuals within communities to reach desired destinations. This is achieved through using mobility services and facilities provided. The services and facilities should be accessed and function equally for everyone.

To address the city mobility issue in this regard, we investigate and evaluate the mobility of cities from the perspectives of supply and demand. The literature on city mobility supply and demand is relatively limited. Transport supply and demand have been studied in the field of transport economics, at either local or regional geographical scope [6]. Exploring mobility internationally and comparing cities worldwide, based on their supply and demand is an innovative approach. Albalate and Bel [6, 7] considered mobility supply and demand for European cities using econometrics methods to verify the relationship of supply and demand for transport. Their work is the first using an international sample, jointly investigating supply and demand of city mobility.

To perform a comparative analysis on city mobility supply and demand, it is essential to choose appropriate indicators to represent those dimensions adequately. Numerous studies have analysed the characteristics of qualified and appropriate indicators for comparative evaluations. Indicators must be selected according to the availability of data and features like comprehensiveness, target relevance, measurability, validity, reliability and transparency [4, 8, 9]. Moreover, selecting indicators should focus on the questions that the indicators are aiming to answer [10].

To represent the supply and demand dimensions of city mobility, 25 indicators are selected from the MCD [11]. Consequently, a mobility supply and demand dataset with 63 cities (rows) and 25 indicators (columns) is created. These indicators are shown in Table 1, including 12 supply indicators and 13 demand indicators. The supply indicators are concerned with the availability of transport facilities and mobility options. Therefore, they represent what is supplied for city mobility. The demand indicators represent users' perspective since they indicate various choices and transport needs of users. These demand indicators reflect what users' mobility needs are.

Table 1. Mobility supply and demand indicators

Supply dimension		Demand dimension	
Indicator	Indicator description	Indicator	Indicator description
S_1	Proportion of the urbanised surface used for local passenger transport	D_1	Annual passenger car vehicle kilometres per metre of road
S_2	Length of road per thousand inhabitants	D_2	Annual average distance travelled in passenger cars per inhabitant
S_3	Length of motorway per thousand inhabitants	D_3	Total public transport vehicle kilometres per inhabitant
S_4	Length of road per urban hectare	D_4	Total public transport vehicle kilometres per urban hectare
S_5	Length of motorway per urban hectare	D_5	Total public transport journeys per inhabitant
S_6	Passenger cars per thousand households	D_6	Total public transport passenger kilometres per inhabitant
S_7	Average annual distance travelled per passenger car	D_7	Average public transport place occupancy rate

(*continued*)

Table 1. (*continued*)

Supply dimension		Demand dimension	
Indicator	Indicator description	Indicator	Indicator description
S_8	Average speed on the road network	D_8	Daily trips per inhabitant
S_9	Total public transport vehicles per million inhabitants	D_9	Daily motorised trips per inhabitant
S_{10}	Total public transport place kilometres per inhabitant	D_{10}	Average length of a private motorised trip
S_{11}	Total public transport place kilometres per urban hectare	D_{11}	Average length of a public transport trip
S_{12}	Average public transport speed	D_{12}	Average duration of a private motorised trip
		D_{13}	Average duration of a public transport trip

3 The Hybrid Missing Data Imputation Method

The indicators shown in Table 1 comprise the mobility supply and demand dataset used in this research. Thirty six percent of the values in this dataset are missing. As each record in the dataset represents a city, to compare cities based on mobility, all missing values must be estimated. To impute the missing values, we propose a new hybrid imputation method, called DIFC, by integrating decision trees and iterative fuzzy clustering. The DIFC method is an integration of supervised (decision trees) and unsupervised (iterative fuzzy clustering) machine learning methods.

To the best of our knowledge, no imputation method has been applied on the MCD prior to [12] where an iterative fuzzy clustering algorithm is used to impute missing data in the MCD database. This algorithm improves the imputed values through iterations. To further improve the imputation accuracy of the missing values in MCD, we incorporate the decision tree method into the iterative fuzzy clustering algorithm. The main advantage of this new approach is that the imputation accuracy is likely to be higher when records are similar to each other [13]. In DIFC, a decision tree divides the dataset into similar subsets. Among each subset, fuzzy clustering updates the imputed values iteratively.

Decision trees segment the predictor space into several regions to predict given records. Decision trees are close to human decision making [14]. Fuzzy clustering groups records into several subsets (clusters) in a way that attribute values of records within each subset are similar while different subsets are distant. However, the borders of these subsets are uncertain. The fuzzy c-means clustering method, which is the most well-known fuzzy clustering method, allocates records to clusters with a set of membership degrees which indicate to what extent a record belongs to the clusters [15]. The steps of the hybrid DIFC method are described as follows:

Step 1: Normalise all variables to within the range between 0 and 1 as:

$$x_{ij}^{N} = \frac{x_{ij} - \underset{j}{min}\ x_{ij}}{\underset{j}{max}\ x_{ij} - \underset{j}{min}\ x_{ij}},\ i:\ 1,\ldots,N;\ j:1,\ldots,P \tag{1}$$

where N is the number of records (cities in the MCD), P is the number of variables, $\underset{j}{min}\ x_{ij}$ and $\underset{j}{max}\ x_{ij}$ are the minimum and maximum values in jth column respectively, x_{ij}^{N} is the normalized value, and x_{ij} is the original non-missing value.

Step 2: Build a set of decision trees based on variables having missing values. The number of decision trees is the same as the number of variables having missing values. A decision tree is built for each variable with missing values, so that the end nodes (leaves) indicate separate groups of the targeted variable and decision nodes are based on other variables.

Step 3: Impute the missing values initially, based on the end nodes to which the records are assigned. Each missing variable of a record is assigned to its associated decision tree. The missing value is then imputed using the average of records that are in that end node.

Step 4: Determine an optimal number of clusters within records in each end node. The optimal number of clusters is found using the fuzzy silhouette (*FS*) index [16]. The maximum value of *FS* denotes the optimal number of clusters. *FS* is formulated as:

$$FS(k) = \frac{\sum_{i=1}^{N} \left(u_{ig} - u_{ig'}\right)^{a} S_i(k)}{\sum_{i=1}^{N} \left(u_{ig} - u_{ig'}\right)^{a}} \tag{2}$$

where U is the membership degree matrix with records as rows (i) and clusters as columns (k), u_{ig} and $u_{ig'}$ are the largest and the second largest elements of the ith row respectively, a is a weighting coefficient (equal to 1 as a default), and $S_i(k)$ is the silhouette of record i, and is calculated by:

$$S_i(k) = \frac{b_i - a_i}{\max(a_i, b_i)} \tag{3}$$

where a_i and b_i are the average dissimilarity between record i and other records in the same cluster and the lowest average dissimilarity of record i to other clusters to which record i does not belong respectively. $S_i(k)$ has a value ranging from −1 to 1, and it is set to zero when a cluster contains just one record.

Step 5: Apply fuzzy clustering to records assigned to each end node. The membership degree and clusters' centroids, provided by the fuzzy clustering, are used to improve the initial decision tree imputation. Fuzzy clustering provides an opportunity to estimate a weighted average for imputation among assigned records of each end node. The similar records achieve higher weights using fuzzy clustering.

Step 6: Update the previous imputation values (*PV*). A new value (*NV*) is estimated using the membership degrees and centroids, obtained from fuzzy clustering, as:

$$NV_{ij} = PV_{ij} + \frac{\sum_{g=1}^{k} u_{ig} \times c_{gj} - PVi_j}{T}, \quad \forall (i,j) \in MS \tag{4}$$

where MS is the set of coordinates of missing values, u_{ig} is the membership degree of the ith record in cluster g, c_{gj} shows the value of jth variable in centroid of the gth cluster, and T is a real value greater than 1. The imputed value estimated by fuzzy clustering updates the *PV*.

Step 7: Terminate the fuzzy clustering iterations if a stopping criterion is met. If the average variation (AV) of the imputed values in comparison with the previous iteration is less than a specified threshold, the iteration stops. The formula for AV is shown in (5). If the stopping criterion is not met, go to Step 5 to continue the iteration.

$$AV = \sum\nolimits_{(i,j) \in MS} |NV_{ij} - PV_{ij}| \tag{5}$$

Step 8: De-normalisation. Once the iteration stops, the values in the final iteration are de-normalised to retrieve the original values. Figure 1 shows the schematic flowchart of the hybrid DIFC imputation method.

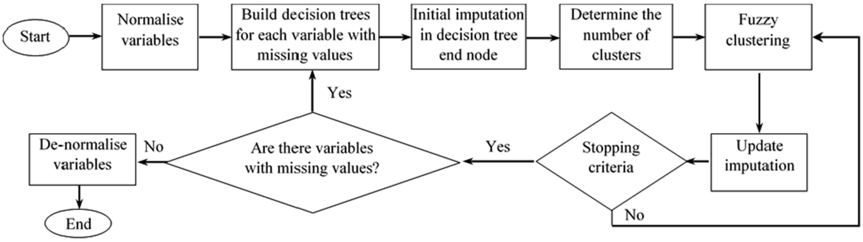

Start
Normalise variables
Build decision trees for each variable with missing values
Initial imputation in decision tree end node
Determine the number of clusters
Fuzzy clustering
Yes
De-normalise variables
No
Are there variables with missing values?
Yes
Stopping criteria
Update imputation
End
No

Fig. 1. The hybrid DIFC imputation flowchart

4 Performance Evaluation of DIFC

To investigate how well the hybrid method works, extensive experimental analyses have been conducted. Further missing values are artificially created in the MCD dataset to measure the error of imputation in comparison with the actual values. Two accuracy measures, root mean squared error (RMSE) and mean absolute error (MAE), are used to evaluate the imputation error. As shown in (6), RMSE penalises the imputed values which are more distant from the actual values in comparison with MAE, as shown in (7), which gives an equal influence from each error [17].

$$RMSE = \sqrt{\frac{1}{M}\sum_{m=1}^{M}(I_m - A_m)^2} \tag{6}$$

$$MAE = \frac{1}{M}\sum_{m=1}^{M}|I_m - A_m| \tag{7}$$

where I_m and A_m are the imputed and actual values for the mth missing value respectively and M is the number of created missing values. We compare the performance of DIFC with two existing imputation methods, which have been proven to be effective and whose concepts are used in DIFC. These methods are iterative fuzzy clustering (IFC) [12], and decision trees (DT) [18]. We use two derived data sets with increased missing values of 1% and 4% of the total MCD dataset respectively. This results in 16 and 63 more missing values respectively. These artificial missing values are added in the presence of actual missing values, making the total percentage of missing values be 37% and 40% respectively. Each record at most has one artificial missing value. For each derived dataset, 10 different combinations of missing values are generated, providing a 10-fold cross-validation. The average error of these combinations is used to compare the performance of methods. Table 2 shows the accuracy measures for DIFC and four effective methods.

Table 2. RMSE and MAE comparison

Percentage increase in missing values	RMSE			MAE		
	DIFC	IFC	DT	DIFC	IFC	DT
1	0.191	0.239	0.203	0.135	0.167	0.144
4	0.184	0.211	0.191	0.132	0.153	0.137

The results of RMSE and MAE comparisons suggest that DIFC outperforms two relevant and effective methods across both derived datasets with two different ratios of missing values. To compare the performance of DIFC with other existing methods, we conduct experiments on three commonly used datasets from the UCI machine learning repository, including Adult, Yeast and Pima Indians Diabetes. To conduct the experiments, we randomly remove some data points in each data set to create artificial missing values. As a result, different missing value ratios of 1%, 3%, 5% and 10% are created. The missing value ratio is the percentage of total values of a data set that is missing.

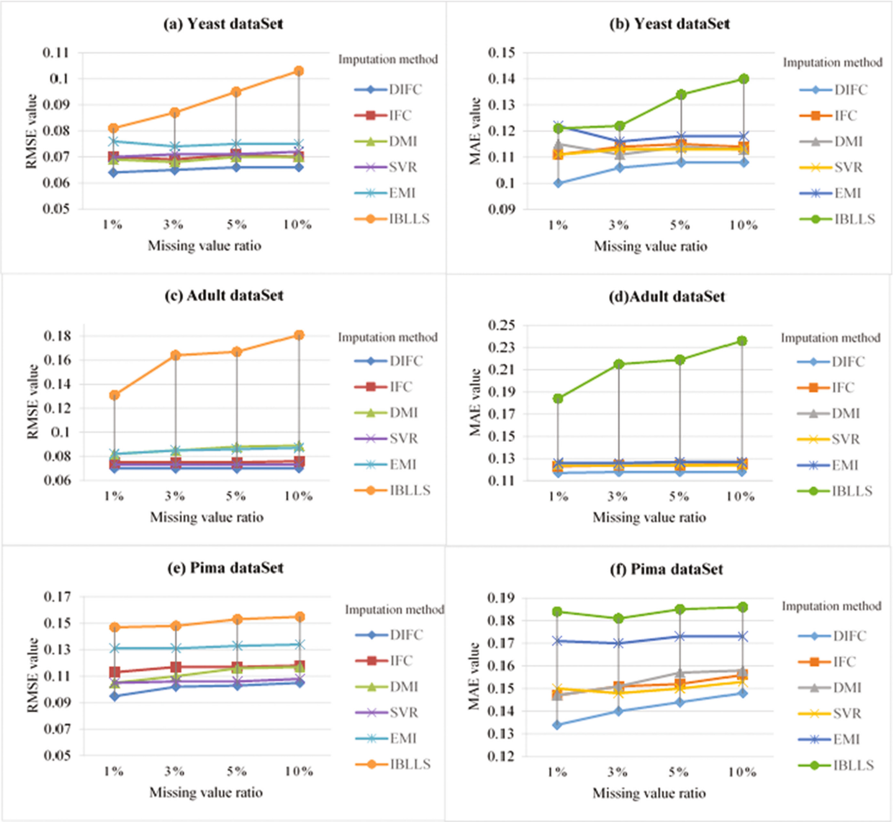


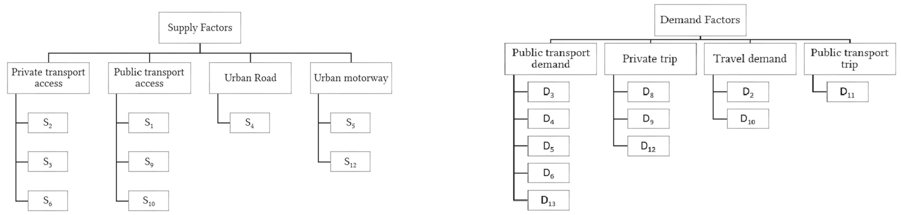
Fig. 2. RMSE and MAE comparison of six imputation methods on three datasets

The performance of DIFC is compared with five existing imputation methods, which are IFC [12], decision tree based missing value imputation (DMI) [19], iterative bi-cluster based local least square (IBLLS) [20], SVR [21] and expectation maximization imputation (EMI) [22]. Figure 2 shows the RMSE and MAE comparison from the experiment results. The experimental results show that DIFC imputes more accurately than existing imputation methods.

5 Factor Analysis for Dimension Reduction

The first step in constructing the city mobility indices is to apply factor analysis to the MCD dataset with imputed values for missing data using DIFC. The purpose of factor analysis is to summarise the contents of some variables into smaller sets of variables (factors). This is intended to reveal the hidden structure among the original variables [23]. The structure design is the most prominent characteristic in developing composite indices. The hierarchical structure is one type of structure design which provides an opportunity to aggregate indicators into sub-indices, and then aggregate the sub-indices into one index [24]. In this paper, we develop a new hierarchical structure for city mobility's supply and demand indicators, which has not been addressed in the literature. Therefore, an exploratory factor analysis (EFA) is performed separately on supply and demand indicators to find the influential indicators and to reduce the influential indicators into the most influential factors by merging indicators that are highly correlated [25]. In EFA, all variables are related to every factor according to the factor loadings estimations. However, the correlated factor can be achieved when each variable load is high on one factor and has smaller loadings on other factors [23]. Once the hierarchical structure is constructed the sub-indices for supply and demand are developed. Then, the sub-indices are aggregated into mobility supply and demand indices. Finally, a composite mobility index is proposed by aggregating the supply and demand indices which contain the information on all influential indicators.

As mentioned in Sect. 2, there are 12 indicators for representing the supply dimension of mobility. The number of factors is obtained based on the eigenvalues with values greater than 1.0, using the scree test. The scree test uses the graph of eigenvalues to either find the number of factors with eigenvalue more than 1, or above the break point (if applicable) [23]. The scree test indicates that there are 4 factors for mobility supply indicators. We also apply factor analysis on 13 indicators for the demand dimension of mobility, as given in Sect. 2. The number of factors for mobility demand indicators is 4. The indicators that do not have any factor loadings above 0.3 (D_1 and D_7) are removed from the investigation as factor loadings less than 0.3 are not practically significant [23]. Figure 3 shows the result of the factor analysis for supply and demand indicators. We label the extracted factors based on their variable construct.



(a) Hierarchical structure for supply indicators

(b) Hierarchical structure for demand indicators

Fig. 3. Hierarchical structure for supply and demand indicators

6 City Mobility Index Construction

In this study, we use the first principal component to construct composite indices for city mobility comparison. Principal component analysis (PCA) has been used extensively for developing indices. The World Bank adopts the first principal component as the wealth index [26, 27]. Principal components are linear combinations of variables as

$$Z_l = \varnothing_{11}\mathrm{X}_1 + \varnothing_{21}\mathrm{X}_2 + \cdots + \varnothing_{\mathrm{Pl}}\mathrm{X}_{\mathrm{P}} \tag{8}$$

where Z_l is the lth principal component, X_p is the pth variable, $\O_{jl}$ are factor loadings, and the summation of their square values is equal to 1. The principal components are uncorrelated to each other. The first principal component has the largest variance, and other principal components have lower variances in order. The first principal component works as a weighted average so that weights are factor loadings [28]. It contains the most information of all variables and can be used as a representative of the variables. PCA is a way of presenting the data while pointing out the similarities and differences. In the following section, we generate a sub-index for each factor based on the first principal component. Cities are prioritised based the index. Then, the loadings of each sub-index are aggregated to develop indices for supply, demand and city mobility.

6.1 Supply Factor Sub-indices

The index for the private transport access (*PrTA*) factor is constructed based on indicators S_2, S_3, S_6, S_7 and S_8. The variables S_1, S_9, S_{10} and S_{11} are used for the public transport access (*PuTA*) index. The index of Urban road (*UR*) is equal to S_4 and the urban motorway (*UM*) index is a combination S_5 and S_{12}. The formulae for supply factor indices are given in (9)–(12). The top-10 cities based on each supply factor index are shown in Table 3.

$$PrTA\,\text{index} = 0.575S_2 + 0.025S_3 + 0.051S_6 + 0.816S_7 + 0.001S_8 \tag{9}$$

$$PuTA\,\text{index} = -0.001S_1 + 0.061S_9 + 0.993S_{10} + 0.104S_{11} \tag{10}$$

$$UR\,\text{index} = S_4 \tag{11}$$

$$UM\,\text{index} = -0.375S_5 + 0.927S_{12} \tag{12}$$

Table 3. City rankings based on supply factor indices

Rank	Supply sub-indices			
	PrTA	*PuTA*	*UR*	*UM*
1	Illinois	Moscow	Strasbourg	Zurich
2	Gothenburg	Hong Kong	Ankara	Tokyo
3	Kocaeli	Prague	Athens	Moscow
4	Phoenix	Barcelona	Rome	Oslo
5	Melbourne	London	Tokyo	Brisbane
6	Montreal	Taipei	Munich	Stockholm
7	Dubai	Madrid	Montreal	Paris
8	Abu Dhabi	Stockholm	Lisbon	Illinois
9	Glasgow	Copenhagen	Helsinki	Sydney
10	Brisbane	Tokyo	Hamburg	London

6.2 Demand Factor Sub-indices

The index for the public transport demand (*PTD*) factor is constructed based on indicators D_3, D_4, D_5, D_6 and D_{13}. The indicators D_8, D_9 and D_{12} are used in the private trip (*PT*) index. The travel demand (*TD*) index is a combination of D_2 and D_{10} and the index of public transport trip (*PTT*) is equal to D_{11}. The formulae of supply factor indices are given in (13)–(16). The top-10 cities based on each demand factor index are shown in Table 4.

$$PTD\,\text{index} = 0.004D_3 + 0.994D_4 + 0.013D_5 + 0.109D_6 + 0.001D_{13} \tag{13}$$

$$PT\,\text{index} = -0.055D_8 - 0.041D_9 + 0.998D_{12} \tag{14}$$

$$TD\,\text{index} = 0.999D_2 + 0.001D_{10} \tag{15}$$

$$PTT\,\text{index} = D_{11} \tag{16}$$

Table 4. City rankings based on demand factor indices

Rank	Demand sub-indices			
	PTD	*PT*	*TD*	*PTT*
1	Hong Kong	Phoenix	Illinois	Addis Ababa
2	Taipei	Prague	Dubai	Copenhagen
3	Moscow	Jerusalem	Phoenix	Tokyo
4	Barcelona	Portland	Gothenburg	Delhi
5	Beijing	Turin	Portland	Johannesburg
6	Milan	Sydney	Abu Dhabi	Nairobi
7	London	Strasbourg	Sydney	Mumbai
8	Singapore	Melbourne	Hamburg	Ankara
9	Madrid	Brussel	Brisbane	Tshwane
10	Prague	Zurich	Rome	Berlin

6.3 Mobility Supply and Demand Indices

The first components derived from four supply and demand factor indices are used as the mobility supply and demand indices as formulated in (17) and (18) respectively.

$$\mathit{Mobility\,supply\,index} = 0.328 \times TA + 0.945 \times PTA - 0.002 \times UR + 0.001 \times UM \tag{17}$$

$$\mathit{Mobility\,demand\,index} = 0.986 \times PTD + 0.001 \times PT - 0.168 \times TD + 0.001 \times PTT \tag{18}$$

To find an overall index for city mobility, the supply and demand indices are aggregated as (19). The top-10 cities based on these mobility indices, including supply, demand and overall, are shown in Table 5.

$$\mathit{Mobility\,index} = 0.511 \times \mathit{Supply\,index} + 0.860 \times \mathit{Demand\,index} \tag{19}$$

Cities have different rankings based on their mobility indices. The first principal component meaningfully aggregates the information of sub-indices to find out which cities supply better transport infrastructures and facilities and which cities have a higher demand for their supplied facilities. Finally, the high-ranked cities based on the composite mobility index are the cities that have high supply and demand for their transport system. The constructed mobility indices represent city mobility knowledge discovered from mining the MCD in relation to a city's supply and demand in mobility.

Table 5. City rankings based on mobility indices

Rank	Mobility indices		
	Mobility supply	Mobility demand	Mobility
1	Moscow	Hong Kong	Hong Kong
2	Hong Kong	Taipei	Moscow
3	Prague	Moscow	Taipei
4	Gothenburg	Barcelona	Barcelona
5	Kocaeli	Beijing	London
6	Barcelona	Milan	Prague
7	London	London	Singapore
8	Stockholm	Singapore	Milan
9	Singapore	Prague	Madrid
10	Zurich	Madrid	Tokyo

7 Conclusions

Many datasets with global data such as the MCD for city mobility have missing values which prevent them from being used for discovering useful knowledge. To overcome this data mining limitation, an effective method is required to impute missing values. In this paper, we have proposed an effective hybrid missing data imputation method by integrating the advantages of decision trees and fuzzy clustering into an iterative algorithm. We have also conducted extensive experiments to demonstrate its performance over other existing imputation methods on the MCD and on three commonly used datasets. The performance of the proposed hybrid imputation method enables us to fully use all the relevant mobility indicators in the MCD for examining the mobility of 63 cities around the world. To gain insightful city mobility knowledge from mining the MCD, we have initiated the concept of mobility supply and demand, with which a hierarchical structure of mobility supply and demand indicators have been constructed using factor analysis. This hierarchical structure has enabled us to construct city mobility indices at different levels using principal component analysis, thus providing useful insights into city mobility comparisons under different contexts. This study makes a methodological contribution to missing data imputation research as well as a conceptual and practical contribution to city mobility research.

References

1. Nikfalazar, S., Amiri, M., Khorshidi, H.A.: Social impact assessment on metro development with a case study in Eastern District of Tehran. Int. J. Soc. Syst. Sci. **6**(3), 245–263 (2014)
2. Rassafi, A.A., Vaziri, M.: Sustainable transport indicators: definition and integration. Int. J. Environ. Sci. Technol. **2**(1), 83–96 (2005)
3. Violato, R.R., Galves, M.L., de Oliveira, D.D.G.: Non-motorized mobility in central urban areas: application of multi-criteria decision aid in the city of campinas, Brazil. Int. J. Sustain. Transp. **8**(6), 423–446 (2014)
4. Haghshenas, H., Vaziri, M.: Urban sustainable transportation indicators for global comparison. Ecol. Ind. **15**(1), 115–121 (2012)
5. Moeinaddini, M., Asadi-Shekari, Z., Zaly Shah, M.: An urban mobility index for evaluating and reducing private motorized trips. Measurement **63**, 30–40 (2015)
6. Albalate, D., Bel, G.: What shapes local public transportation in Europe? Economics, mobility, institutions, and geography. Transp. Res. Part E Logist. Transp. Rev. **46**(5), 775–790 (2010)
7. Albalate, D., Bel, G.: Tourism and urban public transport: holding demand pressure under supply constraints. Tour. Manag. **31**(3), 425–433 (2010)
8. Alonso, A., Monzón, A., Cascajo, R.: Comparative analysis of passenger transport sustainability in European cities. Ecol. Ind. **48**, 578–592 (2015)
9. Reisi, M., Aye, L., Rajabifard, A., Ngo, T.: Land-use planning: implications for transport sustainability. Land Use Policy **50**, 252–261 (2016)
10. Joumard, R., Gudmundsson, H., Folkeson, L.: Framework for assessing indicators of environmental impacts in the transport sector. Transp. Res. Rec. **2242**, 55–63 (2011)
11. UITP: Mobility in cities database. International Association of Public Transport, Brussels (2015)
12. Nikfalazar, S., Yeh, C.-H., Bedingfield, S., Khorshidi, H.A.: A new iterative fuzzy clustering algorithm for multiple imputation of missing data. In: IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 1–6. IEEE, Naples (2017)
13. Rahman, M.G., Islam, M.Z.: Missing value imputation using a fuzzy clustering-based EM approach. Knowl. Inf. Syst. **46**(2), 389–422 (2016)

14. James, G., Witten, D., Hastie, T., Tibshirani, R.: An Introduction to Statistical Learning: with Applications in R. Springer, New York (2013). https://doi.org/10.1007/978-1-4614-7138-7
15. Sato-Ilic, M., Jain, L.C.: Innovations in Fuzzy Clustering: Theory and Applications. Springer, Heidelberg (2006). https://doi.org/10.1007/3-540-34357-1
16. Campello, R.J.G.B., Hruschka, E.R.: A fuzzy extension of the silhouette width criterion for cluster analysis. Fuzzy Sets Syst. **157**(21), 2858–2875 (2006)
17. Junninen, H., Niska, H., Tuppurainen, K., Ruuskanen, J., Kolehmainen, M.: Methods for imputation of missing values in air quality data sets. Atmos. Environ. **38**(18), 2895–2907 (2004)
18. Cevallos Valdiviezo, H., Van Aelst, S.: Tree-based prediction on incomplete data using imputation or surrogate decisions. Inf. Sci. **311**, 163–181 (2015)
19. Rahman, M.G., Islam, M.Z.: Missing value imputation using decision trees and decision forests by splitting and merging records: two novel techniques. Knowl. Based Syst. **53**, 51–65 (2013)
20. Cheng, K.O., Law, N.F., Siu, W.C.: Iterative bicluster-based least square framework for estimation of missing values in microarray gene expression data. Pattern Recogn. **45**(4), 1281–1289 (2012)
21. Wang, X., Li, A., Jiang, Z., Feng, H.: Missing value estimation for DNA microarray gene expression data by support vector regression imputation and orthogonal coding scheme. BMC Bioinform. **7**(32), 1–10 (2006)
22. Schneider, T.: Analysis of incomplete climate data: estimation of mean values and covariance matrices and imputation of missing values. J. Clim. **14**, 853–871 (2001)
23. Hair, J.F., Black, W.C., Babin, B.J., Anderson, R.E.: Multivariate Data Analysis, 7th edn. Pearson Prentice Hall, Upper Saddle River (2014)
24. Tate, E.: Social vulnerability indices: a comparative assessment using uncertainty and sensitivity analysis. Nat. Hazards **63**(2), 325–347 (2012)
25. Reckien, D.: What is in an index? Construction method, data metric, and weighting scheme determine the outcome of composite social vulnerability indices in New York City. Reg. Environ. Change **18**(5), 1439–1451 (2018)
26. Eyler, L., Hubbard, A., Juillard, C.: Assessment of economic status in trauma registries: a new algorithm for generating population-specific clustering-based models of economic status for time-constrained low-resource settings. Int. J. Med. Inf. **94**, 49–58 (2016)
27. Tajik, P., Majdzadeh, R.: Constructing pragmatic socioeconomic status assessment tools to address health equality challenges. Int. J. Prev. Med. **5**(1), 46–51 (2014)
28. Vidal, R., Ma, Y., Sastry, S.S.: Generalized Principal Component Analysis. Interdisciplinary Applied Mathematics. Springer, New York (2016). https://doi.org/10.1007/978-0-387-87811-9

A Novel Learning-to-Rank Method for Automated Camera Movement Control in E-Sports Spectating

Hendi Lie[1], Darren Lukas[1](✉), Jonathan Liebig[2], and Richi Nayak[1]

[1] Queensland University of Technology, Brisbane, Australia
{h2.lie, lukasd, r.nayak}@qut.edu.au
[2] Liebig Productions, Aachen, Germany
jj@liebig.gg

Abstract. The popularity of modern competitive gaming (or e-sports) has skyrocketed in the past decade. A key part of e-sports is the spectating experience where fans watch tournament games through a camera of the observer. Bigger tournaments hire professional human observers with high-end tools to monitor important events in the game map for broadcasting the game. This setup is prone to errors. It results in missing important events within the game and lowers the spectating experience overall. It is also not sustainable in long-term and not affordable for the small-scale tournaments. This paper proposes a novel method of automated camera movement control using the AdaRank learning-to-rank algorithm to find and predict important events so the camera can be focused on time. The *Dota 2* game setup and its replay data are used in extensive experimental testing. The proposed method has shown to outperform the accuracy of both a past machine learning approach and a professional team of human observers.

Keywords: E-sports · Learning-to-rank · Automated camera control · *Dota 2*

1 Introduction

Competitive gaming also called as e-sports, is a fast-growing industry with almost a billion dollar in revenue that attracts millions of dedicated players and fans world-wide [1–3]. Many tournaments covering different games, ranging from small online competitions to large international ones, are organised every day. The increasing interest in e-sports and exponential growth in fan-base have resulted in some of these tournaments to offer a gigantic prize pool. For example, Dota 2's International offers a total prize pool of US$24.6 million, exceeding the financial prestige of many traditional sports events such as The Masters in golf [1].

With more fans following the games of their favourite e-sports teams, broadcasting of competitive matches has become a key business of the industry. Large studios bid for tournament broadcasting rights and employ many casters (i.e., the game commentators) and observers (i.e., the camera operators tasked for observing and capturing important events in the game). In comparison to traditional sports like football where important events tend to occur around a single focal point (e.g. the ball), observing an

R. Islam et al. (Eds.): AusDM 2018, CCIS 996, pp. 149–160, 2019.
https://doi.org/10.1007/978-981-13-6661-1_12

e-sports game is more challenging as multiple important events can occur simultaneously on different parts of the game [4]. To solve this problem, studios commonly invest in multiple dedicated observers, large screens, and the state-of-the-art visual and recording tools for the large tournaments. However, this setup is not affordable for smaller, online tournaments is not sustainable in long-term. Moreover, it is prone to missing important events, thus limiting the spectating experience for many fans.

As an alternative to human observers, automated methods have been proposed. Past approaches include using a machine learning model to predict the occurrence of important events, then apply a heuristic camera controller to focus on certain entities in the game [5]. These approaches have been reported to be inaccurate, missing many important events and reducing overall spectating experience [5, 6].

To provide a more accurate and affordable camera operation, this paper proposes a novel method based on learning-to-rank. Instead of predicting the occurrence of important events explicitly, we propose to rank entities based on their respective importance values. An entity's importance is calculated as the sum of related events values. We train an AdaRank [7] model to produce this ranking ahead of time and use a simple camera controller to focus on the top entities. To the best of our knowledge, this is the first work where the camera control in e-sports is predicted using a learning-to-rank model.

The proposed method is evaluated using a past matches replay data of a popular game, *Defense of the Ancients 2 (Dota 2)* [8]. We identified entities to be ranked as player-controlled *heroes* and produced a dataset annotated with important events retrieved from the game. Empirical analysis reveals that the proposed method outperforms a past automated approach and a professional team of human observers in accurately predicting important events and focusing the camera on top important entities ahead of time. This approach is a step forward in providing quality spectating experience for an audience of this growing industry.

2 Background and Related Work

E-sports is a term commonly used to refer to competitive video gaming coordinated by different leagues, ladders, and tournaments and watched by fans over internet broadcasting platforms [3]. There are many popular titles/games in e-sports, such as Dota 2, League of Legends, Fortnite, Counter Strike and Overwatch, bringing diverse games play and significant audience. Computer games and e-sports have been a subject of past machine learning research, focusing efforts in building game platforms [9], predicting winners of games [10] and building intelligent bots [11, 12].

While significant efforts have been devoted to building game platforms, and training players and bots in making intelligent moves, the research problem of spectating experience has not received much attention. Due to the overall increased interest in e-sports games, the exponential growth of fan base and wide availability of high-speed Internet access for video streaming [3], this problem has become increasingly important. Application of automated camera system is not new to traditional sports. For instance, Pixellot [13] is an automated camera system used in the broadcasting of a diverse range of sports including football, basketball, and rugby. It utilises multiple sensors and cameras which are connected to an autonomous server that controls the point of focus.

Specific to e-sports, many games have a built-in automated camera in their spectating system. For example, in *Dota 2* and *League of Legends*, there is a camera system available in spectator mode called directed camera [14, 15]. Although no official documentation regarding details of their approach can be found, we believe they implement a mixture of heuristics and machine learning algorithms to follow events in a game automatically. These built-in systems, however, are known to regularly miss important events and produce unsatisfactory spectating experience [6].

The following work comes closest to the proposed method. A K-Nearest Neighbour (KNN) model was used to perform event classification on game entities [5]. The predicted events are then used as an input to a heuristic-based camera controller. At a given timeframe, each game entity is labelled with up to 6 different game events. The dataset included the entity features computed based on movements, combat activities and damage received.

Based on visual empirical analysis with human viewers, the resulting camera movements in [5] were found to miss important events and jumped too much between entities in the game. Many viewers thought the default directed camera performs better. The model has shown overfitting and poor performance on test data as it was trained on the data of a single match only. In addition, the camera movement is highly dependent on a complex, game specific heuristic and not extendable to any other e-sports.

To overcome problems with past approaches, this paper offers a different perspective. Instead of explicitly finding and focusing the camera on important events, we focus on important entities involved in these events. We hypothesize that predicting the actual importance of entities is not important, instead, we use a learning-to-rank model to produce optimal ordering. The model is trained on large amount of past replay data to prevent overfitting. To ensure this method is extendable to other e-sports games, we implement a simple heuristic that rely on the top-ranked entities output by the model. This approach results in a camera that exceed accuracy of human observers and still produce smooth spectating experience.

3 Proposed Method

To enable accurate and smooth automated camera movements, we propose a method based on a learning-to-rank model. The proposed method (shown in Fig. 1) contains 3 components: (1) a game data parser; (2) an Ada-rank model, and (3) a camera controller.

In maximising spectating experience, the camera needs to show important events and related entities as they take place. We conjecture that the model should be trained to rank entities based on future importance. We define a parameter δ, time duration when future importance values of entities are computed. For each e-sport match, starting at timestamp $t = 0$, we first apply the game data parser to extract game logs, perform data pre-processing and produce features based on condition and interactions of entities at t. During the training process, we label each entity in the dataset with importance value based on events taking place between t and $t + \delta$. This allows the model to predict entity importance ahead of time. The most important entities output by the model is then followed by the camera controller until $t + \delta$ and this process is repeated until the match ends. Different values of δ were tested to find the optimal value.

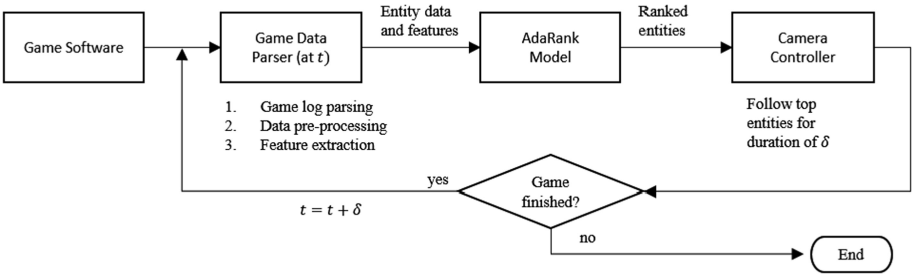


Fig. 1. Components in the proposed method.

3.1 The Learning-to-Rank Model

In observing an e-sports match, the camera is expected to follow entities involved in important events, such as scoring, killing opposition team or achieving a specific goal. Assuming importance values of all events can be correctly quantified, the importance of an entity can be determined by the sum of all events' importance the entity has been involved in. In predicting these important entities, we conjecture that the actual importance value is not significant, instead, we need to optimise the ordering of the top entities. Therefore, we model the problem of finding entity importance as the entity ranking problem and propose to solve it using a learning-to-rank model.

Learning-to-rank is a class of machine learning techniques to solve the ranking problem [16]. In Information Retrieval, it is commonly used to order a set of documents $\boldsymbol{D}$ optimally based on a query q by utilizing a ranking model $f(q, d), d \in \boldsymbol{D}$ [17]. Learning-to-rank is considered as supervised learning which requires document-query pairs labelled with relevance judgement. There are two types of data labelling in learning-to-rank: absolute judgment and relative judgement. Absolute judgment compares the document directly to the query independently to other documents, while in relative judgement, all relevant documents are ranked together based on preference [18]. We employed relative judgement in this work because the relevance of a game entity is relative to other entities.

There are three major approaches to learning-to-rank: (1) pointwise which aims to assign each document-query pair with a relevancy score; (2) pairwise that investigates ranking as a comparison over pairs of documents; and (3) listwise that tries to produce optimal ranking in the entire list of documents. We choose listwise approach as it has been shown to generally outperform the other methods [16] and it fits to the ranking problem of entities in an online game more naturally.

To formally define the problem, let $\boldsymbol{T}$ be the set of all timestamps sampled every time interval δ from an e-sport match. For each $t \in \boldsymbol{T}$, let $\boldsymbol{E_t} = \{e_{t,1}, e_{t,2}, \ldots, e_{t,n}\}$ be set of n game entities to be ranked, $\boldsymbol{Y_t} = \left\{y_{t,1}, y_{t,2}, \ldots, y_{t,n}\right\}$ be the set of label importance values and $y_{t,i} \in \boldsymbol{Y_t}$ be the importance label of $e_{t,i} \in \boldsymbol{E_t}$. Depending on the context of the game (e.g. games where entities could be summoned or killed), the value of n could increase or decrease over different ts. In computing $y_{t,i}$, we consider all important events $\boldsymbol{v}$ occurring between t to $t + \delta$ that $e_{t,i}$ is involved in. This way of labeling allows the model to predict important entities before they are involved in future important events.

Suppose a feature vector $X_t = \left\{x_{t,1}, x_{t,2}, \ldots, x_{t,n}\right\}$ is created at every t using feature function ϕ, with $x_{t,i} = \phi(t, e_{t,i})$. We present a training example as $\boldsymbol{S_t} = \left\{\left(x_{t,1}, y_{t,1}\right), \left(x_{t,2}, y_{t,2}\right), \ldots, \left(x_{t,n}, y_{t,n}\right)\right\}$. The proposed listwise learning-to-rank model learns a ranking model $f(\boldsymbol{x_{t,i}})$ such that $\boldsymbol{E_t}$ is sorted by $\boldsymbol{Y_t}$ in descending order.

Objective Function. To produce an accurate entity ranking, we employed Normalised Discounted Cumulative Gain (NDCG) as the listwise objective to optimise. NDCG uses a graded relevance and discount function which allows a listwise algorithm to prioritise top-ranked entities in the optimisation process [19].

Let f be a ranking model trained on a dataset $\boldsymbol{S_t}$ as described before. Suppose $|\boldsymbol{E_t}|$ be the list of entities sorted according to their respective $\boldsymbol{Y_t}$. NDCG of f on $\boldsymbol{S_t}$ for top $k(k \leq n)$ entities is computed as follows [12]:

$$NDCG_k(f, \boldsymbol{S_t}) = \frac{DCG_k(f, \boldsymbol{S_t})}{IDCG_k(\boldsymbol{S_t})}$$

Where DCG (Discounted Cumulative Gain) can be computed by:

$$DCG(f, \boldsymbol{S_t})_k = \sum_{i=1}^{k} \frac{2^{y_{t,i}} - 1}{log_2(i+1)}$$

And IDCG (Ideal Discounted Cumulative Gain):

$$IDCG_k = \sum_{i=1}^{|\boldsymbol{E_t}|} \frac{2^{y_{t,i}} - 1}{log_2(i+1)}$$

AdaRank. We implemented AdaRank [7] as the learning-to-rank model. Compared to other learning-to-rank algorithms, AdaRank can optimise any query-based performance measures with values in the range of [−1, +1] efficiently, by fitting the chosen NDCG optimisation function [7]. It also provides a practical framework for ranking which assumes that the document pairs for the same query are independently distributed. AdaRank focuses on training the top of document list and it treats queries, not document pairs. Thus, the algorithm is not biased toward queries with a higher number of document pairs, which is important in games where a number of entities n could change as the game progresses (such as first-person shooting games, where entities/players could die and get eliminated from the game).

3.2 Camera Controller

The proposed camera controller component follows a simple heuristic and takes the top-2 ranked entities as input. As per the heuristic, the top-ranked entity should always be on camera between two consecutive time frames, t and $t + \delta$. For the second top ranked entity, the camera controller follows how close it is to the first entity. If the two entities are within the reach of camera frame, the controller will focus on the midpoint of these two entities. An illustration of this heuristic is presented in Fig. 2.

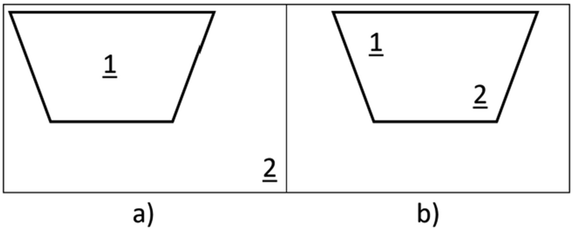


Fig. 2. A heuristic algorithm for setting camera focus given top-ranked entity (1) and second-ranked entity (2). At (a), the distance between (1) and (2) is too large, therefore the camera only focuses on (1). At (b), (1) and (2) are close, thus the camera captures both.

4 Empirical Analysis

4.1 Selection of E-Sports Game

In this paper, we train the learning-to-rank model and build the camera controller based on *Dota 2 (Defense of the Ancients 2)*. *Dota 2* is a free to play MOBA (Multiplayer Online Battle Arena), released in 2013 by Valve Corporation. It is one of the most popular games in the world with a large and mature e-sports scene [1, 8]. This game is chosen due to its wide popularity as well the authors' sound knowledge of the game and its mechanics.

A *Dota 2* match has 10 players divided into 2 teams, the *Radiant* (green) and the *Dire* (red), each with bases located at different ends of the map. At the start of a match, each player chooses to play a *hero*. There are 115 heroes available to play, each with different abilities, attributes, and roles in the team. Each player can only choose one hero at the start of the game. The game is played on a large map with buildings (*towers*, *barracks*), *creeps/minions*, neutral *creeps* and the largest neutral creep called *Roshan*. The objective of the game is to destroy the opposing team's largest building called *Ancient* located in their base while keeping your team's *Ancient* alive. A *Dota 2* match usually lasts between 30–50 min.

Like traditional sport, fans can *spectate* live matches in *Dota 2* game client. At any given time, only a small portion of the map is visible to the camera, as illustrated in Fig. 3. Everyone can control their own camera and focus on different parts of the map through mouse movement. However, with the vast size of the Dota 2 game map and many events occurring simultaneously at the same time, it is difficult to watch relevant events. In a common tournament spectating setting, fans follow the movement of the tournament official observer camera.

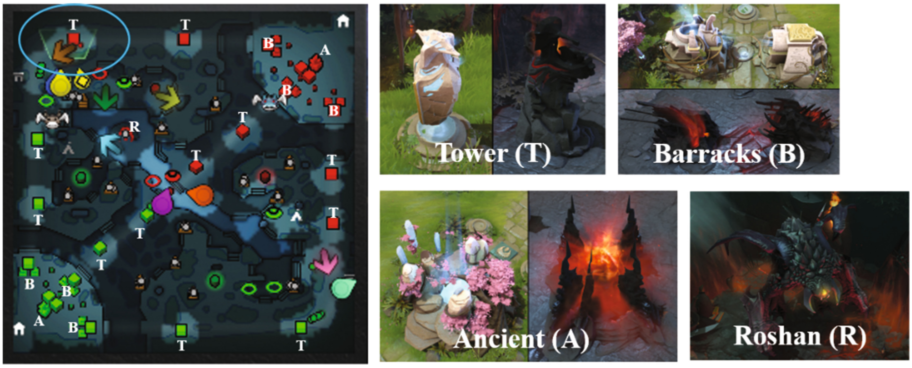


Fig. 3. Dota *2* game map, showing bases of Radiant (green) and Dire (red) teams. The markers on the map represent towers (T), barracks (B), ancients (A), Roshan (R), arrows (5 Dire heroes) and teardrops (5 Radiant heroes). The green trapezium tagged by the blue oval illustrates a portion of the map available to see for a spectating camera (Color figure online).

4.2 Dataset

The dataset is constructed from the game logs of *Dota 2* matches taken place between October-November 2017 in *Dreamleague*. To acquire the relevant features, we downloaded replay files through Valve's API and OpenDota API [9]. Using these files, we extracted the following three main log files using the Clarity file parser [20]:

1. **Combat logs:** a detailed damage/heal information for every entity in the game including tick (server time), inflictor, receiver, the value of damage/heal, health value before and after the event.
2. **Life state logs:** a death/spawn information for heroes, buildings, and Roshan.
3. **Property change logs:** a history of property (e.g. health point, mana point, level, location and net worth) of heroes throughout the game.

We process log entries from combat and property change logs to generate features for each player's hero. Over every time interval δ, we produced critical features for each hero including health, mana, alive status, modifier status, distance from other important entities and damage/heal received and given. A total of 1502 numerical and binary features were generated. *Dota 2* server could produce log entries every tick (1 tick = 1/30 s), thus time interval value δ onward will be in tick unit.

We used these three logs to identify important events and assign a value/weight to each event and subsequently to each hero. For instance, the kill events are labelled by utilising hero spawn/death recorded in life state log and hero to *Roshan* damage are identified through significant damage Roshan received from that hero in the combat log. There are many events considered as important in a *Dota 2* match, including hero kills/team fights, tower, and barrack destructions, and Roshan kills (Table 1).

To quantify the importance of different events, we constructed an arbitrary importance structure in Table 2. The value of each type of event is set based on impact these events make to the game. For example, killing a hero eliminates it temporarily from the game and weakens the opponent team, while destroying a tower or barracks allow you to advance further to the *Ancient*. The significant combat damages indicate an attempt to fight and should be shown to the spectators. Different e-sports will require different events importance structure, but it is easily extendable.

Table 1. Dataset statistics

Statistics	Training + validation	Test
Matches	14	7
Avg. game length (mins:secs)	46:15	46:25
Dimension (rows × features)	1,000,000 × 1502	366,700 × 1502
Size (GB)	7.8	4.51

Lastly, the dataset was transformed to Microsoft LETOR (**Learning-to-rank**) format. Each row contains a tick as ID, hero features, and relevance labelling. As there are maximum 10 heroes in any given time t, we assigned a degree of relevance of 10 to hero $e_{t,i}$ with highest $y_{t,i}$, 9 to the next highest, subsequently until 1 degree of relevance the lowest. If there is more than one hero with same y_t, they are assigned same degree of relevance. Table 3 shows a sample transformed data of 10 heroes in a tick.

Table 2. Event importance hierarchy for *Dota 2*

Event type	Event name	Value
Death	Roshan	75.0
	Hero	60.0
	Tower/barracks	60.0
	Neutral	3.0
	Creep	0.5
Damage	Hero to tower/barracks	4.0
	Hero to hero	2.0
	Hero to Roshan	2.0
	Roshan to hero	2.0
	Tower to hero	1.5
	Hero to neutral	1.2
	Creep to hero	0.5
	Hero to creep	0.4
Other	Smoke usage	5.0

Table 3. A sample LETOR formatted row

Hero	Importance	Query ID	F#1: Health	F#2: Max health		F#1501: Stunned	F#1502: Disabled
#0	1.0	qid:52190	1:800.0	2:800.0	…	1501:0.0	1502:0.0
#1	10.0	qid:52190	1:1061.0	2:1240.0	…	1501:0.0	1502:0.0
#2	9.0	qid:52190	1:276.0	2:1080.0	…	1501:0.0	1502:0.0
#3	8.0	qid:52190	1:1326.0	2:1340.0	…	1501:0.0	1502:0.0
#4	7.0	qid:52190	1:1001.0	2:1540.0	…	1501:0.0	1502:0.0
#5	1.0	qid:52190	1:880.0	2:880.0	…	1501:0.0	1502:0.0
#6	1.0	qid:52190	1:1415.0	2:1620.0	…	1501:0.0	1502:0.0
#7	1.0	qid:52190	1:1180.0	2:1180.0	…	1501:0.0	1502:0.0
#8	6.0	qid:52190	1:454.0	2:1080.0	…	1501:1.0	1502:0.0
#9	5.0	qid:52190	1:991.0	2:1100.0	…	1501:0.0	1502:0.0

4.3 Implementation

We used the AdaRank implementation from RankLib library [21]. All other components from the data parser to camera controller are implemented in Python 3.6 using standard libraries (e.g. numpy, pandas and scikit-learn) running on Windows OS.

4.4 Evaluation

In this work, we performed the two-stage empirical analysis. Firstly, we test different values of time interval δ. Lower δ allows the camera to change entities ranking more often and be more reactive to catch important events. However, very frequent camera movements could result in dizziness and unpleasant spectating experience, hence finding optimal δ is important.

Table 4. Hyperparameters settings

Hyperparameter	Values	Details
Time interval δ	{15, 30, 60 and 120 ticks} (0.5, 1.0, 2.0 and 4.0 s respectively)	Lower δ result in more responsive camera, at risk of dizziness and unpleasant spectating experience
For each δ		
Z-score Normalisation	{yes, no}	Features in the dataset are on different scale and variance
Tolerance	{0.001, 0.002, 0.004}	Tolerance between two consecutive rounds of learning in AdaRank. Higher tolerance should result in less epoch/training iteration, preventing overfitting at risk of underfitting

For each δ tested, we performed a grid search on a set of hyperparameters (detailed on Table 4) and evaluate each combination using NDCG@2 on 5-fold cross validation. The optimal set is used to produce the best performing model for that specific δ, then this model is evaluated on the test dataset. NDCG@2 is chosen as the implemented camera controller for *Dota 2* takes top 2 entities as input.

Next, to ensure the model and camera controller works well in capturing important events, we benchmarked the accuracy of the proposed method against (1) human observers and (2) KNN-based event classification method [5]. For human observers, we captured the *Dreamleague* 2017 observer team's camera movement data from the replay files. For the KNN-based method, as described by the author, we used K = 5 to yield the best result.

5 Results

5.1 NDCG

Table 5 summarises the NDCG@2 evaluation values for various experiment settings. Our results show a gradual drop in NDCG@2 values from the best 15 ticks model to the 60 ticks model, followed by a significant drop for the 120 ticks model. Low δ value (15 ticks) allows the model to only predict the occurrence of events in short immediate future, resulting in the better ranking model. However, based on this result, using slightly higher δ (at 60 ticks) does not significantly reduce the performance and is preferable to produce reduce dizziness and produce a more pleasant spectating experience.

Table 5. NDCG@2 evaluation of best hyperparameter set for each δ

Time interval δ	Dataset	
	5-fold cross training and validation	Test
15 ticks (0.5 s)	0.7147	0.7222
30 ticks (1.0 s)	0.7095	0.7099
60 ticks (2.0 s)	0.6911	0.6892
120 ticks (4.0 s)	0.6354	0.6295

Table 6. Accuracy testing result on test data

Time interval δ	Total miss; % miss		
	Our method	KNN-based	Human observers
15 ticks (0.5 s)	118; 24.1%	179; 36.5%	85; 17.3%
30 ticks (1.0 s)	71; 14.5%	191; 38.9%	
60 ticks (2.0 s)	78; 15.9%	171; 34.8%	
120 ticks (4.0 s)	100; 20.4%	164; 33.4%	
Total number of events			490

5.2 Accuracy

Table 6 shows the accuracy score evaluation of the proposed learning-to-rank model, KNN-based classifier (K = 5) and professional human observers. Overall, the proposed method performs at a similar accuracy with human observers, with the 30-ticks and 60-ticks models even outperforming them. By following the top-2 ranked entities output by the models, the camera can focus on important events as they occur in the game.

The 15-ticks model missed more events than all other learning-to-rank models. With such small δ, the model could only capture limited number of events per time interval, resulting in sparse importance values and high variance error. Setting higher δ value could allow the model to capture more events in per time interval and generalise better.

The KNN-based classifiers do not perform well and are consistently less accurate than both learning-to-rank models and human observers. Other than accuracy, during the experiments, we found that the KNN approach is significantly slower in making predictions. As it needs to store all training examples, KNN produces a very memory consumptive model despite having significantly less features. Our method is faster, lighter and more suitable for producing predictions in real-time situation.

6 Discussion and Future Work

We have presented a novel method of the automated camera movement control for e-sports spectating. We proposed an innovative approach based on learning-to-rank, specifically using an AdaRank model trained on replay data, complemented by a simple camera controller heuristic. This method was tested using the *Dota 2* replay game data. We explored the effect of different time intervals δ and found that 60 ticks model produce good result while still maintaining reasonable reactiveness. The whole system (model + camera controller) was compared against a past machine learning approach and human observers and was found to be more accurate.

In future, we would like to expand the work. Firstly, since the *Dota 2* camera is embedded inside the game client software without documentation or API, we could only follow entities by mimicking keyboard presses to focus on certain heroes. With a free camera control which could move to any coordinate in the game, the camera can capture more non-hero related events, such as tower destroyed by creeps/illusions, resulting in a more extensive spectating experience.

Lastly, the produced automated camera system should be viewed from the spectators' perspective and therefore, it is advised to conduct surveys on the camera system to evaluate the spectating satisfaction. In addition, as our camera controller is implemented with *Dota 2* in perspective, we would like to expand the algorithm and test our method on games from a different genre, such as first-person shooters. Another possible expansion is to incorporate external features such as player's popularity or spectator's preference in prioritising camera focus.

References

1. Khan, I.: Dota 2's The International 7 breaks esports prize pool record. http://www.espn.com.au/esports/story/_/id/19861533/dota-2-international-7-breaks-esports-prize-pool-record. Accessed 15 Aug 2018
2. Perez, M.: Report: Esports to Grow Substantially and Near Billion-Dollar Revenues in 2018. https://www.forbes.com/sites/mattperez/2018/02/21/report-esports-to-grow-substantially-and-near-a-billion-dollar-revenues-in-2018. Accessed 15 Aug 2018
3. Hamari, J., Sjöblom, M.: What is eSports and why do people watch it? Internet Res. **27**, 211–232 (2017)
4. Cook, M., Summerville, A., Colton, S.: Off The Beaten Lane: AI Challenges In MOBAs Beyond Player Control. arXiv Artificial Intelligence (cs.AI) (2017)
5. Phang, D.W.: Intelligent Camera Control in Game Replays (2014)
6. Did you know that the directed camera is broken?: DotA2. https://www.reddit.com/r/DotA2/comments/3yt2yc/did_you_know_that_the_directed_camera_is_broken/. Accessed 15 Aug 2018
7. Xu, J., Li, H.: AdaRank: a boosting algorithm for information retrieval. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 391–398. ACM Press, New York (2007)
8. Valve Inc.: Dota 2 (2018). http://blog.dota2.com/
9. OpenDota: OpenDota API. https://docs.opendota.com/. Accessed 15 Aug 2018
10. Semenov, A., Romov, P., Neklyudov, K., Yashkov, D., Kireev, D.: Applications of machine learning in Dota 2: Literature review and practical knowledge sharing. In: CEUR Workshop Proceedings, pp. 1–5 (2016)
11. OpenAI: OpenAI Five. https://blog.openai.com/openai-five/
12. Vinyals, O., Gaffney, S., Ewalds, T.: DeepMind and Blizzard open StarCraft II as an AI research environment. https://deepmind.com/blog/deepmind-and-blizzard-open-starcraft-ii-ai-research-environment/
13. Tamir, M., Oz, G., Ridnik, T.: Method and system for automatic television production (2017)
14. Fandom: Spectator Mode - League of Legends Wiki
15. Dota 2 Gamepedia: Spectating - Dota 2 Wiki
16. Liu, T.-Y.: Learning to rank for information retrieval. Found. Trends Inf. Retrieval **3**, 225–331 (2007)
17. Li, H.: A short introduction to learning to rank. IEICE Trans. Inf. Syst. **E94-D**, 1854–1862 (2011)
18. Niu, S., Guo, J., Lan, Y., Cheng, X.: Top-k learning to rank: labeling, ranking and evaluation. In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 751–760 (2012)
19. Wang, Y., Wang, L., Li, Y., He, D., Liu, T.-Y., Chen, W.: A theoretical analysis of NDCG type ranking measures. In: Conference on Learning Theory, pp. 1–26 (2013)
20. Schrodt, M.: Clarity - Comically fast Dota 2 and CSGO replay parser (2018). https://github.com/skadistats/clarity
21. Dang, V.: RankLib (2013). https://sourceforge.net/p/lemur/wiki/RankLib/

Homogeneous Feature Transfer and Heterogeneous Location Fine-Tuning for Cross-City Property Appraisal Framework

Yihan Guo[1](✉), Shan Lin[1], Xiao Ma[1], Jay Bal[1], and Chang-tsun Li[1,2]

[1] University of Warwick, Coventry, UK
{yihan.guo,shan.lin,x.ma,jay.bal,c-t.li}@warwick.ac.uk
[2] Charles Sturt University, Wagga Wagga, Australia
chli@csu.edu.au

Abstract. Most existing real estate appraisal methods focus on building accuracy and reliable models from a given dataset but pay little attention to the extensibility of their trained model. As different cities usually contain a different set of location features (district names, apartment names), most existing mass appraisal methods have to train a new model from scratch for different cities or regions. As a result, these approaches require massive data collection for each city and the total training time for a multi-city property appraisal system will be extremely long. Besides, some small cities may not have enough data for training a robust appraisal model. To overcome these limitations, we develop a novel **H**omogeneous **F**eature **T**ransfer and **H**eterogeneous **L**ocation **F**ine-tuning **(HFT+HLF)** cross-city property appraisal framework. By transferring partial neural network learning from a source city and fine-tuning on the small amount of location information of a target city, our semi-supervised model can achieve similar or even superior performance compared to a fully supervised Artificial neural network (ANN) method.

Keywords: Property valuation · Transfer Learning · Mass appraisal

1 Introduction

Real estate is always one of the pivotal parts of national economic development in many developing countries. Real estate is an enabler of business activity, that the growth of business activities requires consistent approaches to the valuation of real estate for accounting, banking activity, stock exchange listing and leverage lending purposes [23]. For these reasons, developing Automated Valuation Models(AVM) or Computer Assisted Mass Appraisal (CAMA) systems to support appraiser for more accurate property valuation is one crucial and recurrent research topics in both academia and industry. In the past three

R. Islam et al. (Eds.): AusDM 2018, CCIS 996, pp. 161–174, 2019.
https://doi.org/10.1007/978-981-13-6661-1_13

decades, many appraisal methodologies and applications have been proposed. They can be classified into two categories: traditional econometric approaches and machine learning approaches. The traditional approaches such as comparable method [28] and Multiple Regression Analysis (MRA)[4] usually rely on manual analysis of the characteristics of the properties. The recent machine learning based approaches cover a broad spectrum of machine learning techniques from k-Nearest Neighbors(k-NN)[9], Regression Decision Trees [13], Regression Random Forest [1], Artificial Neural Networks(ANN)[3], rough sets theory [10] and some hybrid approaches [22,25].

One significant challenge for residential real estate appraisal modelling is to account for the differences in location. Some recently proposed methods introduce geographic information system (GIS) features into their models [8,16,25]. However, most of these works only focus on analysing the geographic information within the same city but pay little attention to a cross-city scenario. When adding a new city into the AVM system, these approaches have to train a new model from scratch and require a massive data collection from the new city. Besides, the real estate markets in some low-tier cities are usually much smaller than those top-tier cities. Therefore, the amount of data collected from the small cities may not be sufficient enough to train a robust and reliable property appraisal model. If a robust model learned from one city can be transferred to other cities, the amount of time and resources required for training and data collection could be significantly reduced. Hence, in this paper, we proposed a novel **H**omogeneous **F**eature **T**ransfer and **H**eterogeneous **L**ocation **F**ine-tuning **(HFT+HLF)** cross-city property appraisal framework. As a case study of the Chinese real estate market, we collected the residential real estate resale data of six cites selected from three different city-tiers in China and evaluate the housing valuation performance of our methods after transfer learning. The contributions of our work are summarised below:

- Our work is one of the first research works focusing on cross-city transfer problem in property appraisal. Because the datasets collected from different cities contain entirely different sets of location features (district names, apartment names, etc.), most of the transfer learning methods cannot be directly applied to cross-city property appraisal models due to these heterogeneous location features. Hence, the cross-city property appraisal model transfer is a challenging task.
- We proposed a novel **H**omogeneous **F**eature **T**ransfer and **H**eterogeneous **L**ocation **F**ine-tuning **(HFT+HLF)** framework. By transferring part of our neural network and semi-supervised fine-tuning the remaining part, our model can surpass the fully supervised single-city ANN model by using only 20% to 30% of the available training data.

2 Related Work

2.1 Traditional Econometric Approaches

In the last few decades, there has been a large number of academic studies around the real estate appraisal area. Many of them employed the regression model for their valuation. Two major categories of traditional econometric approaches are hedonic regression models and hedonic price models.

The hedonic regression models have been extensively researched in academia and widely used in the industry for residential real estate mass appraisal for the past three decades. They range from simple hedonic regression [12,18], ridge regression [15] to the more complex quantile regression [14,26]. On the other hand, the hedonic price models make econometric analysis of the property attributes (e.g size, state, material etc.) as well as situational aspects such as the local environment (e.g. access bus and train stations) to understand trends in the housing market and accessing the factors which affect house prices. Additional variables such as inflation adjustment and regional planning strategy are often added [5,19].

However, the traditional econometric models usually assume these house characteristic are independent and non-interrelated which means that the value influence of the property attributes is considered to be constant. This assumption usually cannot correctly reflect the real-world real estate market. The traditional econometric methods are essentially model-oriented approaches which are aiming to explain the real estate prices and their variations.

2.2 Machine Leaning Based Approaches

The recent development of machine learning and deep learning are primarily driven by the abundance of available data and advances in computer technology. Many research works now focus on implementing the machine learning techniques into real estate appraisal system. Unlike the traditional approaches which are modelled assuming explicit rules, the machine learning based approaches try to learning the feature to price mapping automatically from data.

The genetic algorithm (GA) is a machine learning method which has been introduced into real estate appraisal in recent year [11,20]. The GA based approaches consider the real estate appraisal task as a multi-parameter optimisation problem and tackle this problem by stochastic search techniques based on Charles Darwin's evolutionary principle [17]. The decision tree approaches had also been applied to real estate appraisal area to overcome the potential problems relating to fundamental model assumption and independent variables [13]. Easy to understand and the ability to handle categorical variables make the decision tree approaches frequently used in many property appraisal systems. The random forest approach is an extended decision tree method by ensemble many simple regression trees to increase the overall appraisal accuracy [1]. For small and noise-free dataset, the random forest method is one of the most accurate approaches comparing with many other approaches like KNN or

ANN [1]. However, in a noisy real-world dataset, the performance of the random forest approach is significantly worse than ANN-based models [21]. Overlooking all kinds of models, the most commonly used machine learning approaches in the recent year are Artificial Neural Networks(ANN) [3,8,24,27]. The ANN-based approaches can outperform most of the traditional methods if the training dataset is large enough and the right training parameters are set [27]. Because the ANN approach yields the best performance when handling large datasets, our cross-city transfer model is developed base on ANN structure as well.

Most of the machine learning based studies only consider the location as a primary feature variable for training real estate appraisal models. However none has paid attention to location extensibility of the property appraisal models. For each new location such as a new city or a new region, a new model has to be retrained from scratch in order to accommodate the different location information. To overcome the particular limitation, we propose a cross-city transferable ANN-based property appraisal framework.

3 Methodology

3.1 Homogeneous and Heterogeneous Features

Real estate datasets usually contain three feature categories: location characteristics, building characteristics and apartment characteristics. The location characteristics tell where the buildings are located. The building characteristics describe the condition of the entire building and its neighbourhood area. The apartment characteristics are the apartment's internal construction such as the number of bedroom or living room. The variables of building characteristics and apartment characteristics are usually homogeneous for different cities. For example, the decoration level of two apartments of city A and city B can both be labelled from the same set of categorical variables: None, Partial, Simple, Mid-range, Deluxe, or Luxury. However, different cities have different sets of location information: different districts, different residential communities, etc. Hence, the location variables are heterogeneous for different cities. The heterogeneous location features and homogeneous property features are demonstrated in Fig. 1.

3.2 HFT+HLF Framework

Most of the ANN-based property appraisal methods have to replace and retrain the regressors for each new city due to the heterogeneous location features. Our model addresses this problem by separating the homogeneous and heterogeneous features during the training. The homogeneous features will be used to learn a cross-city transferable model and the city-dependant heterogeneous features will be used for location-based fine-tuning. As a result, our proposed network consists of two joined ANNs: homogeneous feature transferable ANN and heterogeneous location fine-tuning ANN, as shown in Fig. 2. The homogeneous features such as

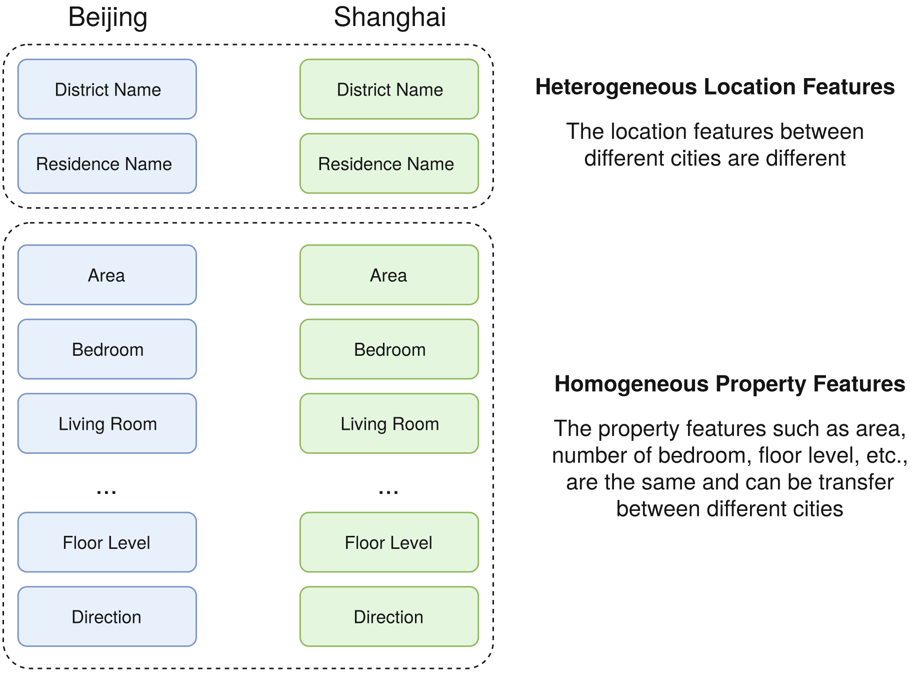

Beijing
Shanghai
District Name
Residence Name
District Name
Residence Name
Heterogeneous Location Features
The location features between different cities are different
Area
Bedroom
Living Room
...
Floor Level
Direction
Area
Bedroom
Living Room
...
Floor Level
Direction
Homogeneous Property Features
The property features such as area, number of bedroom, floor level, etc., are the same and can be transfer between different cities

Fig. 1. The cross-city estate dataset can be divided into two categories: homogeneous features and heterogeneous features. The homogeneous features usually describe the characteristics of the property such as direction, floor level, area. These features are usually city invariant. However, the location information like district names and apartment community names are city-dependent

apartment features and building features will be the inputs for the transferable part. The output feature maps from the transferable ANNs will then be concatenated with the heterogeneous location features and become the new inputs to the fine-tuning section of our proposed network. Since the homogeneous features are commonly shared between apartments of different cities, this part of our neural network can be considered as a generic apartment and building features learning network which could be transferred between different cities. By transfer this part of our network to a new city, the new city model only need to optimise the weight of the fine-tuning network. For example, the weights of the transferable ANN learned from the source city Beijing can be transferred to the new model of the target city Baotou and then fine-tuned with only a few data from the target city Baotou. The transferable ANN consists of 5 hidden layers with $(200, 100, 50, 20, 10)$ hidden nodes and the fine-tuning ANN consists of 4 hidden layers with $(100, 50, 20, 10)$ hidden nodes. All the hidden layers are equipped with 0.1 negative slope Leaky ReLU activation function, a 0.5 drop-out rate and Batch Normalization. We use the popular mean squared error (MSE) as the criterion loss for the price regression. Our network can be optimised by

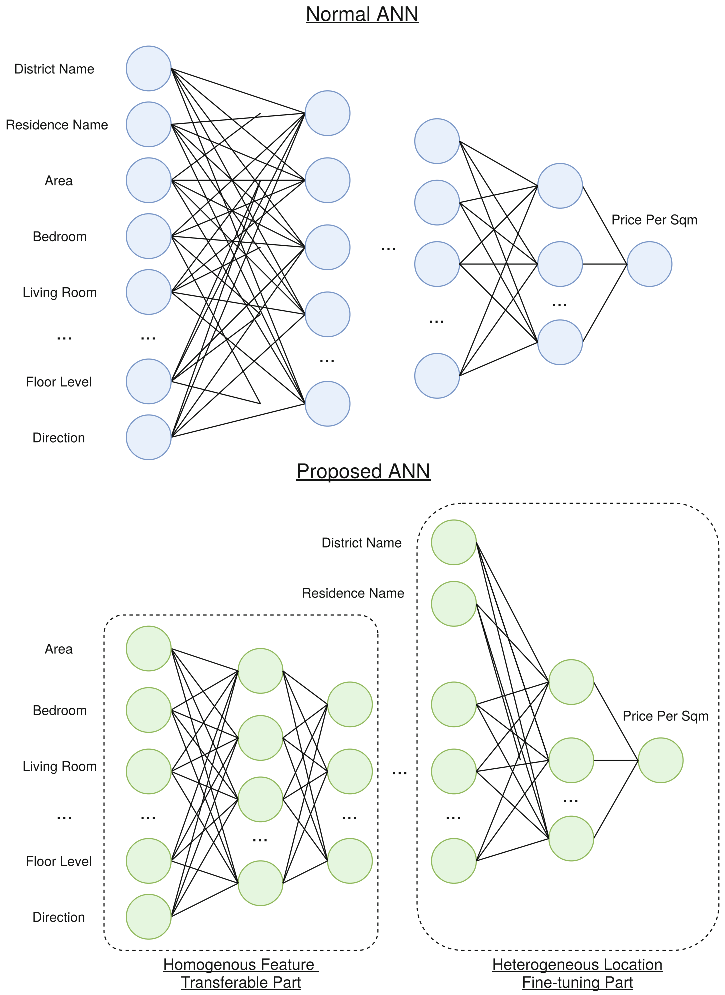


Fig. 2. The top diagram is the traditional ANN structure for property appraisal. The bottom figure is our proposed structure tailored for cross-city transfer learning. The transferable part can be transferred to the new model of a different city. Only the parameters in the fine-tune part need to be learned based on the data from a new city

Adam algorithm with AMSGrad moving average variant. The overview of our architecture is shown in the lower section of Fig. 2 with the standard traditional ANN structure added on top for comparison.

4 Empirical Evaluation

4.1 Database

In addition to the proposed location transfer appraisal method, another contribution of our study is the introduction and employment of a new massive cross-city dataset. The dataset was formed as a result of the collaboration between 2 partners, JinZheng and Cityre. JinZheng real estate appraisal company is a top level real estate appraisal Company in China, which cover all regions in China. Also, Cityre is a leading company of national real estate property information and data-service provider in China. This dataset contains two cities for each of three city tiers of China: Tier 1 includes 127,441 raw sales records (sale tag price) for Beijing, 243,222 raw records for Shanghai; Tier 2 includes 47,124 raw records for Jinan and 67,611 for Qingdao; Tier 3 includes that 5,609 raw records for Hohehaot, 9,614 raw records for Baotou. These cities were selected from the national setting of China for city tier classification. Beijing and Shanghai are the most famous Tier 1 city in China. Jinan and Qingdao are both Tier 2 cities from the same province. Hohhot and Baotou are Tier 3 cities from the same province. The details of our dataset is demonstrated in Table 1:

Table 1. The dataset is divided into three city tiers. Each city-tier contains two most famous or economically impactful cities. The raw records contain many missing values. The final processed data is the clean up data used for the empirical evaluation

Tier	Tier 1		Tier 2		Tier 3		Total
City	Beijing	Shanghai	Jinan	Qingdao	Hohhot	Baotou	
No. of District	18	18	9	12	5	4	66
No. of Residence	1,241	2,581	443	580	84	110	4,997
No. of Raw Data	278,371	580,211	104,149	133,988	19,334	25,407	1,141,460
No. of Processed Data	127,441	243,222	47,124	67,611	5,609	9,614	500,621

Our dataset surpasses most of the datasets used in most of the studies of property valuation. Brown and D'amato's work only use 725 and 390 dwellings receptively [6,10]. The recent works from Garcí?a and Arribas only use 591 and 2,149 records [2,7]. All six cities in our dataset have more data compared to all of them. Each residential real estate record in our dataset consisting of 15 variables were collected for each apartment: 3 describe the location, 10 describe the characteristics, 2 describe the building in which it is sited. The detailed descriptions of the variables are the following:

- **Location Characteristics**
  1. city: Six cities (Beijing, Shanghai, Jinan, Qingdao, Hohhot, Baotou).
  2. district: District the apartment is located
  3. residence: Name of the residential apartment

- **Building Characteristics**
  1. year: Year the building is completed
     - Mean 2003.95, Median 2005, Min 1900, Max 2019, Std 7.90
  2. building_type: 8 building types (Bungalow, High-rise, High-level, Multistorey, Entire Block, Semi-detached House, Detached House, Siheyuan)
- **Apartment Characteristics**
  1. price: Price per square meter
     - Mean 46,878, Median 44,666, Min 2,075, Max 288,690, Std 28,810
  2. area: Total area of the apartment
     - Mean 99.89, Median 88, Min 10, Max 2900, Std 59.87
  3. bedroom: Number of bedrooms
     - Mean 2.29, Median 2, Min 0, Max 9, Std 1.03
  4. livingroom: Number of livingrooms
     - Mean 1.56, Median 2, Min 0, Max 7, Std 0.56
  5. kitchen:Number of kitchen
     - Mean 0.97, Median 1, Min 0, Max 5, Std 0.19
  6. bathroom: Number of bathroom
     - Mean 1.32, Median 1, Min 0, Max 9, Std 0.64
  7. floor: Floor number of the apartment
     - Mean 5.36, Median 4, Min -10, Max 63, Std 4.94
  8. structure: Apartment Structure.
  9. decoration: 6 levels of the decoration (None, Partial, Simple, Mid-range, Deluxe, Luxury)
  10. direction: 10 Direction of the property (North, South, East, West, North-East, NorthWest, SouthEast, SouthWest, NorthSouth, EastWest)

4.2 Training and Settings

Cities of different tiers normally have different sizes of the training samples available. As shown in Table 1, the available data of first-tier cities (Beijing, Shanghai) is nearly 20 times more than the third-tier cities (Hohhot, Baotou). As a result, the batch size and the learning rate have to be individually set for each city tier. In our experiment, we set the learning rate at 0.005 with batch size 256 for Tier 1 cities, a 0.01 learning rate with batch size 128 for Tier 2 cities and 0.02 learning rate with batch size 64 for Tier 3 cities. The number of epochs is set 250 to ensure our network is fully converged during the training. Our network is implemented in PyTorch and the training times range from 2 hours to 30 min depending on the training sample size. We adopted the commonly used Root Mean Square Error (RMSE), R-squared Error (R^2) and Mean Absolute Percentage Error (MAPE) as performance metrics.

4.3 Experiment 1: Traditional ANN Vs Our Proposed ANN

Since our proposed network is a new ANN structure, the first experiment is to check whether the new structure affects the appraisal performance of the ANN model. In this experiment, we compare the performance of our proposed transferable model with a normal non-transferable ANN model in a fully supervised single city setting. This experiment follows the common 10-fold cross-validation strategy which is the average of the experimental results based on 10 random splits of the dataset into 90% training samples and 10% testing samples. The detailed performance metrics are shown in Table 2 below.

Table 2. Fully supervised single city learning performance comparison between traditional ANN and Our transferable ANN. The best results are highlighted by Bold

Model	Performance Metric	Tier 1		Tier 2		Tier 3	
		Beijing	Shanghai	Jinan	Qingdao	Hohhot	Baoto
Traditional ANN	RMSE (Lower Better)	0.246	0.186	**0.184**	**0.202**	**0.150**	0.145
	MAPE (Lower Better)	0.151	0.138	**0.142**	0.157	**0.121**	0.108
	R^2 (Higher Better)	0.804	**0.782**	**0.757**	**0.828**	0.522	**0.485**
Proposed ANN	RMSE (Lower Better)	**0.239**	**0.185**	0.189	0.204	0.170	**0.135**
	MAPE (Lower Better)	**0.149**	**0.136**	0.147	0.159	0.129	**0.102**
	R^2 (Higher Better)	**0.816**	0.778	0.744	**0.828**	**0.523**	0.473

Our model achieved similar and impressive low RMSE and MAPE scores in all six cities which means that our model can converge well on all the training datasets. However, the low R^2 scores of the Tier 3 city indicate the poor regression performance. It is mainly due to the lack of available data in the Tier 3 cities. This discovery supports our claim that the cross-city transfer learning is necessary for property appraisal model, especially to those low volume Tier 3 cities. By comparing our proposed models with standard non-transferable ANN method, our network yields a very similar overall performance. It proves that the new proposed architecture did not affect the overall appraisal performance.

4.4 Experiment 2: Semi-supervised Cross City Transfer Learning

Most other ANN appraisal methods, our proposed model has the ability to transfer a partially pre-trained network learned from the source city to the target city. Experiment 2 is conducted to validate the performance of our model in a cross-city semi-supervised setting.

Transfer to Tier 1 Cities. The first set of experiments is to test our model's ability to transfer the pre-train network to the first tier cities: Beijing and Shanghai. Based on the situation of China, the Tier 1 cities usually have a massive amount of data available for training. Therefore, the demand for transferring appraisal models conducted form Tier 2 or 3 cities to Tier 1 cities is highly unlikely in real-world practice. As a result, we conduct the Beijing to Shanghai and Shanghai to Beijing Transfer Learning in this experiment. For Shanghai to Beijing transfer, we first pre-train our model based on the training dataset of Shanghai and transfer the transferable part of our network to the new model for Beijing. Then, we select 10, 20, 30 records from each location (each residential community) in Beijing to fine tune the Beijing model. The test dataset is randomly selected from the remaining dataset with the size of 10% of the overall Beijing dataset. The processes for Beijing to Shanghai transfer are the same. As we only use a small amount of data from the Beijing training dataset, it can be considered as semi-supervised learning. The detailed performance metrics are shown in Table 3.

Table 3. The R^2 scores for Shanghai to Beijing and Beijing to Shanghai Transfer Learning.

Semi supervised	Training Size	Beijing (R^2 Score)	Training Size	Shanghai (R^2 Score)
		Shanghai -> Beijing		Beijing -> Shanghai
10 Records Per Residence	12,020	0.829	25,630	0.758
20 Records Per Residence	24,040	**0.847**	50,860	0.776
30 Records Per Residence	36,060	**0.845**	76,290	**0.788**
Full Supervised (No Transfer Learning)	127,441	0.816	243,222	0.778

By only using 20 records from each residential community, our semi-supervised transferable model can quickly achieve the similar or even superior performance compared with fully supervised single city learning. If the training dataset increase to 30 records per residential community, the overall performance even surpass the single city supervised learning. If the training set is 20 records from each residential community, the size of the training data is only one-fifth of the original training dataset. As a result, our proposed transferable model can be significantly reduced by five times.

Transfer to Tier 2 Cities. Table 4 shows the experiment results for Tier 1 Cities to Tier 2 Cities Transfer and Within Tier 2 Cities Transfer. The Tier 1 to Tier 2 transfer models usually outperform the inter Tier 2 transfer model. Because the Tier 1 cities have much more training data than Tier 2 and 3 cities, the transferable part of our network learned from Tier 1 cities usually have better generalization-ability and robustness.

Table 4. The R^2 scores for Tier 1 to Tier 2 Transfer and Inter-Tier-2 Transfer

Semi-supervised	Training Size	Jinan (R^2 Score)			Training Size	Qingdao (R^2 Score)		
		Beijing -> Jinan	Shanghai -> Jinan	Qingdao -> Jinan		Beijing -> Qingdao	Shanghai -> Qingdao	Jinan -> Qingdao
10 Records Per Residence	4,400	0.558	0.546	0.495	5,790	0.624	0.636	0.608
20 Records Per Residence	8,800	0.642	0.612	0.561	11,580	0.756	0.755	0.659
30 Records Per Residence	13,200	**0.784**	**0.773**	0.614	17,370	**0.836**	**0.838**	0.732
Full Supervised (No Transfer Learning)	47,124	0.744			67,611	0.828		

Table 5. The R^2 scores for Tier 1 to Tier 3 Transfer

Semi-supervised	Training Size	Hohhot (R^2 Score)		Training Size	Baotou (R^2 Score)	
		Beijing -> Hohhot	Shanghai -> Hohhot		Beijing -> Baotou	Shanghai -> Baotou
5 Records Per Residence	420	0.459	0.446	550	0.424	0.436
10 Records Per Residence	840	0.486	0.512	1100	0.456	0.455
15 Records Per Residence	1260	**0.547**	**0.588**	1650	**0.513**	**0.520**
Full Supervised (No Transfer Learning)	5048	0.523		8652	0.473	

Table 6. The R^2 scores for Tier 2 to Tier 3 Transfer

Semi supervised	Training Size	Hohhot (R^2 Score)		Training Size	Baotou (R^2 Score)	
		Jinan -> Hohhot	Qingdao –> Hohhot		Jinan –> Baotou	Qingdao –> Baotou
5 Records Per Residence	420	0.509	0.482	550	0.347	0.422
10 Records Per Residence	840	0.521	0.555	1100	0.448	0.451
15 Records Per Residence	1260	**0.548**	**0.585**	1650	**0.516**	**0.521**
Full Supervised (No Transfer Learning)	5048	0.523		8652	0.473	

Transfer to Tier 3 Cities. As the Tier 3 cities have limited amount of training data and the supervised Tier 3 cities models have relatively poor performance (very low R^2 scores), our experiment did not include the Tier 3 to Tier 3 cities transfer evaluation. We only focus on evaluating our network on Tier 1 or Tier 2 cities transfer to Tier 3.

Table 5 demonstrate the model transfer from Tier 1 cities (Beijing, Shanghai) to Tier 3 cities (Hohhot, Baotou). Table 6 demonstrate the model transfer from Tier 2 cities (Jinan, Qingdao) to Tier 3 cities (Hohhot, Baotou). By using only 15 records from each residential community, our model already outperforms the fully supervised model. In other words, the performance of our proposed model yield significant improvement after transferring either Tier 1 to Tier 3 or Tier 2 to Tier 3 cities. It shows that by transferring the homogeneous property features learning network trained from a more substantial training data (Tier 1 or 2 cities) can help to boost the performance of low data volume Tier 3 cities.

5 Conclusion

In this paper, we focus on solving a challenging problem in most of the existing property appraisal models: lack of adaptiveness and extensiveness for a different location. Because data for different cities contain completely different location feature, traditional property appraisal models have to be trained from scratch for each city. To address this problem, we presented a novel semi-supervised homogeneous features transfer and heterogeneous location fine-tuning network. We reconstruct the artificial neural network (ANN) into transferable homogeneous feature learning and heterogeneous location fine-tuning. By transferring the homogeneous feature learning component from a source city and fine-tune by a small amount of the target city's location feature, our semi-supervised model can achieve a similar or even superior performance of a fully supervised model. Meanwhile, the amount of training data required for our model is only 20% of fully supervised ANN models. By this proposed method, real estate appraisal models trained on data-rich cities can be applied to cities with insufficient real estate data without compromising the accuracy. It could be used to reduce data collection period, lower the model training cost and establish a better economy benchmark.

References

1. Antipov, E.A., Pokryshevskaya, E.B.: Mass appraisal of residential apartments: an application of random forest for valuation and a CART-based approach for model diagnostics. Expert Syst. Appl. **39**, 1772–1778 (2012)
2. Arribas, I., García, F., Guijarro, F., Oliver, J., Tamošiūnienė, R.: Mass appraisal of residential real estate using multilevel modelling. Int. J. Strateg. Prop. Manag. **20**, 77–87 (2016)
3. Bahrammirzaee, A.: A comparative survey of artificial intelligence applications in finance: artificial neural networks, expert system and hybrid intelligent systems. Neural Comput. Appl. **19**, 1165–1195 (2010)

4. Benjamin, J.D., Guttery, R.S., Sirmans, C.F.: Mass appraisal: an introduction to multiple regression analysis for real estate valuation. J. Real Estate Pract. Educ. **7**, 65–77 (2004)
5. Born, W.L., Pyhrr, S.A.: Real estate valuation: the effect of market and property cycles. J. Real Estate Res. **9**, 455–485 (1994)
6. Brown, K.H., Uyar, B.: A hierarchical linear model approach for assessing the effects of house and neighborhood characteristics on housing prices. J. Real Estate Educ. **7**, 15–24 (2004)
7. Cervelló, R., García, F., Guijarro, F.: Ranking residential properties by a multi-criteria single price model. J. Oper. Res. Soc. **62**, 1941–1950 (2011)
8. Chiarazzo, V., Caggiani, L., Marinelli, M., Ottomanelli, M.: A neural network based model for real estate price estimation considering environmental quality of property location. Transp. Res. Procedia **3**, 810–817 (2014)
9. Chopra, S., Thampy, T., Leahy, J., Caplin, A., LeCun, Y.: Discovering the hidden structure of house prices with a non-parametric latent manifold model. In: ACM International Conference on Knowledge Discovery and Data Mining (KDD) (2007)
10. D 'amato, M.: Comparing rough set theory with multiple regression analysis as automated valuation methodologies. Int. Real Estate Rev. **10**, 42–65 (2007)
11. Del Giudice, V., De Paola, P., Forte, F.: Using genetic algorithms for real estate appraisals. Buildings **7**, 31 (2017)
12. Downes, T.A., Zabel, J.E.: The impact of school characteristics on house prices : Chicago 1987–1991. J. Urban Econ. **52**, 1–25 (2002)
13. Fan, G.Z., Ong, S.E., Koh, H.C.: Determinants of house price: a decision tree approach. Urban Stud. **43**, 2301–2315 (2006)
14. Farmer, M.C., Lipscomb, C.A.: Using quantile regression in hedonic analysis to reveal submarket competition. J. Real Estate Res. **32**, 435–460 (2010)
15. Ferreira, E.J., Sirmans, G.S.: Ridge regression in real estate analysis. Appraisal J. **56**, 311 (1988)
16. Fu, Y., Xiong, H., Ge, Y., Yao, Z., Zheng, Y., Zhou, Z.H.: Exploiting geographic dependencies for real estate appraisal: a mutual perspective of ranking and clustering. In: ACM International Conference on Knowledge Discovery and Data Mining (KDD) (2014)
17. Goldberg, D.E., Holland, J.H.: Genetic algorithms and machine learning. Mach. Learn. **3**, 95–99 (1988). https://doi.org/10.1023/A:1022602019183
18. Isakson, H.R.: Using multiple regression analysis in real estate appraisal. Appraisal J. **69**, 424 (2001)
19. Kanojia, A., Khan, M.Y., Jadhav, U.: Valuation of residential properties by hedonic pricing method-a state of art. Int. J. Recent Adv. Eng. Technol. (IJRAET) (2016)
20. Kauko, T.: Residential property value and locational externalities: on the complementarity and substitutability of approaches. J. Prop. Invest. Financ. **21**, 250–270 (2003)
21. Kempa, O., Lasota, T., Telec, Z., Trawiński, B.: Investigation of bagging ensembles of genetic neural networks and fuzzy systems for real estate appraisal. In: Nguyen, N.T., Kim, C.-G., Janiak, A. (eds.) ACIIDS 2011. LNCS (LNAI), vol. 6592, pp. 323–332. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20042-7_33
22. Kontrimas, V., Verikas, A.: The mass appraisal of the real estate by computational intelligence. Appl. Soft Comput. J. **11**, 443–448 (2011)
23. Mansfield, J.R., Lorenz, D.P.: Shaping the future: the impacts of evolving international accounting standards on valuation practice in the UK and Germany. Prop. Manag. **22**, 289–303 (2004)

24. Mccluskey, W., Anand, S.: The application of intelligent hybrid techniques for the mass appraisal of residential properties. J. Prop. Invest. Financ. **17**, 218–239 (1999)
25. Musa, A.G., Daramola, O., Owoloko, A., Olugbara, O.: A neural-CBR system for real property valuation. J. Emerg. Trends Comput. Inf. Sci. **4**, 611–622 (2013)
26. Narula, S.C., Wellington, J.F., Lewis, S.A.: Valuating residential real estate using parametric programming. Eur. J. Oper. Res. **217**, 120–128 (2012)
27. Nghiep, N., Al, C.: Predicting housing value: a comparison of multiple regression analysis and artificial neural networks. J. Real Estate Res. **22**(3), 313–336 (2001)
28. Pagourtzi, E., Assimakopoulos, V., Hatzichristos, T., French, N.: Real estate appraisal: a review of valuation methods. J. Prop. Invest. Financ. **21**, 383–401 (2003)

Statistical Models of Dengue Fever

Hamilton Link[1(✉)], Samuel N. Richter[2], Vitus J. Leung[1], Randy C. Brost[1], Cynthia A. Phillips[1], and Andrea Staid[1]

[1] Sandia National Laboratories, Albuquerque, NM 87185, USA
helink@sandia.gov
[2] Missouri University of Science and Technology, Rolla, MO 65409, USA

Abstract. We use Bayesian data analysis to predict dengue fever outbreaks and quantify the link between outbreaks and meteorological precursors tied to the breeding conditions of vector mosquitos. We use Hamiltonian Monte Carlo sampling to estimate a seasonal Gaussian process modeling infection rate, and aperiodic basis coefficients for the rate of an "outbreak level" of infection beyond seasonal trends across two separate regions. We use this outbreak level to estimate an autoregressive moving average (ARMA) model from which we extrapolate a forecast. We show that the resulting model has useful forecasting power in the 6–8 week range. The forecasts are not significantly more accurate with the inclusion of meteorological covariates than with infection trends alone.

Keywords: Dengue fever · Gaussian process · ARMA · HMC · NOAA

1 Introduction

People in tropical areas are exposed to a number of diseases not prevalent in more temperate zones. Many of these diseases are spread by mosquitos, which thrive in tropical regions. With rising populations and changing climates increasing outbreaks, predicting outbreaks and preventing loss of life is becoming more urgent [6,18,19]. Using Bayesian data analysis techniques, we attempt to predict outbreak levels of dengue fever and quantify the predictive power of meteorological factors tied to mosquito breeding conditions. We hope improved outbreak forecasts enable preventive measures to save lives and reduce infection spread.

Dengue fever is a mosquito-borne virus prevalent in tropical climates. While many cases are asymptomatic or mild, dengue fever can become severe and even life-threatening in later stages, particularly with repeated infections.

Despite research, there is still no safe, effective vaccine. The most effective ways to prevent outbreaks are still to avoid mosquito bites and to control the mosquito population [18]. Thus, there is strong value in predicting outbreaks so that countermeasures may be deployed to mitigate or prevent them.

In 2015 the US National Oceanic and Atmospheric Administration (NOAA) and the Centers for Disease Control and Prevention, in collaboration with several other organizations, published the Dengue Forecasting project website [11] as

R. Islam et al. (Eds.): AusDM 2018, CCIS 996, pp. 175–186, 2019.
https://doi.org/10.1007/978-981-13-6661-1_14

a public call to encourage scientists to develop better infectious disease forecasting models. The site gave dengue-fever diagnosis data and potentially predictive environmental data for San Juan, Puerto Rico and Iquitos, Peru. They challenged researchers to predict the week of peak incidence, the maximum weekly incidence, and the total number of cases in each of four seasons of withheld case data.

We did not participate in the Dengue Forecasting project but used the data to study dengue prediction. We used Hamiltonian Monte Carlo (HMC) to estimate a seasonal Gaussian process infection rate, and an aperiodic function modeling the "outbreak level" of infection relative to (more or less severe than) seasonal trends. We believe coarse outbreak levels are appropriate for planning countermeasures. We show that an autoregressive moving average model derived from this outbreak level has useful forecasting power in the 6–8 week range. We chose autoregressive order and the number of exogenous weather covariates based on the posterior density of their model coefficients across the HMC samples (see Sect. 5.2).

2 Related Work

In the past two decades, researchers have taken many approaches to disease forecasting.

Buczak et al. [4] studied dengue cases in Peru, applying fuzzy associated rule mining (FARM) and logistic regression techniques. They use a temporal approach that produces outbreak estimates as time evolves as we do. However, they do not estimate outbreak probability distributions. Johnson, et al. [7] produced effective dengue forecasts using a Gaussian Process (GP) approach similar to ours, although they employed non-environmental hyperparameters instead of environmental factors. These non-environmental parameters required special estimation techniques. Ray et al. [12] predicted infectious disease outbreaks using kernel conditional density estimation (KCDE). They compute predictive distributions for outbreaks in individual weeks using KCDE and then tie those distributions together into joint distributions using copulas. They report that their method outperforms a baseline autoregressive integrated moving average (ARIMA) model for predicting dengue outbreaks in individual weeks.

Some previous modeling and prediction approaches are fairly distinct from ours. Examples include modeling disease spread in separate regions with different internal rates and movement between them [10], neural network techniques [14], statistical models with biological covariates [9], and methods using social media [2] and emergency center call records [13].

Recently researchers have proposed ensemble methods for combining multiple forecasting techniques to achieve improved ensemble performance. Yamana et al. [20] used Bayesian model averaging to combine different forecasts to produce a "superensemble" forecast. Applying their technique to San Juan, Puerto Rico, they reported that their ensemble forecasts outperformed the individual techniques. Buczak, et al. [3] constructed a large ensemble of 300 models, including Method of Analogues and Holt-Winters techniques. Their method produced good full-year prediction metrics such as the total number of cases, but was not as strong at temporally specific predictions.

We use the NOAA Dengue Forecasting challenge [11] data. Contestants had to predict metrics for an entire year: peak height, peak week, and total number of cases over a season. They made predictions at varying points in the year, so the proportion of known data already seen vs. future unknown data varied. Forecast windows varied from an initial 52 weeks to a final 4 weeks. In contrast, we predict future infections weekly with a fixed horizon. Several authors mentioned above have used this data; these include [3,7,14,20].

Other researchers have applied techniques closely related to ours to forecasting other diseases. Abeku, et al. [1] studied the accuracy of five methods for forecasting malaria cases. Forecasting malaria based on morbidity data, they report that a seasonal-adjustment method using deviation from expected seasonal values outperformed an ARIMA method. Soebiyanto, et al. [17] analyzed specific climatic factors affecting influenza in Hong Kong and Maricopa County (Arizona, USA), two regions with comparable temperatures but distinctly different rainfall. Comparing ARIMA models with and without climatic parameters, they found that including climatic variables gave better predictions of outbreaks.

See [5] and [16] for mechanisms, incubation time, and epidemiological factors of dengue fever.

3 Approach

3.1 Model Design

We developed a series of probabilistic models with hidden variables representing parameters governing the dengue fever season; the models were sampled using Stan, an implementation of Hamiltonian Monte Carlo sampling (HMC). HMC is a gradient-aware Markov chain Monte Carlo (MCMC) sampler; the simulator's random steps combine information from the parameter priors and the data likelihood to produce samples from the joint posterior distribution of the unknown variables.

We built our model in layers from a basic infection model $y_i \sim \text{Poisson}(x_i \theta)$, where the number of infections y_i is a Poisson-distributed random variable with a mean equal to the disease's characteristic infection rate θ times the population x_i. To estimate θ over time and across the two regions, we started with a regional reference infection rate and a hidden variable representing amplification – the tendency of environmental factors to increase the circulating virus levels [5] – relative to the reference region. Iquitos' per capita infection rates are substantially higher than San Juan's, and we chose San Juan as the reference region.

We modeled nominal seasonal fluctuations for each region using a shared Gaussian process. This allowed for a different high season in each region, while using data from both regions to estimate how smoothly infection rates change under normal circumstances. Unfortunately under the right conditions, infection rates can rapidly climb orders of magnitude beyond seasonal averages. We describe the outbreak level in any week as $\log(rate_w) - \log(s_w)$, the log-scale difference between the estimated infection rate for a given week of data and the estimated seasonal infection rate for that week of the year in that region. This outbreak level is estimated using a linear combination of Gaussian radial basis functions centered on each week of data, added to the seasonal fluctuations.

Under this model the remaining variation in the weekly observations y_w was plausibly explained by a Poisson distribution, and we used the outbreak level estimates to build an ARMA time series model and construct forecasts. We ran two weekly forecasts: one based solely on historical outbreak behavior and one that added weather data (using ARMA and ARMAX, respectively). Making each weekly forecast included rerunning the entire HMC model to update the outbreak level estimates in light of the previous week of observations, and providing seasonal estimates (historical weekly averages) of the upcoming weather data. We combined the outbreak level forecasts with the seasonal rates to produce a final rate forecast for the following weeks.

In addition to ARMA confidence-level rate windows, we calculated a "likelihood of outbreak" analogous to a conventional weather forecast. Using the ARMA forecast levels and a fixed regional threshold defining an abnormally severe regional "outbreak," we calculated the ARMA model's certainty that the number of observed cases would exceed the threshold for each week.

3.2 Outbreak-Beyond-Seasonal Model

Our Bayesian disease rate estimation model uses the following equations.

Basic Infection Model:

$$\log(\beta_0) \sim \mathcal{N}(0, 50) \tag{1}$$

$$a_{IQ} \sim \mathcal{N}(5, 2) \tag{2}$$

Seasonal Gaussian Process:

$$\eta^2 \sim \text{Cauchy}(0, 5) \tag{3}$$

$$1/\rho^2 \sim \text{Cauchy}(0, 5) \tag{4}$$

$$\sigma_{GP}^2 \sim \text{Cauchy}(0, 5) \tag{5}$$

$$\mathbf{K} = \text{Diag}(\eta^2 + \sigma_{GP}^2) + \left[\eta^2 e^{-\rho^2 \delta^2(i,j)}\right]_{i \neq j} \tag{6}$$

$$\text{where } \delta(i, j) = \min(j - i, (52 + i) - j) \tag{7}$$

$$\mathbf{f}_r \sim \mathcal{N}(\mathbf{0}, \mathbf{K}) \tag{8}$$

Outbreak Severity (beyond seasonal expectations):

$$l \sim \text{Cauchy}(1.5, 5) \tag{9}$$

$$\beta_w \sim \text{Student-}t(1, 0, 0.1) \tag{10}$$

$$b_w(t) = \exp(-|t - t_w|^2 / l^2) \tag{11}$$

$$\mathbf{g}_r(t) = \sum_w \beta_w b_w(t) \tag{12}$$

Final Rate Model:

$$rate_w = \beta_0 e^{\mathbf{f}_r(t_w)+\mathbf{g}_r(t_w)} a_r x_w \tag{13}$$

$$y_w \sim \text{Poisson}(rate_w) \tag{14}$$

Briefly, Eqs. 1 and 2 represent the log infections per million in San Juan, and its relative amplification in Iquitos' environment. Equation 8 is the Gaussian process used to draw random samples of a function for each region that varies smoothly throughout the year. The covariance matrix $\mathbf{K}$ controls the overall smoothness and amplification of these samples.

We estimate the outbreak level over time as a Gaussian basis function approximation with a shared width parameter l. We give the basis function coefficients a prior in Eq. 10, which assumes values near zero barring compelling evidence to the contrary.

Equation 13 combines these factors into an estimated infection rate for each week of data, indexed with w, and linked to the observed number y_w of cases by a Poisson distribution (a conventional distribution for a random number of events per interval).

4 Parameter Priors and Data

The Dengue Forecasting challenge [11] provided weekly numbers of dengue diagnoses, temperature, and precipitation in San Juan, Puerto Rico, and Iquitos, Peru. We averaged temperature and weather data over each week as others have done (e.g., [4]) and assumed seasonal averages for missing weather data.

The infection data from San Juan included 19 seasons (May 1990–April 2009), all of which we used. The data for Iquitos included 9 seasons (July 2000–June 2009), but the number of infections reported prior to January 2002 are not credible and we excluded that portion. We used the last 200 weeks of each region for testing. The number of infections per week ranged from individual cases to weeks-long outbreaks with up to 400 cases per week, with some weeks having no reported cases.

The Dengue Forecasting challenge test dataset provides similarly structured weekly data for the 2010–2016 seasons. This data became available only toward the end of our work and was not used.

The tabulated weeks of data each included a region index, a population estimate, the absolute week from January 1st 1990, the week of the year[1], and the number of cases reported.

The parameter priors we chose are included in the equations in Sect. 3.2. We gave the amplification coefficient for Iquitos a N(5,2) prior based on the observation that Iquitos has about one fifth the population of San Juan and roughly the same total number of dengue fever cases. We constrained the width parameter as $l \in (0, 4)$ with a N(1.5,5) prior to discourage sampling at the boundaries.

[1] Note the ordinal week of the year is not the same as the week of the regional dengue season, which was provided by NOAA but which we did not use.

We gave the coefficients of the basis functions a student-t(1,0,0.1) prior to encourage near-zero values when infections could be explained by seasonal fluctuations. The remaining parameter priors were chosen for convenience.

5 Results

5.1 HMC Performance

When using Monte Carlo sampling (e.g., HMC) it is important to check that the model converges consistently and is efficiently sampling the posterior distribution. The Gelman-Rubin statistic $\hat{\mathrm{R}}$ (good at about 1) and the effective sample size (ESS, good once at least in the hundreds) provide evidence of these properties respectively. Table 1 summarizes the results of running Stan on the available data with this model for a few key variables. The model chains converged to consistent values as measured by $\hat{\mathrm{R}}$, and had reasonably low autocorrelation which resulted in useful effective sample sizes. Visual inspection of the sample chains for other hidden variables was consistent with these results.

Table 1. Summary of HMC Output

	μ	σ^2	2.5%	50%	97.5%	ESS	$\hat{R}$
log Infection Rate per Person	−11.648	0.009	−11.666	−11.648	−11.629	1347	1.002
San Juan, Infections per Million	8.735	0.081	8.574	8.735	8.896	1350	1.002
Iquitos, Amplification	2.535	0.048	2.441	2.534	2.632	1344	1.001
Iquitos, Infections per Million	22.144	0.365	21.448	22.135	22.874	1917	0.999

We also looked for discrepancies between the learned model and the observed data using a visual comparison called a "posterior predictive check" (PPC). This entails plotting random samples from the model's posterior distribution against real observations to assess their similarity. Discrepancies can be particularly revealing when an evolving model begins to match data in summary statistics but has not yet learned subtle structure, and this can motivate further model improvements.

As a posterior predictive check, we sampled a Poisson "infection count" using the estimated rate at every step of the chain, for each week and for each region, and plotted the 2.5% and 97.5% quantile values over time. The seasonal rate fluctuations alone captured the correlation in infections from one week to the next but did not capture the variation in infection rates over multiple orders of magnitude from one year to the next. Once we allowed for rates above and beyond a seasonal average, the amount of variation in the data was far better explained, as shown in a PPC from the full HMC model over all data for San Juan, in Fig. 1.

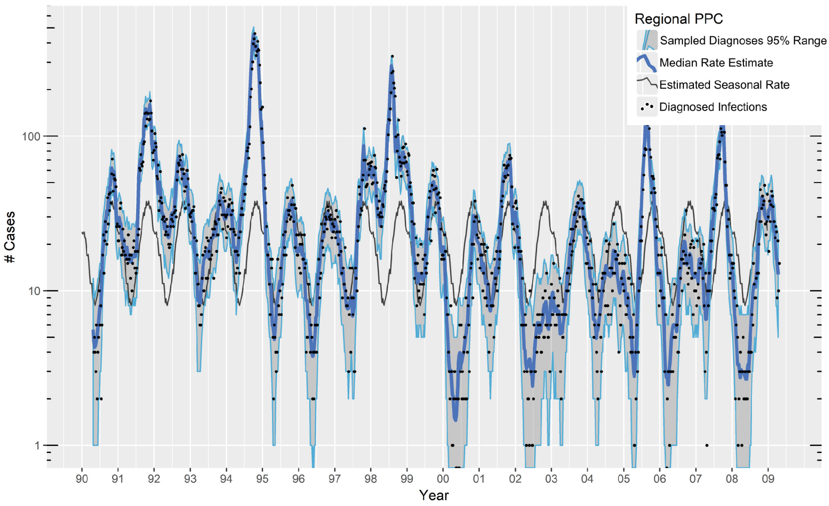


Fig. 1. San Juan (pop. 2.3M) infection rate model. Actual diagnoses made weekly, compared to estimated seasonal variation, posterior median rate estimate, and 95% high density region of expected weekly diagnoses.

5.2 Selection of ARMA/X Depth and Covariates

The Bayesian models let us fit the available data but do not directly make forecasts of the difference between future infection rates and the recurring seasonal fluctuations. These differences vary over several orders of magnitude, as illustrated in Fig. 2, and we applied ARMAX to the log-scale differences which we refer to as outbreak levels.

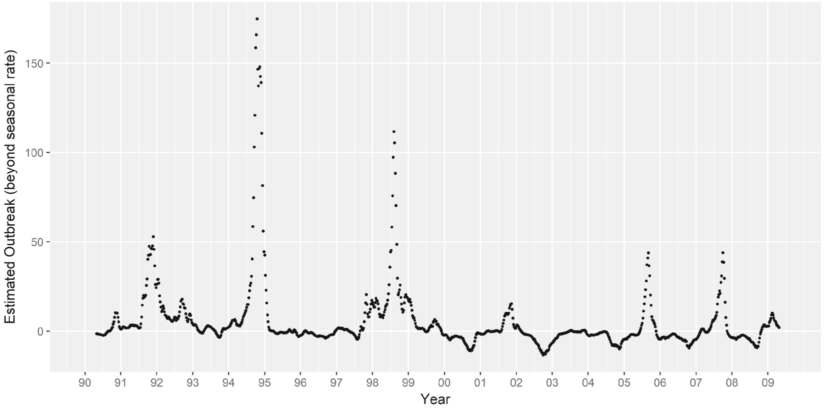


Fig. 2. Differences for San Juan between the weekly mean estimated infection rate and the mean seasonal rate for that week of the year. (In infections per million.)

To determine whether outbreaks could be reasonably attributed to weather characteristics, we looked at the high density region (HDR) of ARMAX coefficients calculated at every MCMC sample, running the rate estimate model over all available data. This analysis is comparable to the transfer-entropy calculation Buczak et al. used for model selection [3]. We used the HDRs to assess whether significant posterior likelihood was assigned to either side of zero and determined that we could conservatively include 6–10 weeks of history for precipitation and temperature. Figure 3 shows sample posterior distributions for the auto-regressive and precipitation covariate coefficients from Iquitos.

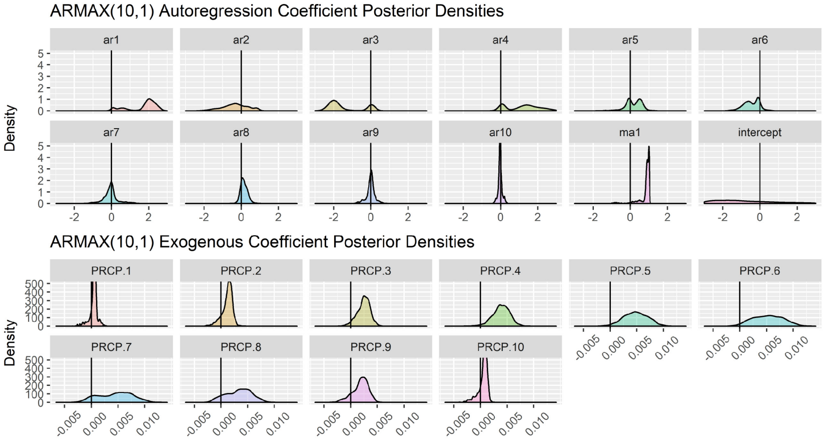


Fig. 3. The variation in some ARMAX coefficients (for Iquitos) due to random sample variation in the rate estimates. Coefficients were calculated as an HMC generated quantity, and are presented as posterior densities for autoregressive and moving average terms as well as the exogenous variable of precipitation.

5.3 Forecast Results

We ran the forecast process across the 200 weeks of test data, feeding in an additional week of data at a time. For both regions, we collected means and 2.5%, 50%, and 97.5% quantiles (of samples of the posterior distributions) for the seasonal infection rates and weekly infection rates over the history provided. At each time step, we then produced ARMA forecasts for the subsequent 13 weeks (3 months) of outbreak levels. From these forecasts we calculated accuracy of prediction comparing the number of infections reported to the rate forecast at different confidence levels. Figure 4 presents the resulting confidence vs. accuracy for each forecast horizon (from 1–13 weeks) for the ARMAX model. ARMA forecasts without weather data performed similarly.

Our forecasts were overconfident over very short (1–5 week) horizons, which we attribute to basis function approximation, which effectively dampens out noise before the ARMA coefficient estimation. After five weeks, the forecast

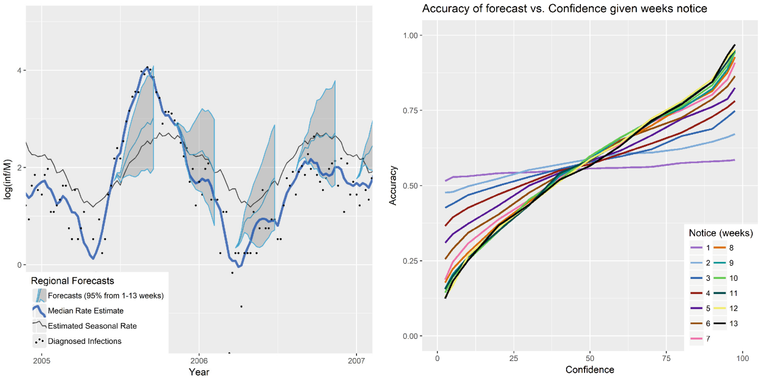


Fig. 4. (Left) Example individual forecasts, with a 13-week horizon. Discrepancies with the estimated median rate are due to the latter being calculated using all data, vs. the forecasts only being able to balance recent weeks of observations against seasonal trends for the immediate future. (Right) Forecast confidence level vs. forecast accuracy, plotted by horizon.

uncertainty balances out and the model begins to represent the varying likelihood of outcomes. Model confidence would typically be normalized at this stage and rechecked for accuracy on a tertiary test dataset; we expect to use the NOAA test data for this purpose.

Finally, we present a running 6-week forecast. We believe this time frame is not yet fully addressed by either early warning systems or full-season forecast models, and that it balances reasonable predictive power with enough advance notice to reduce population exposure and infections. The ARMAX model can produce either a maximum-likelihood estimate forecast or the estimated chance of exceeding some predetermined threshold. We present these side by side in Fig. 5. The former is more useful for evaluating whether a forecast at some horizon is reasonably accurate at an acceptable level of precision. The latter is potentially more useful for communicating short-term risk to the public and policy-makers.

Figure 5 shows the number of actual infection cases against the region's seasonal fluctuations, and against the ARMAX 95% level rate forecast with a 6-week horizon. This envelope generally captures the actual results, but because the ARMAX envelope is being projected from a log-linear process back into actual cases, the amount of uncertainty in outbreak severity is sometimes (perhaps appropriately) extreme. The intersection of these envelopes with the regional thresholds shown is used to estimate the likelihood that the number of infections will grow abnormally high. Note forecast values are shifted from when they are made so they are plotted against the week they pertain to.

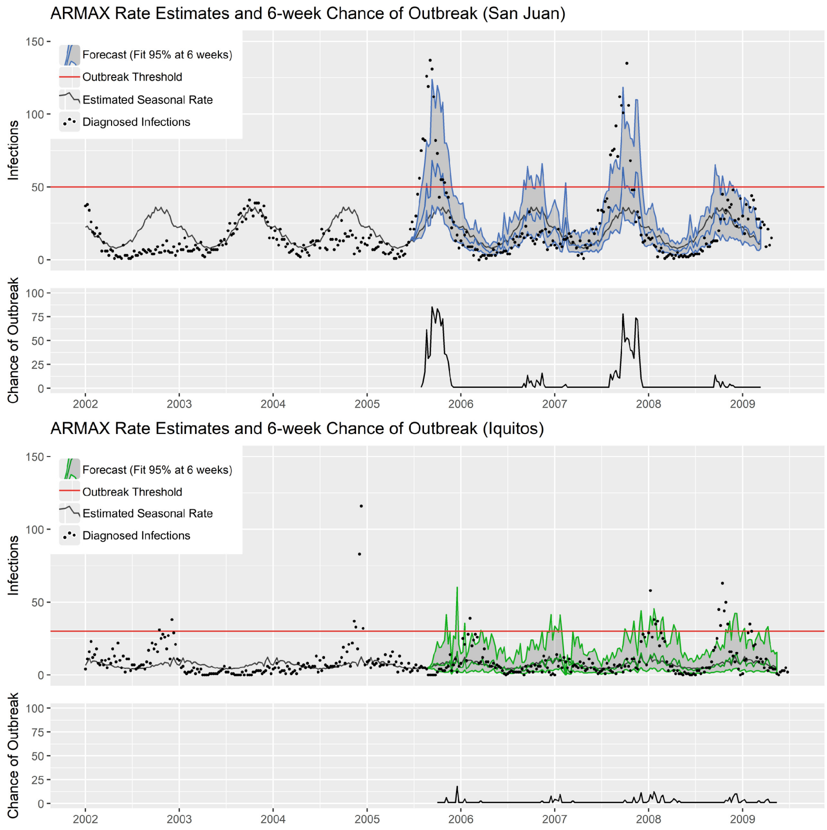


Fig. 5. Running forecasts for San Juan and Iquitos. Here we plot seasonal expectations, weekly diagnoses, 95% ARMAX interval at a 6-week horizon, and the predetermined outbreak threshold. The outbreak forecasts indicate the likelihood (from 0 – 100%) of exceeding the outbreak threshold in any given week, with 6 weeks notice.

6 Discussion

Forecasting disease outbreaks based on HMC and ARMA models of log-scale deviations from the norm is a viable approach and allows for expression of the forecast in terms of likely ranges of outcomes or the chance of exceeding a given threshold. These forecasts can be made far enough in advance that public service announcements and vector mitigation could reduce the spread of infection. Error remains in the specific week and scale of the outbreak that would need to be communicated to policy makers.

In the high density regions of ARMA coefficients built from HMC samples of estimated rates, we see potentially significant mass centered away from zero for coefficients up to around six weeks advance notice. This is consistent with the mosquito life cycle and disease incubation time. However, including weekly weather statistics for these frames did not appear to greatly impact our forecasts. Perhaps phenomena on these time scales have a diffused effect, so that most of the forecasting power present in either the ARMA or ARMAX models comes from mathematically capitalizing on observations from the early stages of outbreaks that start slowly and then grow exponentially.

Precipitation may be more informative than temperature for multiple reasons, based on coefficient HDRs and preliminary tests performed for Granger causality. Buczak et al. [3] showed significant correlation between infection levels and rainfall with 1–2 weeks lead time. However, in our preliminary decision tree ensemble experiments to determine feature importance for predicting Dengue outbreaks within the next nine weeks (omitted for space reasons), the most important was temperature in the most recent 1–2 weeks. Our results might improve with the use of weather data at a finer granularity.

While it remains to be seen what other information would help for long-term dengue predictions, there has been successful research using pharmacy sales or internet searches as strong predictive variables in disease outbreak early warning systems [8,15]. We hope that a mix of seasonal forecasting, short-term forecasting, and early warning systems can be combined to form more precise and accurate disease forecasting systems in the future.

Acknowledgements. This work was supported by the Laboratory Directed Research and Development program at Sandia National Laboratories, a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

References

1. Taekegn, A., et al.: Forecasting malaria incidence from historical morbidity patterns in epidemic-prone areas of Ethiopia: a simple seasonal adjustment method performs best. Trop. Med. Int. Health **7**(10), 851–857 (2002)
2. Ahmad, T., et al.: Characterizing dengue spread and severity using internet media sources. In: Proceedings of ACM DEV 2013, New York. ACM (2013)
3. Buczak, A.L., Baugher, B., Moniz, L.J., Bagley, T., Babin, S.M., Guven, E.: Ensemble method for dengue prediction. PLoS one **13**(1), January 2018
4. Buczak, A.L., Koshute, P.T., Babin, S.M., Feighner, B.H., Lewis, S.H.: A data-driven epidemiological prediction method for dengue outbreaks using local and remote sensing data. BMC Med. Inform. Decis. Mak. **12**, 124 (2012)
5. Gubler, D.J.: Desk Encyclopedia of human and medical virology, Chapter Dengue Viruses, pp. 372–382. Academic Press, Boston (2010)
6. Hales, S., de Wet, N., Maindonald, J., Woodward, A.: Potential effect of population and climate changes on global distribution of dengue fever: an empirical model. Lancet **360**, 830–834 (2002)
7. Johnson, L.R., et al.: Phenomenological forecasting of disease incidence using heteroskedastic gaussian processes: a dengue case study, August 2017

8. Kirian, M.L., Weintraub, J.M.: Prediction of gastrointestinal disease with over-the-counter diarrheal remedy sales records in the San Francisco Bay Area. BMC Med. Inform. Decis. Mak. **10**(1), 39 (2010)
9. Lauer, S.A., et al.: Prospective forecasts of annual dengue hemorrhagic fever incidence in Thailand, 2010–2014. In: Proceedings of the National Academy of Sciences (PNAS), February 2018
10. Masui, H., Kakitani, I., Ujiyama, S., Hashidate, K., Shiono, M., Kudo, K.: Assessing potential countermeasures against the dengue epidemic in non-tropical urban cities. Theor. Biol. Med. Model. **13**, 12 (2016)
11. NOAA. Combating dengue with infectious disease forecasting. Technical report, National Oceanic and Atmospheric Administration, DOC, 5 June 2015. Retrieved from Dengue Forecasting http://dengueforecasting.noaa.gov/
12. Ray, E.L., Sakrejda, K., Lauer, S.A., Johansson, M.A., Reich, N.G.: Infectious disease prediction with kernel conditional density estimation. Stat. Med. **36**(30), 4908–4929 (2017)
13. Rehman, N.A., Kalyanaraman, S., Ahmad, T., Pervaiz, F., Saif, U., Subramanian, L.: Fine-grained dengue forecasting using telephone triage services. Sci. Adv. **2**(7), e1501215 (2016)
14. Sathler, C.: Predictive modeling of dengue fever epidemics: A Neural Network Approach, December 2017
15. Shortridge, J.E., Guikema, S.D.: Public health and pipe breaks in water distribution systems: analysis with internet search volume as a proxy. Water Res. **53**, 26–34 (2014)
16. Simmons, C.P., Farrar, J.J., Nguyen, V.V., Wills, B.: Dengue. N Engl. J Med **366**(15), 1423–1432 (2012)
17. Soebiyanto, R.P., Adimi, F., Kiang, R.K.: Modeling and predicting seasonal influenza transmission in warm regions using climatological parameters. PLoS One **5**(3), e9450 (2010)
18. WHO. Dengue: Guidelines for diagnosis, treatment, prevention and control. Technical report, WHO/TDR (2009)
19. WHO. Dengue and severe dengue fact sheet. Retrieved from World Health Organization http://www.who.int/mediacentre/factsheets/fs117/en/, 29 July 2016
20. Yamana, T.K., Kandula, S., Shaman, J.: Superensemble forecasts of dengue outbreaks. J. R. Soc. **13**, (20160410) (2016)

Privacy and Clustering

Reference Values Based Hardening for Bloom Filters Based Privacy-Preserving Record Linkage

Sirintra Vaiwsri(✉), Thilina Ranbaduge, and Peter Christen

Research School of Computer Science,
The Australian National University, Canberra, ACT 2601, Australia
sirintra.vaiwsri@anu.edu.au

Abstract. Privacy-preserving record linkage (PPRL) is the process of identifying records that refer to the same entities across different databases without revealing any sensitive information about these entities. A popular PPRL technique that is efficient and effective is Bloom filter encoding. However, recent research has shown that Bloom filters are vulnerable to cryptanalysis attacks that aim to re-identify sensitive attribute values encoded into Bloom filters. As counter-measures, hardening techniques have been developed that modify the bit patterns in Bloom filters. One recently proposed hardening technique is BLoom-and-flIP (BLIP), which randomly flips bit values according to a differential privacy mechanism. However, while making Bloom filters more resilient to attacks, applying BLIP can lower linkage quality. We propose and evaluate a reference values based BLIP mechanism which ensures that Bloom filters for similar encoded sensitive values are modified in a similar way, resulting in improved linkage quality compared to standard BLIP hardening.

Keywords: Data linkage · Differential privacy · Encoding · Perturbation

1 Introduction

Many organizations collect millions of records about individuals (such as customers, patients, or tax payers) in their databases which often need to be integrated to facilitate effective data mining. *Record linkage* aims to link records in different databases that refer to the same entity [5]. Privacy is an important aspect that needs to be considered when databases that contain sensitive personal data, such as names and addresses, are linked across databases.

Privacy-preserving record linkage (PPRL) [20] techniques have been developed to match records between databases without revealing sensitive data. In PPRL the values in a set of attributes common to all databases are encoded in some form to ensure their privacy. Different categories of privacy techniques have been developed for PPRL [20]. The first category are secure multi-party computation (SMC) based techniques that perform matching of encrypted records.

R. Islam et al. (Eds.): AusDM 2018, CCIS 996, pp. 189–202, 2019.
https://doi.org/10.1007/978-981-13-6661-1_15

While provably secure, SMC techniques generally have high computation costs. The second category are perturbation based techniques which modify the actual attribute values using an encoding technique, resulting in a trade-off between linkage quality, scalability to linking large databases, and privacy [21].

A widely used perturbation technique for PPRL is Bloom filter (BF) encoding [3,15]. As we discuss in detail in Sect. 3, a BF is a bit vector that encodes values using a set of independent hash functions [15]. BF encoding is now being used in several linkage applications in the health sector [4]. However, recent studies have shown that sets of BFs can be attacked with the aim of re-identifying the sensitive attribute values encoded in them [6,7,12,13]. Most attack methods exploit that frequent BFs or bit patterns correspond to frequent q-grams (sub-strings of length q characters) in the sensitive values encoded in BFs.

To counteract such attack methods, hardening techniques have been developed to improve the security of BF based PPRL techniques [16,19]. As we discuss in detail in the next section, these hardening techniques further modify BFs to reduce or eliminate any frequency information that could be exploited by attack methods. One drawback of existing hardening techniques is however that they have a trade-off between privacy and linkage quality, because modifications of BF bit patterns will likely lead to an increase in falsely matched and missed true matching record pairs. Certain hardening techniques have also shown to be vulnerable to a frequency-based cryptanalysis attack [6].

One recently proposed hardening technique is BLoom-and-flIP (BLIP) [2,16] which flips bit values at certain positions in a BF according to a differential privacy mechanism [9]. In our evaluation we show that such random bit flipping can lead to a considerable decrease in linkage quality. To overcome this weakness of BLIP hardening, we propose to use reference values from a global database to determine the bit positions to be flipped. The use of reference values ensures that similar BFs are modified in the same way (thus maintaining high similarities) while different BFs are modified differently (resulting in lower similarities). We name our approach as RBBF for **R**eference based **BLIP** **BF** hardening.

In this paper we specifically contribute (1) a novel approach to select a suitable set of reference values from a publicly available large database; (2) an improved BLIP hardening technique for BFs based on selected single and multiple reference values; (3) an analysis of our approach in terms of complexity and linkage quality; and (4) an experimental evaluation using a real-world database.

2 Related Work

Since the mid 1990s PPRL techniques have been developed to link sensitive data without having to reveal any actual attribute values. PPRL has evolved from simple exact matching of encrypted strings only to sophisticated approximate matching of encoded values in large databases [20].

In 2009 Schnell et al. [15] proposed to use BF encoding for scalable PPRL that also allowed approximate matching of records by calculating similarities between BFs. Inspired by their approach various BF based PPRL techniques have been developed since then [20]. Varying from two-party to multi-party protocols, these techniques classify record pairs as matches and non-matches based on the number of 1-bits their corresponding BFs have in common [15].

Recently, several attacks on BF encodings for PPRL have been proposed that aim to re-identify the attribute values encoded in BFs [6,7,12,13]. Most of these attacks are based on a frequency analysis of bit pattern distributions. To overcome these attacks hardening methods can be applied on BFs [16,18].

Salting is a hardening technique that can be used for PPRL [17] with the aim to create different bit patterns for the same q-gram by adding an extra value to each q-gram before it is encoded (such as the year of birth for a person). Therefore two attribute values that have the same q-gram set but different salting values (like different years of birth) will be mapped to different bit positions in a BF.

Balancing was proposed by Schnell and Borgs [16] as a hardening method to generate uniform Hamming weight (number of 1-bits) distributions for BFs. To generate a balanced BF, a BF is concatenated with the negated copy of itself, such that the resulting BF will always have 50% of its bits set to 1. The bits in the balanced BFs are then randomly permuted to improve privacy. XOR folding is another hardening method proposed by the same authors [18], where a given BF of length l is first divided into two segments of length $l/2$ and then the bit-wise XOR operation is applied on these segments to generate a new BF.

However, both balanced and XOR folded BFs have been successfully attacked. A recently proposed attack method [6] was able to correctly re-identify some of the attribute values that have been encoded into hardened BFs because the frequency distribution of BFs and their bit patterns does not change even after balancing or XOR folding has been applied.

A novel hardening technique is 'BLoom-and-flIP' (BLIP) [1,2,16], which randomly flips values in certain bit positions in BFs based on differential privacy characteristics [9]. The approach is similar to the RAPPOR method [10] which is being used to anonymously collect user responses during sensitive data collections. As we detail in Sect. 3, one drawback of the original BLIP approach is that the random bit flipping can lead to a significant loss of linkage quality.

In contrast to the original BLIP approach, we use reference values to improve the quality of linked records. The idea of using reference values extracted from a (publicly available) global database for PPRL was first investigated by Pang et al. [14]. However, the use of reference values in the BF encoding and hardening process has not been explored so far.

3 Background and Preliminaries

We now describe the building blocks required for our improved BLIP hardening technique which we then discuss in detail in Sect. 4.

Bloom Filter Encoding for PPRL: Bloom filter (BF) encoding [3] is a widely used perturbation techniques for PPRL [20]. A BF $\mathbf{b}$ is a bit vector of length l initially set to 0-bits. In PPRL the string values from the records to be compared

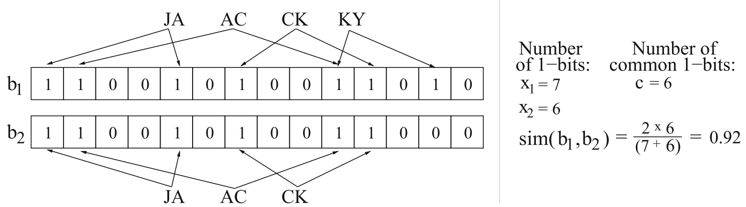


Fig. 1. Two example Bloom filters that are encoding the string value pair 'JACKY' and 'JACK' using two hash functions, with their Dice coefficient similarity calculation.

in the linkage process are first converted into character q-grams which are then encoded into a BF using a set of independent hash functions [15] by setting corresponding bits to 1, as shown in Fig. 1.

The similarity between two BFs $\mathbf{b}_1$ and $\mathbf{b}_2$ can be calculated using the Dice-coefficient [5,15]. First, the number 1-bits of each BF, x_1 and x_2, and the number of 1-bits that occur in common at the same bit positions in both BFs, c, are counted. The similarity is then calculated as: $sim(\mathbf{b}_1, \mathbf{b}_2) = (2 \times c)/(x_1 + x_2)$.

BLoom-and-flIP (BLIP) Hardening: BLIP was originally proposed by Alaggan et al. [2] as a non-interactive differentially private [9] approach to randomize BFs in the context of privacy-preserving comparisons of user profiles in social networks. BLIP randomly flips bits at certain positions in a BF based on a user defined flip probability. We refer the reader to Alaggan et al. [2] for details and a proof showing how BLIP fulfills non-interactive differential privacy. Schnell and Borgs were the first to explore BLIP in the context of PPRL [16].

For a given bit flipping probability, f, following Alaggan et al. [2], a bit $\mathbf{b}[p]$ in a BF $\mathbf{b}$ at position p is flipped according to Eqs. (1) resulting in the value $\mathbf{b}'[p]$ at position p in the new randomized BF $\mathbf{b}'$.

$$\mathbf{b}'[p] = \begin{cases} 1 & \text{if } \mathbf{b}[p] = 0 \text{ with probability } f, \\ 0 & \text{if } \mathbf{b}[p] = 1 \text{ with probability } f, \\ \mathbf{b}[p] & \text{with probability } 1 - f. \end{cases} \tag{1}$$

The BLIP approach used by Schnell and Borgs [16] was based on the idea proposed by Erlingsson et al. [10] as part of their RAPPOR technique which allows anonymous collection of user statistics from software products such as Web browsers. Again assuming a flip probability f, the new bit $\mathbf{b}'[p]$ at position p in the new randomized BF $\mathbf{b}'$ is flipped from $\mathbf{b}[p]$ according to Eq. (2):

$$\mathbf{b}'[p] = \begin{cases} 1 & \text{with probability } \frac{1}{2}f, \\ 0 & \text{with probability } \frac{1}{2}f, \\ \mathbf{b}[p] & \text{with probability } 1 - f. \end{cases} \tag{2}$$

If for example the flip probability is set to $f = 0.05$ for a BF of length $l = 1,000$ bits then 50 randomly selected bits will be flipped using the first approach from Eq. (1) while 950 bits are unchanged. With the approach used

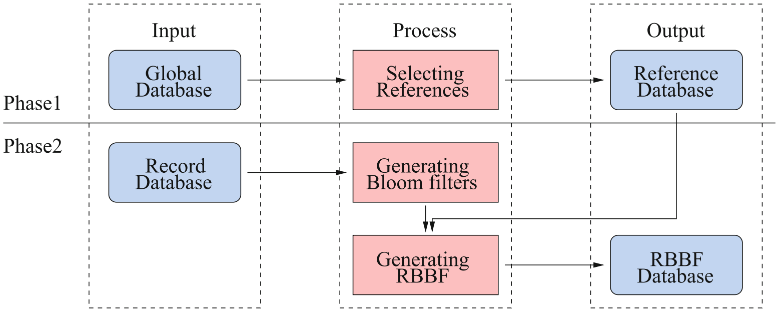


Fig. 2. Overview of reference values based BLIP Bloom Filter hardening.

by Schnell and Borgs [16] in Eq. (2), however, bits will not be flipped according to their original state, but rather 50 randomly selected bits will be set to 0 or 1 with equal probability. As a result, depending upon which BLIP approach is used, the numbers of 1-bits in randomized BFs will likely differ.

If a BF has less than 50% 1-bits then applying Eq. (1) will mean it will have more 1-bits after randomization that when applying Eq. (2). This can potentially lead to lower linkage quality because more 1-bits can increase the similarities between randomized BFs and thus leads to more false positive matches.

4 Protocol Description

As outlined in Fig. 2, in the first phase of our approach we select a set of suitable reference values from a global database, and in the second phase we use these reference values to determine how to apply BLIP when randomizing BFs.

4.1 Phase 1: Selecting Reference Values

We assume all database owners (DOs) [20] who aim to encode and harden their sensitive databases have access to a publicly available global database **G** from which they can extract a set of reference values **R**. Note that for phase 2 of our approach, as described in Sect. 4.2, all DOs must use the same set of reference values, **R**. Therefore this set **R** is either generated in the same way by all DOs, or alternatively one DO generates **R** and distributes it to all other DOs.

As detailed in Algorithm 1, the reference value selection process aims to find string values that are all different to each other according to a given similarity threshold. In other words no pair of selected reference values has an approximate string similarity above a similarity threshold, s_t, according to the used similarity function $sim()$. The first phase (Algorithm 1) consists of the following steps:

1. The reference value set, **R**, is initialized and the first value, g, from the global database, **G**, is added to the empty set **R** (lines 1 to 4).

Algorithm 1. *Selecting reference Values (Phase 1)*

```
Input:
- $\mathbf{G}$:      Publicly available global data set          - $s_t$: Similarity threshold
- $sim()$: Approximate string similarity function
Output:
- $\mathbf{R}$: Reference value set
1:  $\mathbf{R} = \{\}$                // Initialize the reference value set
2:  for $g \in \mathbf{G}$ do:         // Loop over all values in the global data set
3:     if $\mathbf{R} == \{\}$ do:     // Check if the reference set is empty
4:        $\mathbf{R} = \mathbf{R} \cup \{g\}$       // Add the selected first global value to the reference set
5:     else:
6:        $s_{max} = 0$                // Initialize the maximum similarity value
7:        for $r \in \mathbf{R}$ do:   // Loop over all so far selected reference values
8:           $s = sim(g, r)$   // Calculate similarity between the global value and the reference value
9:           $s_{max} = max(s_{max}, s)$       // Get the maximum similarity
10:       if $s_{max} \leq s_t$ do:              // Check maximum similarity is less than the threshold
11:          $\mathbf{R} = \mathbf{R} \cup \{g\}$                // Add global value to reference value set
12: return $\mathbf{R}$
```

2. Using the similarity function $sim()$, we then compare all other values $g \in \mathbf{G}$ with each previously selected reference value $r \in \mathbf{R}$, and we keep track of the maximum similarity, s_{max}, between g and any $r \in \mathbf{R}$ (lines 6 to 9).
3. If the maximum similarity s_{max} between g and any $r \in \mathbf{R}$ is lower than the threshold s_t, then g is different enough from all so far selected reference values and is therefore added to $\mathbf{R}$ (lines 10 to 12). Steps 2 and 3 are repeated until all global values $g \in \mathbf{G}$ have been processed.

4.2 Phase 2: Reference Values Based BLIP Bloom Filter Hardening

In the second phase of our approach each DO first encodes the records in its own database $\mathbf{V}$ into BFs, where these BFs are then hardened using a reference based BLIP approach, as detailed in Algorithm 2. As described below, the BLIP based randomization of BFs using reference values can employ one or more reference values for a given record value v, where these reference values are the k most similar values in $\mathbf{R}$ (from Algorithm 1). The idea of RBBF is that two similar record values, v_i and v_j, from $\mathbf{V}$ will likely have similar sets of reference values, $\mathbf{r}_{v_i}$ and $\mathbf{r}_{v_j}$, and when using these reference values as random seeds means that similar BLIP based randomization will be applied for v_i and v_j.

We harden a basic BF $\mathbf{b}_q$ for the q-gram set of a record value v for each of the k reference values for v, and concatenate all hardened BFs into one final BF $\mathbf{b}_v$ for v that is of length $k \times l$, where l is the length of the original BF $\mathbf{b}_q$. A final random permutation of all BFs $\mathbf{B}_v$ (agreed by all DOs) ensures an external attacker cannot identify the bit positions of an individual hardened BF generated using a certain reference value (which is unknown to an attacker). The second phase (Algorithm 2) consists of the following steps:

1. In lines 1 and 2 the set of BLIP hardened BFs $\mathbf{B}$ is initialized first, as is a list of permuted bit positions $\mathbf{p}$ that will be used to permute all generated BFs in the same way. This list $\mathbf{p}$ basically contains all bit positions from 1 to $k \times l$ (the length of the final hardened BFs) randomly permuted.

Algorithm 2. ***Reference value based BLIP Bloom Filter (RBBF) hardening (Phase 2)***

```
Input:
- $\mathbf{V}$:   Record value set                          - $q$:      Q-gram length
- $\mathbf{R}$:   Reference value set                       - $l$:      BF Length
- $\mathbf{H}$:   Hash function set                         - $f$:      BLIP flip probability
- $\mathbf{A}$:   Attribute value set                       - $b_m$:    Blip method, either $ala$ [2] or $rap$ [10,16]
- $k$:   Number of reference values per BF         - $sim()$: Approximate string similarity function
Output:
- $\mathbf{B}$: Set of RBBF encoded values from $\mathbf{V}$
1:  $\mathbf{B} = \{\}$                                  // Initialize the set of RBBF encoded values
2:  $\mathbf{p} = genBitPosPermList(l \times k)$         // Generate a list of permuted bit positions
3:  for $v \in \mathbf{V}$ do:                           // Loop over all records
4:    $\mathbf{q} = genQgramSet(v, \mathbf{A}, q)$       // Generate q-gram set for the record value $v$
5:    $\mathbf{b}_q = genBF(\mathbf{q}, \mathbf{H}, l)$  // Generate basic Bloom filter for q-gram set $\mathbf{q}$
6:    $\mathbf{r}_v = getMostSimRefValSet(\mathbf{R}, v, sim, k)$ // Get the $k$ reference values most similar to $v$
7:    $\mathbf{b}_v = []$                                // Initialize an empty Bloom filter for record $v$
8:    for $r \in \mathbf{r}_v$ do:                       // Loop over reference values for record value $v$
9:      $setRandomGeneratorSeed(r)$                      // Initialize the PRNG
10:     if $b_m == ala$ then:                            // Apply Eq. 1
11:       $\mathbf{b}_r = applyAlaBLIP(f, \mathbf{b}_q)$
12:     else:                                            // Apply $rap$ (RAPPOR) BLIP method, Eq. 2
13:       $\mathbf{b}_r = applyRapBLIP(f, \mathbf{b}_q)$
14:     $\mathbf{b}_v = concatenateBF(\mathbf{b}_v, \mathbf{b}_r)$  // Append to final hardened Bloom filter for record $v$
15:   $\mathbf{b}_v = permuteBF(\mathbf{b}_v, \mathbf{p})$       // Permute the final Bloom filter for record $v$
16:   $\mathbf{B}[v] = \mathbf{b}_v$                             // Add RBBF hardened Bloom filter to the output set
17: return $\mathbf{B}$
```

2. The main loop (from line 3) iterates over all record values $v \in \mathbf{V}$, where in line 4 a value v is converted into its q-gram set $\mathbf{q}$ based on the set $\mathbf{A}$ of attributes to be encoded into BFs, and length of q-grams q. These q-gram sets are then encoded into a basic BF $\mathbf{b}_q$ (line 5).
3. In line 6, the k most similar reference values to v are identified from the reference values set $\mathbf{R}$ (as generated by Algorithm 1) as the set $\mathbf{r}_v$.
4. We then initialize an empty BF $\mathbf{b}_v$ for record value v (line 7), and loop over the selected reference values $r \in \mathbf{r}_v$ in line 8. We use each reference value r as the random seed for a pseudo-random number generator (PRNG) in line 9, and then we apply the selected BLIP method (using one of Eqs. (1) or (2)) and the flip probability, f (in lines 10 to 13).
5. The resulting BLIP hardened BF $\mathbf{b}_r$ is then appended (concatenated) to the end of the record BF $\mathbf{b}_v$ in line 14.
6. A final permutation of $\mathbf{b}_v$ in line 15 ensures an attacker cannot identify the individual BFs that were BLIP hardened with a certain reference value. The BF $\mathbf{b}_v$ is then inserted into the list of all BFs $\mathbf{B}$ for record v in line 16.

4.3 Complexity and Linkage Quality Analysis

We now analyze our approach in terms of its complexity and linkage quality.

Complexity: The computational complexity of Algorithm 1 depends upon the size of $\mathbf{G}$ as well as the similarity threshold s_t. If s_t is set to a high value then more values in $\mathbf{G}$ are added into $\mathbf{R}$. In the worst case, if $s_t = 1.0$ (assuming the similarity function returns a normalized value $0 \leq sim() \leq 1$) then all values in $\mathbf{G}$ will be added into $\mathbf{R}$ leading to a complexity of Algorithm 1 of $O(|\mathbf{G}|^2)$.

The main loop in Algorithm 2 iterates over all record values in $\mathbf{V}$ leading to a complexity of $O(|\mathbf{V}|)$. Generating the q-gram set $\mathbf{q}$ and the basic BF $\mathbf{b}_q$ in lines 4 and 5 are of complexity $O(Q \cdot |\mathbf{H}|)$ where we assume Q is the average number of q-grams in a value $v \in \mathbf{V}$, and $|\mathbf{H}|$ is the number of hash functions used to encode q-grams into BFs. Finding the k most similar reference values to a given $v \in \mathbf{V}$ from $\mathbf{R}$ requires $|\mathbf{R}|$ similarity calculations. The BLIP randomization in lines 8 to 14 for the selected k reference values has a complexity of $O(k \cdot l)$ where l is the length of the original BF. Finally, the permutation of the concatenated BF $\mathbf{B}_v$ also has a complexity of $O(k \cdot l)$ as a loop over all bit positions is required. Overall, the complexity of Algorithm 2 is $O(|\mathbf{V}| \cdot (Q \cdot |\mathbf{H}| + |\mathbf{R}| + 2 \cdot k \cdot l))$.

Linkage Quality: The two main parameters of our approach that will affect the final linkage quality (besides the quality of the input data and the general parameters used for BF encoding and BLIP randomization) are the similarity threshold, s_t, in Algorithm 1 and the number of reference values, k, in Algorithm 2.

If a lower s_t is used then the set of reference values $\mathbf{R}$ will be smaller. Hence it will be more likely that two dissimilar record values will have the same value(s) in $\mathbf{R}$ and thus the same BLIP randomization will be applied on their BFs. This potentially lowers the precision of linkage quality because it will lead to an increased BF similarity if the same bit positions are flipped to 1-bits. A higher s_t leads $\mathbf{R}$ to contain more values and thus a higher likelihood that dissimilar record values will have different reference values leading to different BLIP randomization. Therefore, a higher s_t should result in higher linkage quality.

When using more reference values, k, per record value then the linkage quality will likely increase because there is a higher chance that similar record values share the same reference value(s), leading to similar BLIP randomization. On the other hand, when using less reference values it is more likely that dissimilar record values share the same reference value(s) which can lower linkage quality.

5 Experimental Evaluation and Discussion

We evaluated our proposed RBBF hardening approach using the North Carolina Voter Registration (NCVR) database (see: https://dl.ncsbe.gov), where we used a subset of 224,073 records as the global database $\mathbf{G}$ from where we extracted reference values. Using stratified sampling we identified 1,000 record pairs where we had 100 pairs in each of the ten similarity intervals $[0.0, 0.1)$, $[0.1, 0.2)$, $\ldots$, $[0.9, 1.0)$. We encoded different attribute combinations into BFs: (1) first name, (2) first and last names, and (3) first, last, street and town names. We set the similarity threshold in Algorithm 1 to $s_t = [0.4, 0.6, 0.8]$, and the number of reference values in Algorithm 2 to $k = [1, 3]$. We converted attribute values into q-grams using $q = 2$ and encoded them into BFs of length $l = 1{,}000$ using different numbers of hash functions, and set the BLIP flip probabilities $f = [0.01, 0.05, 0.1]$.

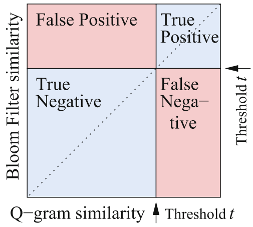


Fig. 3. Calculation of precision and recall based on q-gram and Bloom filter Dice similarities.

Table 1. Number of reference values generated for different attribute combinations using different values for the similarity threshold s_t.

Attribute combinations	$s_t = 0.4$	$s_t = 0.6$	$s_t = 0.8$
First name (FN)	1,217	5,949	14,442
Last name (LN)	2,375	12,943	29,682
Street (ST)	5,084	43,268	119,601
Town names (TN)	280	583	720
FN and LN	4,391	40,725	154,952
FN, LN and ST	16,891	184,606	219,926
FN, LN and TN	2,441	51,793	180,513
FN, LN, ST, and TN	6,027	115,000	216,497

Table 2. Average run times (in seconds) for BLIP and RBBF for different attribute combinations, numbers of reference values, k, and flip probabilities, f.

Attribute combinations	$f = 0.01$				$f = 0.1$			
	BLIP	$k = 1$	$k = 3$	$k = 6$	BLIP	$k = 1$	$k = 3$	$k = 6$
FN	0.159	0.163	0.482	0.968	0.165	0.169	0.511	1.007
FN and LN	0.316	0.322	0.945	1.867	0.321	0.328	0.980	1.954
FN, LN, ST, and TN	0.308	0.316	0.935	1.869	0.323	0.331	0.985	1.957

We implemented all approaches using Python 2.7 and ran experiments on a server with 2.4 GHz CPUs running Ubuntu 16.04. We compared RBBF with BFs without any hardening (No-BLIP) and the two BLIP approaches by Alaggan et al. [2] (BLIP-A), and Schnell and Borgs [16] (BLIP-S). We named RBBF based on Eqs. (1) and (2) as RBBF-A and RBBF-S, respectively.

In the evaluation we compared Dice similarities, as discussed in Sect. 3, calculated between q-gram sets [5] with Dice similarities calculated between BFs. As Fig. 3 shows, for a pair of records we assumed the q-gram Dice similarity s_Q to be the true similarity. For a given similarity threshold t we then classified the corresponding BF pair with its Dice similarity, s_B, as a true positive (TP) if both $s_Q \geq t$ and $s_B \geq t$, a false negative (FN) if $s_Q \geq t$ and $s_B < t$, a false positive (FP) if $s_Q < t$ and $s_B \geq t$, and a true negative (TN) if $s_Q < t$ and $s_B < t$. We calculated precision as $P = TP/(TP + FP)$ and recall as $R = TP/(TP + FN)$, We do not present F-measure results given recent research [11].

In Table 1 we show the number of reference values generated by Algorithm 1 for different attribute combinations and similarity threshold values, s_t. As can be seen, higher values of s_t resulted in more reference values in $\mathbf{R}$. Furthermore, attributes (or combinations) with more unique values ended up with more reference values, which can lead to better linkage quality of RBBF hardened BFs.

Table 3. Precision and recall for attribute first name with 40 hash functions used for Bloom filter encoding. The best results for each f and t setting are shown in bold.

Method	Flip probability (f)	$t = 0.7$		$t = 0.8$		$t = 0.9$	
		Prec	Reca	Prec	Reca	Prec	Reca
No-BLIP	-	0.711	1.0	0.737	1.0	1.0	1.0
BLIP-A	0.01	**0.891**	**1.0**	0.953	0.976	1.0	0.65
BLIP-S	0.01	0.82	1.0	**0.955**	**1.0**	1.0	0.75
RBBF-A	0.01	0.886	1.0	0.883	0.964	1.0	0.725
RBBF-S	0.01	0.825	1.0	0.841	1.0	**1.0**	**0.85**
BLIP-A	0.05	1.0	0.284	1.0	0.167	0.0	0.0
BLIP-S	0.05	1.0	0.658	1.0	0.5	1.0	0.1
RBBF-A	0.05	0.959	0.39	0.778	0.25	1.0	0.325
RBBF-S	0.05	**0.963**	**0.748**	**0.871**	**0.595**	**1.0**	**0.4**
BLIP-A	0.1	1.0	0.013	0.0	0.0	0.0	0.0
BLIP-S	0.1	1.0	0.297	1.0	0.167	0.0	0.0
RBBF-A	0.1	0.889	0.152	0.728	0.226	1.0	0.3
RBBF-S	0.1	**0.962**	**0.432**	**0.778**	**0.333**	**1.0**	**0.35**

However, the computational requirements of RBBF increase with more reference values, as we discussed in Sect. 4.3. Table 2 shows average run times for BLIP (averaged for BLIP-A and BLIP-S) and RBBF. As can be seen, as expected an increase of k led to increased run times, as did longer encoded q-gram sets. However the flip probability, f, did not seem to affect run times.

In Tables 3, 4 and 5 we show precision and recall results (calculated as described above) for three selected attribute combinations. Due to limited space we only show results where the number of reference values is $k = 3$ and the reference similarity value is $s_t = 0.8$ because these settings gave the best results for all attribute combinations. We used a number of hash functions appropriate to the length of the q-grams sets that needed to be encoded into BFs [8,15].

As shown in Tables 3, 4 and 5, and as expected, without hardening the BFs (No-BLIP) Dice similarities were very close to the q-gram Dice similarities. While this will result in good linkage quality the known vulnerability to cryptanalysis attacks of not hardened BFs makes basic BF encoding not suitable for secure PPRL. As can be seen from these results, standard BLIP (BLIP-A and BLIP-S) led to very low precision and recall values of 0.0 with higher flip probabilities, while even with higher flip probabilities our RBBF approach achieved results of high precision while recall suffered for certain parameter settings and attribute combinations. As more attributes were encoded into BFs both precision and recall decreased especially with higher Dice similarity thresholds because the BLIP and RBBF randomization mechanisms led to lower Dice similarities.

Table 4. Precision and recall for attributes first name and last name with 30 hash functions used for Bloom filter encoding. Best results are shown in bold.

Method	Flip probability (f)	$t = 0.7$		$t = 0.8$		$t = 0.9$	
		Prec	Reca	Prec	Reca	Prec	Reca
No-BLIP	-	0.61	1.0	0.719	1.0	0.189	1.0
BLIP-A	0.01	**0.655**	**1.0**	**0.866**	**1.0**	**0.6**	**0.857**
BLIP-S	0.01	0.636	1.0	0.804	1.0	0.368	1.0
RBBF-A	0.01	0.653	1.0	0.834	1.0	0.393	0.929
RBBF-S	0.01	0.634	1.0	0.794	1.0	0.275	1.0
BLIP-A	0.05	**0.91**	**0.905**	1.0	0.203	0.0	0.0
BLIP-S	0.05	0.753	0.995	0.979	0.748	0.0	0.0
RBBF-A	0.05	0.892	0.917	0.962	0.5	**0.518**	**0.5**
RBBF-S	0.05	0.733	0.998	**0.94**	**0.89**	0.383	0.5
BLIP-A	0.1	1.0	0.19	0.0	0.0	0.0	0.0
BLIP-S	0.1	0.909	0.896	1.0	0.203	0.0	0.0
RBBF-A	0.1	0.989	0.441	0.942	0.309	**0.433**	**0.5**
RBBF-S	0.1	**0.885**	**0.934**	**0.911**	**0.537**	0.35	0.5

Table 5. Precision and recall for attributes first and last names, street, and town name with 20 hash functions used for Bloom filter encoding. Best results are shown in bold.

Method	Flip probability (f)	$t = 0.7$		$t = 0.8$		$t = 0.9$	
		Prec	Reca	Prec	Reca	Prec	Reca
No-BLIP	-	0.46	1.0	0.421	1.0	0.155	1.0
BLIP-A	0.01	0.471	1.0	**0.444**	**1.0**	**0.254**	**1.0**
BLIP-S	0.01	0.467	1.0	0.438	1.0	0.195	1.0
RBBF-A	0.01	**0.474**	**1.0**	0.442	1.0	0.204	1.0
RBBF-S	0.01	0.466	1.0	0.431	1.0	0.175	1.0
BLIP-A	0.05	**0.547**	**1.0**	**0.679**	**0.984**	1.0	0.067
BLIP-S	0.05	0.488	1.0	0.52	1.0	**1.0**	**0.733**
RBBF-A	0.05	**0.547**	**1.0**	0.626	0.996	0.289	0.567
RBBF-S	0.05	0.489	1.0	0.475	1.0	0.278	0.9
BLIP-A	0.1	**0.741**	**0.992**	1.0	0.071	0.0	0.0
BLIP-S	0.1	0.546	1.0	**0.679**	**0.984**	1.0	0.067
RBBF-A	0.1	0.71	0.998	0.705	0.567	**0.337**	**0.533**
RBBF-S	0.1	0.54	1.0	0.593	0.996	0.236	0.567

Table 6. Re-identification results for a frequency based attack [6] on first name values with 40 hash functions used for Bloom filter encoding and different flip probabilities.

	No BLIP	$f = 0.01$				$f = 0.1$			
		BLIP-A	BLIP-S	RBBF-A	RBBF-S	BLIP-A	BLIP-S	RBBF-A	RBBF-S
1-1 Corr %	0	0	0	0	0	0	0	0	0
1-Many Corr %	9	100	0	8	8	100	100	8	8
Wrong %	2	0	2	2	2	0	0	2	2
No %	89	0	98	90	90	0	0	90	90

Overall the results shown in Tables 3, 4 and 5 also indicate that the RAPPOR [10,16] based BLIP hardening approach from Eq. (2) seemed to outperform the approach proposed by Alaggan [2] from Eq. (1). Our proposed RBBF approach outperformed both standard BLIP approaches with regard to linkage quality for most parameter settings and attribute combinations.

In Table 6 we show re-identification results using a recently proposed frequency-based cryptanalysis attack [6], showing exact one-to-one, one-to-many, wrong and no re-identification percentages for the top 100 most frequent first names. As can be seen from these results, RBBF slightly improved privacy compared to not hardened BF encoding. Interestingly, the attack was still able to correctly re-identify 100% of all first names in a one-to-many manner in BLIP. This indicates that standard BLIP might not be as secure as originally hoped, and further research is required to investigate these results.

6 Conclusion

We have presented and improved the BLoom-and-flIP (BLIP) hardening technique for Bloom filter encoding for privacy-preserving record linkage. Our approach selects reference values from a large publicly available database and uses these values to modify the BLIP approach such that similar record values are randomized in a similar way. Our results on a real voter database showed that our approach is able to outperform standard BLIP approaches [2,10,15] while ensuring the hardened Bloom filters are secure with regard to attacks. In future work we aim to investigate linkage quality and privacy of RBBF on different data sets, develop approaches to calculate the optimal flip probability for RBBF to minimize the number of false positives while providing enough privacy, and improve the privacy of RBBF to avoid re-identification by adding randomness into the reference value selection process based on q-gram frequencies.

References

1. Alaggan, M., Cunche, M., Gambs, S.: Privacy-preserving Wi-Fi analytics. PET **2018**(2), 4–26 (2018)
2. Alaggan, M., Gambs, S., Kermarrec, A.-M.: BLIP: non-interactive differentially-private similarity computation on Bloom filters. In: Richa, A.W., Scheideler, C. (eds.) SSS 2012. LNCS, vol. 7596, pp. 202–216. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33536-5_20
3. Bloom, B.: Space/time trade-offs in hash coding with allowable errors. Commun. ACM **13**(7), 422–426 (1970)
4. Boyd, J.H., Randall, S.M., Ferrante, A.M.: Application of privacy-preserving techniques in operational record linkage centres. In: Gkoulalas-Divanis, A., Loukides, G. (eds.) Medical Data Privacy Handbook, pp. 267–287. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23633-9_11
5. Christen, P.: Data Matching - Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31164-2
6. Christen, P., Schnell, R., Vatsalan, D., Ranbaduge, T.: Efficient cryptanalysis of Bloom filters for privacy-preserving record linkage. In: Kim, J., Shim, K., Cao, L., Lee, J.-G., Lin, X., Moon, Y.-S. (eds.) PAKDD 2017, Part I. LNCS (LNAI), vol. 10234, pp. 628–640. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-57454-7_49
7. Christen, P., Vidanage, A., Ranbaduge, T., Schnell, R.: Pattern-mining based cryptanalysis of Bloom filters for privacy-preserving record linkage. In: Phung, D., Tseng, V.S., Webb, G.I., Ho, B., Ganji, M., Rashidi, L. (eds.) PAKDD 2018, Part III. LNCS (LNAI), vol. 10939, pp. 530–542. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93040-4_42
8. Durham, E., Kantarcioglu, M., Xue, Y., Toth, C., Kuzu, M., Malin, B.: Composite Bloom filters for secure record linkage. IEEE TKDE **26**(12), 2956–2968 (2014)
9. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006, Part II. LNCS, vol. 4052, pp. 1–12. Springer, Heidelberg (2006). https://doi.org/10.1007/11787006_1
10. Erlingsson, Ú., Pihur, V., Korolova, A.: Rappor: randomized aggregatable privacy-preserving ordinal response. In: ACM SIGSAC (2014)
11. Hand, D., Christen, P.: A note on using the F-measure for evaluating record linkage algorithms. Stat. Comput. **28**(3), 539–547 (2018)
12. Kroll, M., Steinmetzer, S.: Who is 1011011111...1110110010? Automated cryptanalysis of Bloom filter encryptions of databases with several personal identifiers. In: Fred, A., Gamboa, H., Elias, D. (eds.) BIOSTEC 2015. CCIS, vol. 574, pp. 341–356. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-27707-3_21
13. Kuzu, M., Kantarcioglu, M., Durham, E., Malin, B.: A constraint satisfaction cryptanalysis of Bloom filters in private record linkage. In: Fischer-Hübner, S., Hopper, N. (eds.) PETS 2011. LNCS, vol. 6794, pp. 226–245. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22263-4_13
14. Pang, C., Gu, L., Hansen, D., Maeder, A.: Privacy-preserving fuzzy matching using a public reference table. In: McClean, S., Millard, P., El-Darzi, E., Nugent, C. (eds.) Intelligent Patient Management. SCI, vol. 189, pp. 71–89. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00179-6_5
15. Schnell, R., Bachteler, T., Reiher, J.: Privacy-preserving record linkage using Bloom filters. BMC Med. Inform. Decis. Mak. **9**(1), 41 (2009)

16. Schnell, R., Borgs, C.: Randomized response and balanced Bloom filters for privacy preserving record linkage. In: ICDMW DINA (2016)
17. Schnell, R.: Privacy-preserving record linkage. In: Harron, K., Goldstein, H., Dibben, C. (eds.) Methodological Developments in Data Linkage (2015)
18. Schnell, R., Borgs, C.: XOR-folding for Bloom filter-based encryptions for privacy-preserving record linkage. Working paper, German Record Linkage Center (2016)
19. Schnell, R., Rukasz, D., Borgs, C., Brumme, S., et al.: R PPRL toolbox (2018). https://cran.r-project.org/web/packages/PPRL/
20. Vatsalan, D., Sehili, Z., Christen, P., Rahm, E.: Privacy-preserving record linkage for big data: current approaches and research challenges. In: Zomaya, A.Y., Sakr, S. (eds.) Handbook of Big Data Technologies, pp. 851–895. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-49340-4_25
21. Vatsalan, D., Christen, P., O'Keefe, C.M., Verykios, V.: An evaluation framework for privacy-preserving record linkage. JPC **6**(1), 35–75 (2014)

Canopy-Based Private Blocking

Yanfeng Shu[1](✉), Stephen Hardy[2], and Brian Thorne[2]

[1] Data61, CSIRO, Hobart, Australia
Yanfeng.Shu@csiro.au
[2] Data61, CSIRO, Eveleigh, Australia
{Stephen.Hardy,Brian.Thorne}@csiro.au

Abstract. Integrating data from different sources often involves using personal information for linking records that correspond to the same real-world entities. This raises privacy concerns, leading to development of privacy preserving record linkage (PPRL) techniques which aim to conduct linkage without revealing private or confidential information of the corresponding entities. To make privacy methods scalable to large datasets, in this paper, we propose a novel blocking approach that adapts canopy clustering for a private setting. Our approach features using public reference data as a basis to form blocks, and involving redundancy in block assignments. We provide an analysis on the approach's privacy and experimentally evaluate its performance in terms of efficiency and effectiveness. The results show that our approach is scalable with the size of datasets and achieves better quality than the state-of-the-art sorted neighborhood based approaches.

Keywords: Record linkage · Privacy · Blocking · Canopy clustering

1 Introduction

Many real-world applications require integration of data from different sources to improve data quality or to facilitate data analysis. This often involves using personal information such as names and addresses for identifying and linking records that correspond to the same real-world entities across different sources. Such personal information, however, may not be available due to privacy and confidentiality concerns. For example, it may be a privacy violation for hospitals to disclose the information that could identify patients. To address this, there has been development of privacy preserving record linkage (PPRL) techniques, which aims to conduct linkage without revealing private or confidential information of the corresponding entities.

Two main types of privacy methods have been used in PPRL: secure multiparty computation (SMC) methods (e.g. [1]) and data transformation methods (e.g [3]). SMC methods provide secure and accurate solutions to the problem of PPRL, but are computationally prohibitive in practice. Data transformation methods, on the other hand, are more practical but can have problems of information leakage and false matches. Both SMC and transformation methods involve comparing records across data sources. To reduce the number of record

R. Islam et al. (Eds.): AusDM 2018, CCIS 996, pp. 203–215, 2019.
https://doi.org/10.1007/978-981-13-6661-1_16

pair comparisons and make privacy methods scalable to large datasets, blocking techniques are deployed, which group together record pairs that are likely to match into blocks. As a result, comparisons between record pairs only have to be made on those belonging to the same blocks instead of on all possible pairs, thus facilitating the matching process.

In this paper, we propose a novel blocking technique that adapts canopy clustering [13] to a *private* setting. Through the use of publicly available reference data, we partition the data space to generate overlapping subspaces referred to as reference canopies. These canopies are then used as a basis to form overlapping blocks on the data sources. In particular, we assign records to multiple blocks based on their distances to canopies, and merge blocks based on their sizes and distances. We provide an analysis on the approach's privacy and experimentally evaluate its performance in terms of efficiency and effectiveness. The results show that our approach is scalable with the size of datasets and achieves better quality than the state-of-the-art sorted neighborhood based approaches.

The rest of the paper is organised as follows. Section 2 reviews related work. In Sect. 3, we formulate the problem and outline the canopy clustering technique which we base on for our blocking. We present our approach in Sect. 4 and its privacy analysis in Sect. 5. In Sect. 6, we report our experimental results. Finally, we conclude the paper in Sect. 7.

2 Related Work

Various blocking techniques have been used for record linkage applications. In this part, we focus on private blocking techniques for PPRL. Blocking for PPRL was first introduced by Al-Lawati et al. [2] who proposed several blocking schemes based on tokens. Further developments on private blocking have been made in recent years. One use of blocking techniques is to eliminate unnecessary secure similarity computations. Inan et al. [9] proposed one such approach where a SMC step was preceded by a blocking step that used value generation hierarchies to filter out the record pairs that were expected to be non-matches. The work by Kuzu et al. [12] followed similar ideas, but achieved private blocking through hierarchical clustering on public identifiers and controlling information leakage using differential privacy [7].

Another use of blocking techniques is to reduce the complexity of data transformation methods. For example, Durham [6] proposed two locality sensitive hashing (LSH) based blocking approaches, Jaccard LSH (JLSH) and Hamming LSH (LSH), for partitioning bloom filter encoded records, and showed that both approaches considerably reduced the number of record pair comparisons required while achieving high result quality, and in particular, HLSH had better performance than JLSH. Based on bloom filters, Ranbaduge et al. [14–16] proposed several approaches for multi-party blocking through the construction of trees on bloom filters, splitting bloom filters based on their secure sums, or clustering bloom filters based on hashing.

Our proposal belongs to a group of private blocking techniques that can be used with any SMC or data transformation methods. There are several such proposals. Karakasidis and Verykios [11] proposed a blocking technique (k-NN) based on nearest neighbour clustering, where privacy was addressed by using a publicly available reference table and having each block consisting of at least k elements of the reference table. Vatsalan et al. [18,19] proposed a sorted neighborhood based blocking technique for both two-party and three-party cases that combined k-anonymous clustering and the use of public reference values. Karakasidis et al. [10] proposed a blocking technique based on the use of multiple reference sets, where records were assigned to multiple blocks and blocks were defined over overlapping pairs of blocking attributes. Han et al. [8] proposed a blocking technique based on k-anonymous blocking on numerical attributes and secure similarity computations for measuring the similarity of two blocks.

Among these private blocking proposals [8,10,11,18,19], the proposals of [10,11,18] are most related to ours in that they all involve three parties and use public reference data for privacy. Both our work and the work in [11,18] assumed only one reference set used, and could be extended by using multiple reference sets as in [10] to achieve better blocking quality. As the sorted neighborhood based approaches [18] have been shown to have better performance than the K-NN approach [11] in both efficiency and effectiveness, in this work, we focus on comparing our approach with the sorted neighborhood based approaches only. One main feature of the sorted neighborhood based approaches is that blocks are sorted based on the lexicographic order of blocking values of records (as such, they are quite efficient). This, however, may lead to the problem of records being assigned to wrong blocks when there are errors in the start of record blocking values. Our approach does not suffer from this problem as it performs blocking based on the similarity of blocking values. We evaluate our approach against the sorted neighborhood based approaches experimentally in Sect. 6.

3 Preliminaries

We provide a formal description of the private blocking problem of PPRL and briefly describe the canopy clustering approach on which our blocking technique is based.

3.1 Problem Formulation

Without loss of generality, we assume two datasets D^A and D^B, owned by parties Alice and Bob respectively, and that D^A and D^B have the same schema with attributes $\{Attr_1, ..., Attr_m\}$[1].

PPRL. The task for PPRL is to identify and link all pairs of D^A and D^B's records that refer to the same real world entities without disclosing any private or confidential information of the entities. As in [12], we consider the task as a classification task where the classifier labels each record pair as “match” or

[1] Schema matching is another research topic beyond the scope of this paper.

"non-match". Let $sim : dom(Attr_i) \times dom(Attr_i) \mapsto \mathbb{R}$ be a similarity function, w_i be the weight for attribute $Attr_i$ and $\theta > 0$ be the threshold, the classifier f is defined as follows:

$$f(a,b) = \begin{cases} match & \text{if } \sum_{i=1}^{m} w_i \cdot sim(a.Attr_i, b.Attr_i) \geq \theta \\ nonmatch & otherwise \end{cases} \tag{1}$$

where $a \in D^A$ and $b \in D^B$. The goal here is to do the classification in a private, accurate and efficient manner.

Besides Alice and Bob, We also assume a third party in PPRL, Carol, who facilitates the linkage process without learning any record from either A or B. We adopt a common assumption that all parties involved are semi-honest and do not collude with each other.

Private Blocking. Blocking is a process that groups records that are likely to match into blocks based on the values of a subset of attributes (called *blocking attributes*). With blocking, the classifier f only needs to consider the record pairs that are in the same blocks. Given a set of blocks C, a blocking function g maps a record to a subset of C, such that if $g(a) \cap g(b) \neq \emptyset$, then the probability that $f(a,b) = match$ is high. Private blocking is to perform blocking in a privacy-preserving manner.

3.2 Canopy Clustering

Canopy clustering [13] is an efficient technique for clustering large, high-dimensional datasets. The key idea is to perform clustering in two stages, first a rough and quick stage in which a cheap, approximate distance measure is used to divide data into overlapping subsets called canopies, then a more rigorous stage in which expensive distance measurements are made only among points that occur in a common canopy.

The creation of canopies involves the use of two distance thresholds, T_1 and T_2, where $T_1 \geq T_2$. Given a list of data points, a point is first picked off the list as the centre of a canopy and its distances to all other points are measured: all points that are within distance threshold T_1 are put into the canopy, and all points that are within T_2 are removed from the list and thus are excluded from forming new canopies. The process is repeated until the list is empty.

4 Proposed Solution

We base our solution on the use of publicly available reference datasets. The two data owners, Alice and Bob, have access to a public reference dataset D^R and construct their local blocks based on the values of reference attributes (a subset of D^R attributes used for reference)[2]. And Carol, the third party, has no knowledge of the reference dataset being used and constructs global blocks based on the local blocks sent by Alice and Bob. Figure 1 outlines the steps involved in our solution. In this section, we describe each step in detail.

[2] The values used for reference should be in the same domain as the values of the local blocking attributes (e.g. both are surnames).

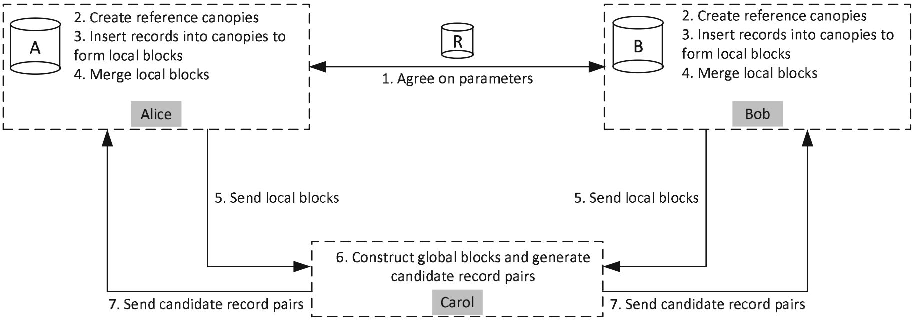

A
2. Create reference canopies
3. Insert records into canopies to form local blocks
4. Merge local blocks
Alice
R
1. Agree on parameters
B
2. Create reference canopies
3. Insert records into canopies to form local blocks
4. Merge local blocks
Bob
5. Send local blocks
5. Send local blocks
6. Construct global blocks and generate candidate record pairs
Carol
7. Send candidate record pairs
7. Send candidate record pairs

Fig. 1. Overview of the protocol.

4.1 Canopy Construction

Before blocking starts, Alice and Bob agree on a set of parameters (step 1 in Fig. 1): the reference dataset D^R to be used; the reference attributes; the blocking attributes; the parameters for canopy clustering including two distance thresholds T_1 and T_2 ($T_1 \geq T_2$), the length of q-grams q, and a random seed for selecting the same set of reference values as canopy centres; the level of approximation ϵ (≥ 0) for finding blocks within a certain distance range; and the merging size k for determining whether a block needs to be merged with other blocks.

After agreeing on the parameters to be used, Alice and Bob each cluster D^R based on the reference attribute values using canopy clustering (step 2). A random record in D^R is first selected as the centre of a canopy, then its distances to all other records are measured. If a record is within distance threshold T_1, it is considered to be part of the canopy; if the record is further within distance threshold T_2, it is excluded from the remaining process and cannot be part of other canopies. The process is repeated until no records are left out for clustering. To measure the distance between two records r_1 and r_2, the Jaccard distance measure is employed:

$$dist(r_1, r_2) = 1 - \frac{|(qg(r_1) \cap qg(r_2)|}{|qg(r_1)| + |qg(r_2)| - |qg(r_1) \cap qg(r_2)|} \tag{2}$$

where $qg(r_i)$ is the q-gram representation of the reference attribute values of r_i. An inverted index data structure can be used to speed up the process, where the keys are q-grams and the values are the set of records whose reference attribute values contain the q-gram.

The process results in a set of overlapping canopies. Each canopy is identified by a record ID in D^R (corresponding to the canopy centre). As Alice and Bob use the same parameters for canopy construction, they have the same set of canopies constructed.

Algorithm 1. Merging blocks based on canopy clustering

```
Input:
- $C$: set of initial blocks
- $k$: merging size
- $T_1$, $T_2$: two distance thresholds
Output:
- $C'$: set of merged blocks
while $C \neq \emptyset$ do
    $c_x = C.next()$
    $c_{xy} = c_x$
    while $(|c_{xy}| < k)$ and $(c_y = C.next())$ do
        $d_{xy} = dist(c_{xy}, c_y)$
        if $d_{xy} \leq T_1$ then
            $c_{xy} = c_{xy} + c_y$
            if $d_{xy} \leq T_2$ then
                $C.remove(c_y)$
    $C.remove(c_x)$
    $C'.add(c_{xy})$
```

4.2 Block Assignment

Alice and Bob then assign their local records to respective canopies to form local blocks (step 3). A redundant block assignment strategy is employed, which ensures that no records are missed out, and if two records are similar, there is a high chance for them to be assigned in the same block. Specifically, for each record, its distance to each canopy, i.e. the Jaccard distance between the record's blocking attribute values and the canopy centre's reference attribute values, is measured. Based on the distances measured, there are three cases: (1) If the distance of a record to a canopy, d, is within T_2 (i.e. $d \leq T_2$), the record is assigned to all such canopies; (2) If no such canopies exist in (1), the record is then assigned to those canopies to which the distance is within T_1 (i.e. $T_2 < d \leq T_1$); (3) Finally, if all distances are larger than T_1 (i.e. $d > T_1$), the record is assigned to $(1+\epsilon)$-approximate nearest canopies, where $\epsilon \geq 0$. In this way, each local record is assigned to at least a canopy, and the local records contained in a canopy form a local block which can be identified by the canopy ID (i.e. a record ID in D^R).

4.3 Block Merging

Each block is next checked; if its size is less than k, it is merged with other blocks at the same data source (step 4). This enables some adjustment of blocks. A merged block is identified by an ID in the form of "r_u-. . .-r_v", where $r_{u(v)}$ is the ID of the block involved in the merging (recall that a block is initially identified by the canopy ID, i.e. a record ID in D^R). For example, if a new block is formed by merging blocks r_1, r_2 and r_3, then its block ID will be r_1-r_2-r_3.

The merging process follows the same idea of canopy clustering. Algorithm 1 shows the merging process. Basically, block c_x is merged with block c_y if the size of c_x is less than k, and the distance between c_x and c_y is less than or equal to T_1 (lines 4–7); if the distance is further less than or equal to T_2, b_y is excluded from the remaining merging process and cannot be merged with other blocks (lines 8–9). This process again introduces redundancy among blocks so that similar records can be assigned to the same blocks.

To measure the distance between two blocks, each block is summarised by a set of q-grams that occur frequently in its records' blocking attribute values. In particular, the q-grams of the blocking attribute values are ordered firstly by frequency (in the descending order) and secondly by alphabet (in the ascending order), and the first l q-grams are used (in no particular order) to represent the block where l is the average number of q-grams of the blocking attribute values. For example, suppose that a block includes three records whose blocking attribute values are s_1 = "smith", s_2 ="smitch" and s_3 = "smash" respectively. Also suppose that the length of q-grams is 2. Then $q(s_1) = \{\_s, sm, mi, it, th, h\_\}$, $q(s_2) = \{\_s, sm, mi, it, tc, ch, h\_\}$, $q(s_3) = \{\_s, sm, ma, as, sh, h\_\}$, and the block will be represented by $\lfloor \frac{6+7+6}{3} \rfloor$ most frequent bigrams, i.e. $\{\_s$, sm, mi, it, as, $h\_\}$. After the summarisation, the Jaccard distance (2) is used to compute distances between blocks based on their q-gram representations.

4.4 Candidate Generation

The merged blocks at each data source are then sent to Carol, with each record ID being encrypted (step 5). Based on the blocks from Alice and Bob, Carol generates candidate record pairs (step 6). Algorithm 2 shows the generation process. For each block from Alice, c^A, Carol finds all blocks from Bob, C', such that the block ID of each block in C' shares with the block ID of c^A at least one canopy ID or one reference record ID (lines 1–5). Recall that a block ID is in the form of "r_u-…-r_v", where $r_{u(v)}$ is a canopy ID or a reference record ID. Thus, $C' = \{c^B \mid R(c^A) \cap R(c^B) \neq \phi\}$, where $R(c^i)$ refers to the set of canopy IDs included in the block ID of c^i. Based on that, Carol generates candidate record pairs with records from c^A and records from each c^B in C' (lines 6–9). For example, suppose $c^A = \{a_1, a_2\}$, $c^B = \{b_1, b_3\}$, and their block IDs are r_1-r_5 and r_1-r_3-r_4 respectively. Since r_1 appears in both c^A and c^B's block IDs, records in c^A need to be compared with records in c^B, i.e. the candidate record pairs generated are (a_1, b_1), (a_1, b_3), (a_2, b_1) and (a_2, b_3).

Finally, Carol sends the record IDs of the generated candidate record pairs back to Alice and Bob (step 7), who can then use existing PPRL algorithms (e.g. Bloom Filters [17]) for matching record pairs in global blocks.

Algorithm 2. Generation of candidate record pairs

```
Input:
- $C^A$: set of blocks from Alice
- $C^B$: set of blocks from Bob
Output:
- $RP$: set of candidate record pairs
for $c^A \in C^A$ do
    $C' = \{\}$
    for $c^B \in C^B$ do
        if $R(c^A) \cap R(c^B) \neq \phi$ then
            $C'.add(c^B)$
    for $c^B \in C'$ do
        for $a \in c^A$ do
            for $b \in c^B$ do
                $RP.add((a, b))$
```

5 Privacy Analysis

As mentioned in Sect. 3, we take a common assumption that all parties involved are semi-honest without collusion. Thus, Alice, Bob and Carol will follow the protocol honestly, but may try to learn private information based on the information they receive during the execution without collusion. Under such a setting, we discuss what kind of information can be leaked between them.

Alice and **Bob** do not have direct access to each other's data. They only need to agree on the reference dataset and the parameters to be used, and then perform blocking on each's data independently without any further communication between them. As a result, none of them can perform frequency analysis on the other's data.

Carol receives blocks from Alice and Bob, and learns the number of blocks from each and the cardinality of each block. However, it is difficult for Carol to perform a frequency attack as she does not know the reference dataset that was used and how blocks were generated. The merging of blocks further makes it harder for the attack.

6 Experimental Evaluation

We experimentally evaluated our approach in terms of its blocking efficiency and effectiveness. We generated our datasets as follows. We first used Febrl [4] to generate a dataset consisting of 1M original records. We then sampled the dataset and generated datasets of 4 different sizes, i.e. 1K, 5K, 10K and 50K. For each size, one pair of datasets were generated which have 25% of their records in common respectively. Thus, all together we have four pairs of datasets. For each pair, we corrupted the common records of one dataset by introducing one

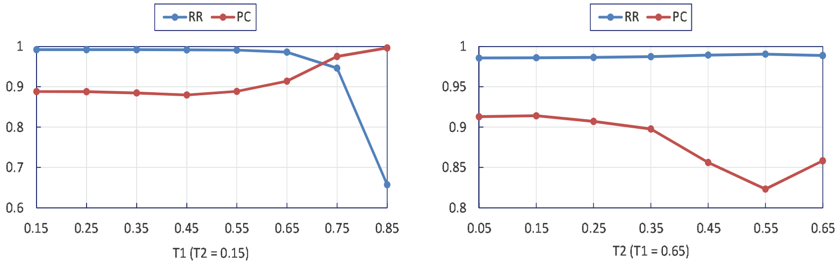


Fig. 2. Effect of distance thresholds T_1 or T_2 on RR and PC.

typographical error to each of their ‘Surname’ values (we chose the ‘Surname’ attribute as the blocking attribute). The error can be a character insertion, deletion or substitution, which was chosen at equal probability.

The reference dataset was generated by sampling 5K records from the combined longitudinal North Carolina Voter Registration (NCVR) dataset [5]. We used the ‘last_name’ attribute as the reference attribute. We first evaluated the performance of our approach and then compared it with the sorted neighborhood based approaches. As discussed in Sect. 2, among existing approaches, the k-NN approach [11] and the sorted neighborhood based approaches [18] are most relevant to our work. In [18], two sorted neighborhood based approaches were proposed, SNC-Sim and SNC-Size. Both approaches were shown to have better performance than the k-NN approach, and also SNC-Sim was shown to have better performance than SNC-Size. Thus, in our study, we compared our approach with SNC-Sim only. The measures we used include the time required for blocking, the time for both blocking and matching (we used Bloom Filters [17] for matching), the fraction of record pairs that are removed by blocking, i.e. reduction ratio (RR), and the fraction of true matching record pairs generated by blocking that are included in the candidate record pairs, i.e. pair completeness (PC). We conducted all experiments on a Dell laptop with 64bit Intel(R) Core(TM) i7-6600U (2.6 GHz) CPU and 8 GB RAM.

By default, we conducted experiments on the 10K dataset pair, and set the approximation level ϵ to 0.2 and the merging size k to 20. We fixed the length of q-grams to 2. The values of T_1 and T_2 relate to the size and overlap of blocks. High values of T_1 increase the size of blocks, and low values of T_2 increase the overlap. Both affect RR and PC in some way by having more candidate record pairs generated or more true matches included in the record pairs. As shown in Fig. 2, in general, RR decreases with T_1 but increases with T_2, and PC increases with T_1 but decreases with T_2; a good balance between RR and PC can be achieved when $T_1 = 0.65$ and $T_2 = 0.15$, which provides more than 98% savings in the number of comparisons with a cost of about 8% loss in the identification of true matches. Thus, in the following experiments, we set T_1 and T_2 to 0.65 and 0.15 respectively.

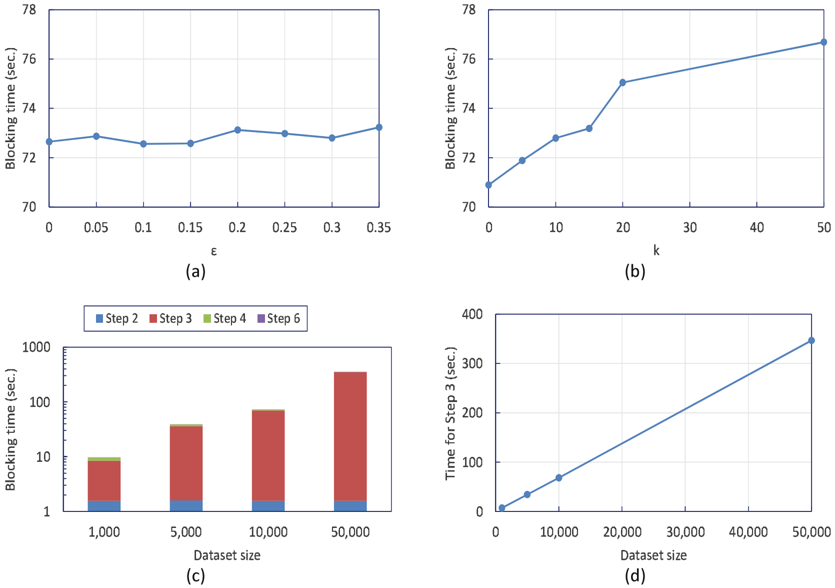


Fig. 3. Effect of (a) approximation level ϵ, (b) merging size k, and (c–d) dataset size on the blocking time.

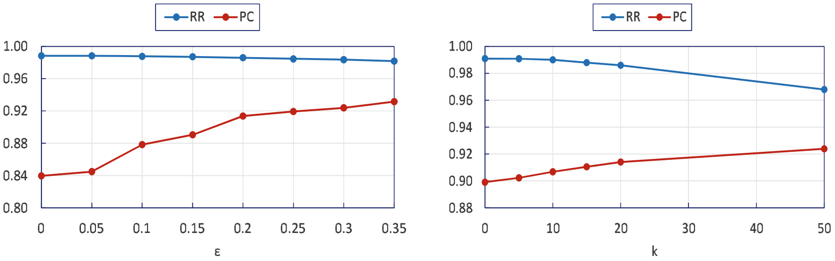


Fig. 4. Effect of approximation level ϵ or merging size k on RR and PC.

For SNC-Sim (abbreviated as SNC), we used the same parameter settings in [18]: $k = 100$, and $s_t = 0.9$. For matching with Bloom Filters, we set the length of bloom filters to 1024 and the number of hashing functions to 30.

Figure 3(a) and (b) show the effect of ϵ and k on the blocking time respectively. As shown, the time required for blocking changes little on varying ϵ, and increases slightly with k. Figure 3(c) shows the breakdown of the blocking time, including the time spent by the data owners for step 2 (canopy construction), step 3 (block assignment) and step 4 (block merging), and the time by the third party for step 6 (candidate generation), which indicates that the blocking time is dominated by the time for step 3 and the time for other steps is almost negligible for even large dataset sizes. Figure 3(d) further shows that the time for step 3 scales linearly to the dataset size.

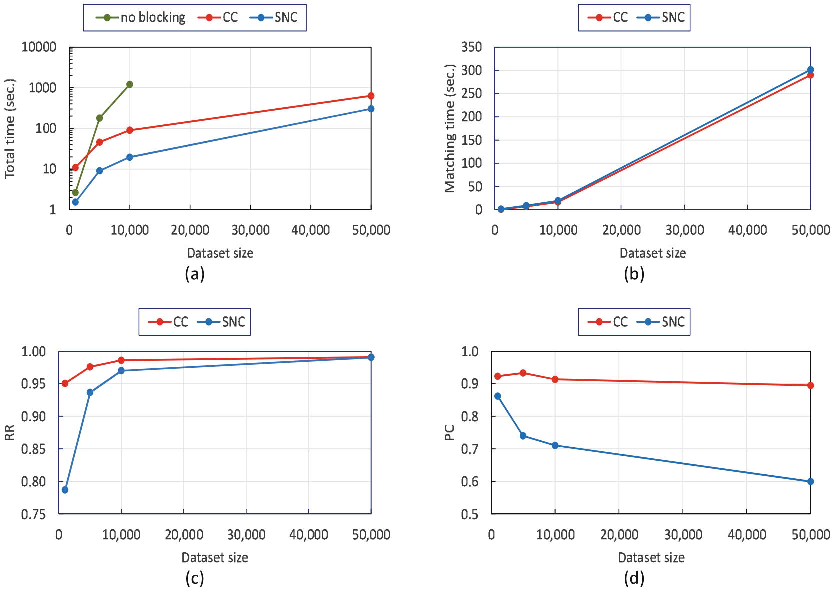


Fig. 5. Comparison with the SNC approach: (a) total time, (b) matching time, (c) RR and (d) PC.

Figure 4 shows the effect of ϵ or k on RR and PC. With larger ϵ, PC improves greatly as there is more overlap between blocks and thus more true matches are potentially included in the record pairs generated. This, however, is at the cost of slight decrease in RR as the blocks created are of larger size and thus more candidate record pairs are generated. A similar effect of k, but to a less extent, on RR and PC is also observed in the figure.

Figure 5(a) shows the total time required for the linkage process with our approach (CM) and with the SNC approach. As shown, for datasets larger than 1K, linkage with both approaches is much faster than that without blocking; in the case with our approach, nearly one magnitude less time is required (we were unable to conduct the experiment for the no-blocking linkage on the 50K dataset pair due to its memory requirements). The time difference between the linkage with our approach and with the SNC approach, which becomes smaller with the dataset size, is mainly due to the difference in the blocking time. Nevertheless, our approach still scales linearly to the dataset size as mentioned earlier. Also, with our approach, less (though not much) time is required for matching than with the SNC approach (Fig. 5(b)). This is because less candidate record pairs are generated for comparison using our approach. As shown in Fig. 5(c), our approach achieves a higher RR than the SNC approach; moreover, it has a much higher PC (Fig. 5(d)). Thus, a better quality of blocking is provided by our approach, largely attributed to its redundant, similarity-based record-to-block assignments.

7 Conclusion

In this paper, we proposed a novel private blocking approach based on canopy clustering. As future work, we plan to investigate the use of differential privacy [7] in our work to provide strong privacy guarantees.

References

1. Agrawal, R., Evfimievski, A., Srikant, R.: Information sharing across private databases. In: Proceedings of SIGMOD (2003)
2. Al-Lawati, A., Lee, D., McDaniel, P.: Blocking-aware private record linkage. In: Proceedings of IQIS (2005)
3. Bonomi, L., Xiong, L., Chen, R., Fung, B.C.M.: Frequent grams based embedding for privacy preserving record linkage. In: Proceedings of CIKM (2012)
4. Christen, P.: Febrl – an open source data cleaning, deduplication and record linkage system with a graphical user interface. In: Proceedings of SIGKDD (2008)
5. Christen, P.: Preparation of a real temporal voter data set for record linkage and duplicate detection research. Technical report, ANU (2014)
6. Durham, E.: A framework for accurate, efficient private record linkage. Ph.D. thesis, Vanderbilt University (2012)
7. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006, Part II. LNCS, vol. 4052, pp. 1–12. Springer, Heidelberg (2006). https://doi.org/10.1007/11787006_1
8. Han, S., Shen, D., Nie, T., Kou, Y., Yu, G.: Scalable private blocking technique for privacy-preserving record linkage. In: Li, F., Shim, K., Zheng, K., Liu, G. (eds.) APWeb 2016, Part II. LNCS, vol. 9932, pp. 201–213. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45817-5_16
9. Inan, A., Kantarcioglu, M., Bertino, E., Scannapieco, M.: A hybrid approach to private record linkage. In: Proceedings of ICDE (2008)
10. Karakasidis, A., Koloniari, G., Verykios, V.S.: Scalable blocking for privacy preserving record linkage. In: Proceedings of SIGKDD (2015)
11. Karakasidis, A., Verykios, V.S.: Reference table based K-anonymous private blocking. In: Proceedings of SAC (2012)
12. Kuzu, M., Kantarcioglu, M., Inan, A., Bertino, E., Durham, E., Malin, B.: Efficient privacy-aware record integration. In: Proceedings of EDBT (2013)
13. McCallum, A., Nigam, K., Ungar, L.H.: Efficient clustering of high-dimensional data sets with application to reference matching. In: Proceedings of SIGKDD (2000)
14. Ranbaduge, T., Vatsalan, D., Christen, P.: Tree based scalable indexing for multi-party privacy-preserving record linkage. In: Proceedings of AusDM (2014)
15. Ranbaduge, T., Vatsalan, D., Christen, P.: Clustering-based scalable indexing for multi-party privacy-preserving record linkage. In: Cao, T., Lim, E.-P., Zhou, Z.-H., Ho, T.-B., Cheung, D., Motoda, H. (eds.) PAKDD 2015, Part II. LNCS (LNAI), vol. 9078, pp. 549–561. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-18032-8_43
16. Ranbaduge, T., Vatsalan, D., Christen, P., Verykios, V.: Hashing-based distributed multi-party blocking for privacy-preserving record linkage. In: Bailey, J., Khan, L., Washio, T., Dobbie, G., Huang, J.Z., Wang, R. (eds.) PAKDD 2016, Part II. LNCS (LNAI), vol. 9652, pp. 415–427. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-31750-2_33

17. Schnell, R., Bachteler, T., Reiher, J.: Privacy-preserving record linkage using bloom filters. BMC Med. Inform. Decis. Making **9**, 41 (2009)
18. Vatsalan, D., Christen, P.: Sorted nearest neighborhood clustering for efficient private blocking. In: Pei, J., Tseng, V.S., Cao, L., Motoda, H., Xu, G. (eds.) PAKDD 2013, Part II. LNCS (LNAI), vol. 7819, pp. 341–352. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37456-2_29
19. Vatsalan, D., Christen, P., Verykios, V.: Efficient two-party private blocking based on sorted nearest neighborhood clustering. In: Proceedings of CIKM (2013)

Community Aware Personalized Hashtag Recommendation in Social Networks

Areej Alsini[1,2](✉), Amitava Datta[1], Du Q. Huynh[1], and Jianxin Li[1]

[1] Department of Computer Science and Software Engineering,
The University of Western Australia, Crawley, WA 6009, Australia
areej.alsini@research.uwa.edu.au,
{amitava.datta,du.huynh,jianxin.li}@uwa.edu.au

[2] Department of Computer Science, Umm Al-Qura University, Makkah, Saudi Arabia
aosini@uqu.edu.sa

Abstract. In the literature of social networks research, community detection algorithms and hashtag recommendation models have been studied extensively but treated separately. Community detection algorithms study the inter-connection between users based on the social structure of the network. Hashtag recommendation models suggest useful hashtags to the users while they are typing in their tweets. In this paper, we aim to bridge the gap between these two problems and consider them as inter-dependent. We propose a new hashtag recommendation model which predicts the top-y hashtags to the user based on a hierarchical level of feature extraction over communities, users, tweets and hashtags. Our model detects two pools of users: in the first level, users are detected using their topology-based connections; in the second level, users are detected based on the similarity of the topics of the tweets they previously posted. Our hashtag recommendation model finds influential users, reweighs their tweets, searches for the top-n similar tweets from the tweets pool of users who are socially and topically related. All hashtags are then extracted, ranked and the top-y are recommended. Our model shows better performance of the recommended hashtags than when considering the topology-based connections only.

Keywords: Social networks · Twitter · Hashtag recommendation · Community detection · Topics model

1 Introduction

Social networks are connections built between individuals who are family members, friends, colleagues or people with similar interests via social media platforms. Topology-based community detection algorithms explore the structure of a social network. They address the interrelationships between users through their following relations, mentions, replies or retweets. Topic-based community detection algorithms group users by detecting their similarity based on their topics [15]. Ding et al. [4] studied the topology-based versus the topic-based

R. Islam et al. (Eds.): AusDM 2018, CCIS 996, pp. 216–227, 2019.
https://doi.org/10.1007/978-981-13-6661-1_17

community detection algorithms on a co-authorship network. They reported that using purely topology-based community detection algorithms is insufficient as they lack semantic characteristics.

Twitter hashtag recommendation system is a predictive model which recommends the top-y hashtags to the user [5,6,8,9,12,19,23,24]. All these models have been tested on a large data and have proven to be effective. A piece of previous work [1] shed the light on a hashtag recommendation system on detected communities. It was found that the social factor is the most influential factor in hashtag recommendation and the chosen algorithm in detecting communities affects the performance of the recommended hashtags. However, the hashtag recommendation performance was tested only on communities detected using topology-based algorithms. In addition, none of the previous research studied the importance of influential users in hashtag recommendation.

In this paper, we propose a personalized hashtag recommendation system which comprises a hierarchical level of common feature extraction stages taking into account the topology structure between users and the topics discussed by the communities. The importance of the influential user in hashtag recommendation is also considered. Our model, which we name *Topology-Topic Hashtag Recommendation* (TTHR) is a hybrid hashtag recommendation system as it integrates the content-based and user-based methods.

The contributions of this paper are two-fold: Firstly, we propose a novel hashtag recommendation pooling algorithm which combines tweets of users who are socially and topically related. Secondly, our algorithm can automatically identify influential users and re-weigh their tweets.

The paper is organized as follows: Section 2 discusses previous works relevant to our research. Section 3 describes our proposed model TTHR. Section 4 explains the dataset, the conducted experiments, the experimental results and discussions. Section 5 concludes the paper and outlines our future work.

2 Related Work

Topic Models. Topic models such as Latent Dirichlet Allocation (LDA) [3] is an unsupervised machine learning algorithm which finds topics and semantic patterns from texts. Although LDA has proven to be a very valuable technique for large corpus [3], Yan et al. [21] and Yin et al. [22] show that using the LDA model is not efficient on short texts. The short texts generated from the social networks, such as Twitter, have a much more complicated problem. The terms in tweets are noisy and the patterns of word co-occurrence are very sparse [21]. In order to enhance the quality of the topics generated using LDA and overcome the problem of the tweets' short texts, different schemes which aggregate set of tweets are proposed [2,13,20]. The best topic quality was generated when the LDA model was trained using the Author-based scheme over the Hashtag-based scheme and Conversation scheme [2]. Author-based scheme aggregates all tweets posted by a single user and treats them as a single document [13]. Usually it is applied at the pre-processing phase prior to training the model.

Community Detection Algorithms. The Clique Percolation Method (CPM) [14] is a topology-based community detection algorithm. Using CPM, densely connected users based on their social relationships are grouped to form cliques of size k users and if $k-1$ users are members of two cliques then these two cliques are also grouped forming an overlapped community. Other works integrated both topological and topical features to find semantic communities [10,26]. However, Ding [4] showed that different sets of topics within each community can be detected using topology-based algorithms and sub-topological groups are found to be present within the communities using the topical-based community detection algorithms.

Identifying Influential Users. An *influencer* is a person who is well connected with many other people and can influence their thoughts and decision making. The identification of influential users was used in marketing [16] and studying the propagation of information [7]. The *in-degree* measure is used in Twitter to interpret the user influence which means the number of followers a user may have [20]. Riquelme et al. [17] studied several methods of identifying influential users based on activity, popularity, influence measures and topics. However, the content quality of tweets was not considered in any of these metrics.

Hashtag Recommendation Systems. Previous research proposed general and personalized hashtag recommendation systems which are content-based, user-based or hybrid. Content-based hashtag recommendation systems mostly focus on the tweet content. Research in this category used text similarity methods [24], text classification [5,12] and topic models [6,25]. Topic-based hashtag recommendation models used topics model to discover topics from a large collection of text and used the generated topics to annotate documents accordingly [6,25]. User-based hashtag recommendation systems mostly focus on grouping similar users through *user characteristics* [9,23,25] or *relations* [8] or a combination of them [19]. User characteristics can be the user's historical hashtags usage, favourite topics or tweeting behaviour. User relations are the connections with other people. User-based attributes was always integrated with other factors in hashtag recommendation systems. Hybrid hashtag recommendation systems combine both the content and user-based methods. Kywe et al. [9] proposed a personalized hashtag recommendation system based on the similarity between tweet contents and users. The historical hashtag preference was the attribute of measuring user similarity. Yu et al. [23] incorporated hashtag textual information, user behavior (interactions) through time in their personalized hashtag recommendation model. The *relevance* between a user's interest and his/her hashtags in this model was weighted based on the number of interactions performed using that hashtags where *interactions* meant the number of "create", "comment", "favourite", "retweet", and "add friend". The BLL model [8] integrated text similarity, social relations and time. It identified hashtags used by the user and his/her followee in a certain time. Out of the mentioned hashtag recommendation models, a single work [1] addressed the impact of the community detection algorithms on hashtag recommendation. However, it was studied on topology-based community detection algorithms.

3 Proposed Model

Our TTHR model is composed of three levels. The two base levels of TTHR detect two pools of users while the top level is the hashtag recommendation system. TTHR is shown in Fig. 1 and summarized as follows:

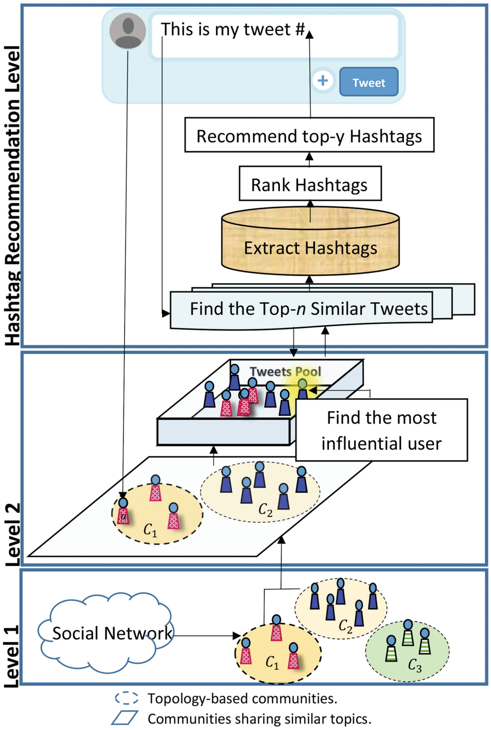

Hashtag Recommendation Level
This is my tweet #
+
Tweet
Recommend top-y Hashtags
Rank Hashtags
Extract Hashtags
Find the Top-n Similar Tweets
Tweets Pool
Find the most influential user
C_1
C_2
Level 2
Level 1
Social Network
C_1
C_2
C_3
Topology-based communities.
Communities sharing similar topics.

Fig. 1. Structure of our Topology-Topic Hashtag Recommendation Model (TTHR).

- Level 1: Topology-based communities. This level detects communities based on the structure of the network considering the following links.
- Level 2: Semantic feature extraction. This level discovers the topics of interests discussed by each of the detected communities exposing *hashtagged tweets* which are tweets containing one or more hashtags. Based on the selected semantic features, communities with strong content similarity are grouped and their tweets are pooled. Tweets of influential users are reweighted. Pooled tweets are used in hashtag recommendation.
- Hashtag recommendation level: Hashtags of the top-n similar tweets are extracted from the tweets pool, ranked based on popularity and relevance, and the top-y candidate hashtags ($y < n$) are recommended to the user.

Algorithm 1. Community-Topics Relationship

```
INPUT: $\mathscr{C}$: a set of communities,
       $N_{\mathscr{T}}$ : number of topics,
       $N_{\mathscr{W}}$ : number of terms per topic.
OUTPUT: $CTM$: a Community-Topics Map.

Pre-process all user profiles $u_j$;
Initialize: $CTM \leftarrow \phi$;
for $C_i \in \mathscr{C}$ do
    Generate the training and testing corpus by considering every $u_j$;
    Train the LDA model on the training corpus;
    Generate the topics set $\mathscr{T}$, where each $T_i \in \mathscr{T}$ has $N_{\mathscr{W}}$ terms;
    Store $N_{\mathscr{M}}$ topic-terms in $s_i$.
    Add $(i,\ s_i)$ to $CTM$;
end for
return $CTM$;
```

3.1 Topology-Based Communities

A social network $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ is an undirected graph where $\mathscr{V}$ is a set of nodes representing users and $\mathscr{E}$ is a set of edges connecting pair of users based on the following relationship. Let $\mathscr{C} = \{C_1, C_2, ...\}$ be the set of communities detected from $\mathscr{G}$ using a community detection algorithm $\mathscr{A}$ and let $N_{\mathscr{C}} = |\mathscr{C}|$ denote the number of communities in $\mathscr{C}$. Every community $C_i = \{u_1, u_2, ...\}$ contains a set of user profiles where every user profile consists of a set of hashtagged tweets.

3.2 Semantic Feature Extraction

The Community-Topics and the Inter-Community relationships are then inferred from the detected communities $\mathscr{C}$ as described below and in Algorithms 1 and 2, respectively.

Community-Topics Relationship. The LDA model is used to infer the topics discussed by each of the detected communities $C_i \in \mathscr{C}$. The LDA implemented in this paper uses the Author-Based scheme [13] which considers every user's profile u_j as a document to overcome the problem of short texts. Every u_j is represented as a probability distribution over a set of topics $\mathscr{T} = \{T_1, T_2, ...\}$, where $N_{\mathscr{T}}$ denotes the number of topics. The number of terms in all topics is $N_{\mathscr{W}}$. In the same context, it is common that terms of the same topic co-occur. Applying the LDA model to each of the communities generates completely different topics and different topic structure. For this reason, Community-Topics Map (CTM) is constructed by creating a community semantic profile s_i for each community C_i, where s contains $N_{\mathscr{M}}$ most probable topic-terms of all topics. The input to the LDA model is a set of communities $\mathscr{C}$. The number of topics $N_{\mathscr{T}}$ and the number of the most probable words per topic $N_{\mathscr{W}}$ are specified in advance. After the pre-processing and division of the dataset into training and testing sets are performed (described in Sect. 4), the LDA model is trained to learn the relationships between words and topics. The output of this algorithm is a CTM where each community ID i is attached to its community semantic profile s_i. Algorithm 1 explains how the Community-Topics relationship is built.

Inter-community Semantic Relationship. The generated CTM is used here to address the semantic relationships between communities. Algorithm 2 takes the CTM as input. The TF-IDF algorithm [18] is used to weigh a topic-term t in any community semantic document s, where $\mathscr{S}$ denotes the collection of semantic documents of all communities. The weight $w_{t,s}$ of topic-term t in document s is calculated as follows:

$$w_{t,s} = \mathrm{TF}_{t,s}.\mathrm{IDF}_{t,\mathscr{S}}, \tag{1}$$

where

$$\mathrm{IDF}_{t,\mathscr{S}} = 1 + \log\left(\frac{|\mathscr{S}|}{1 + \mathrm{DF}_t}\right). \tag{2}$$

$\mathrm{TF}_{t,s}$ is the frequency of a topic-term t in the community semantic profile s. $\mathrm{IDF}_{t,\mathscr{S}}$ is the inverse document frequency of a topic-term t, where $|\mathscr{S}|$ is the count of the total number of the community semantic profiles in the dataset and DF_t is the number of the community semantic profiles that contain the topic-term t. One is added to the DF to avoid division by zero and to the final result to avoid ignoring topic-terms with zero IDF. TF gives higher weight to the topic-term if it is more frequently used in many topics by a community and a higher weight is assigned to the topic-term by IDF if the topic-term was rarely used by all other communities. Then, the cosine similarity function is used to calculate the degree of similarity between communities where $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$ are $N_{\mathscr{M}}$-dimensional vectors over the weights of the topic-terms in $\mathscr{W}$. The variable Score in Algorithm 2 represents the strength of the connection between any two communities. If Score is greater than a threshold ξ, then the two communities are grouped. Communities close to the community i are added to the Inter-Community Map (ICM).

3.3 Hashtag Recommendation

For every community $C_i \in \mathscr{C}$, let $\mathscr{X} = (\mathscr{D}, \mathscr{U})$ be a training set where every hashtagged tweet $d \in \mathscr{D}$ is written by a user $u \in \mathscr{U}$. Let $\mathscr{Z} = (\mathscr{Q}, \mathscr{U}, \mathscr{R})$ be a testing set containing query tweets with no hashtags. Every query $q \in \mathscr{Q}$ is treated as an active tweet typed in by a user $u \in \mathscr{U}$ and has a ground truth hashtag $r \in \mathscr{R}$. In this paper, an influential user is the user who is a member in more than a community when grouping by topics.

Algorithm 3 explains the hashtag recommendation of the TTHR model. For every query q written by a user u, the set of top-n similar tweets is extracted from the tweets pool $\mathscr{P}$. The tweets pool $\mathscr{P}$ is formed by pooling the tweets of social and topical users who are related to u. If u is influential, we increase the importance of his tweets by duplicating them once in the tweets pool $\mathscr{P}$. The weight $w_{t,p}$ of term t in a tweet p is calculated the same way as $w_{t,s}$ given in

Algorithm 2. Inter-Community Relationship

```
INPUT: $CTM$: a Community-Topics Map containing $(i, s_i)$,
       $\xi$: a threshold that measures the degree of similarity between communities.
OUTPUT: $ICM$: Inter-Community Map, the sets of the most similar communities.

Initialize: $ICM \leftarrow \phi$;
Weigh the topic-terms of all $s$ using TF-IDF, $\mathscr{W}$;
for $\boldsymbol{w_i} \in \mathscr{W}$ do
    $ICM \leftarrow i$
    Calculate Score$(\boldsymbol{w_i}, \boldsymbol{w_j}) = \frac{\boldsymbol{w_i}.\boldsymbol{w_j}}{|\boldsymbol{w_i}|.|\boldsymbol{w_j}|}$;
    if Score$(\boldsymbol{w_i}, \boldsymbol{w_j}) > \xi$ then
        Join $j$ to $ICM$;
    end if
end for
return $ICM$;
```

Algorithm 3. Hashtag Recommendation

```
INPUT: $\mathscr{Z} = (\mathscr{Q}, \mathscr{U}, \mathscr{R})$: a testing set,
       $\mathscr{X} = (\mathscr{D}, \mathscr{U})$: a training set,
       $n$: the number of similar tweets,
       $\nu$: threshold for degree of tweet similarity,
       $ICM$: the sets of the most similar communities.
OUTPUT: The top-$y$ recommended hashtags.

1:  for $\mathscr{Z}_{q,u,r}$ do
2:      if $u$ in $ICM_i$ then
3:          Create the tweets pool $\mathscr{P}$ for $u$;
4:      end if
5:      if $u$ is influential AND q not in $\mathscr{D}$ then
6:          Duplicate tweets written by $u$ one time and store into $\mathscr{P}$;
7:      end if
8:      Weigh the tweet-terms of all $p$ in $\mathscr{P}$ using TF-IDF (Eq. 1) and store in $\boldsymbol{p}$;
9:      Weigh the tweet-terms of $q$ using TF-IDF (Eq. 1) and store in $\boldsymbol{q}$;
10:     Calculate Score$(\boldsymbol{q}, \boldsymbol{p}) = \frac{\boldsymbol{q}.\boldsymbol{p}}{|\boldsymbol{q}|.|\boldsymbol{p}|}$;
11:     if Score$(\boldsymbol{q}, \boldsymbol{p}) > \nu$ then
12:         Join $p$;
13:     end if
14:     Retrieve the top-$n$ similar tweets to $q$ from $\mathscr{P}$;
15:     Extract all hashtags from the set of top-$n$ retrieved tweets;
16:     Rank hashtags by popularity;
17:     Extract the hashtags by relevance from the tweet with the highest similarity
        score;
18:     Combine candidate hashtags from step 16 and 17;
19: end for
20: return top-$y$ candidate hashtags to be recommended to $u$;
```

Eq. (1), except that the weight is now computed on a given tweet p instead of a document s.

$\mathrm{TF}_{t,p}$ is the frequency of the term t in the tweet p. The entity $|\mathscr{P}|$ is the total number of tweets in the tweets pool and the variable DF_t denotes the number of tweets that contain t. Then, the similarity score between an active tweet and any other tweet is calculated using the cosine similarity. Following [1], a threshold ν is used to dismiss the dissimilar tweets so only highly similar tweets would be considered in the subsequent steps. From the set of the similar tweets, the top-n similar tweets are retrieved. Candidate hashtags from these tweets are collected. Two ranking methods on the hashtags are used: *popularity* and *relevance*, where hashtag popularity means hashtag frequency and hashtag relevance means hashtags in the tweet with the highest similarity score. Finally, the top-y hashtags obtained from the top-n similar tweets are suggested to a user.

4 Experiments

4.1 Dataset and Pre-processing

We use the *Dataset-UDI-TwitterCrawl-Aug2012* [11] to evaluate our TTHR algorithm. This dataset was collected during the period from 2011 to 2012. Due to hardware constraints, our sub-network consists of 2 million following relationships among almost of 745,262 users. We re-crawled the tweets of users encountered in our experiments to up to 3,000 tweets. We filtered away all the user profiles that have no hashtagged tweets. Duplicated tweets are also deleted. User profiles are then tokenized and all tokens are transformed into lower case letters. Stop words and punctuation are removed. An additional pre-processing step is performed prior to training the LDA. Only the nouns are kept and all plural nouns are singularized.

For our experiments, communities are generated using CPM when $k = 3$. From the sub-network, there are $N_{\mathscr{C}} = 510$ communities. Each community dataset is divided into 80% tweets to be used in the training and 20% for the testing. For the hashtag recommendation testing, all hashtags are removed from the testing tweets and considered as ground truth, i.e., hashtags to be used later in the evaluation.

4.2 Experimental Results

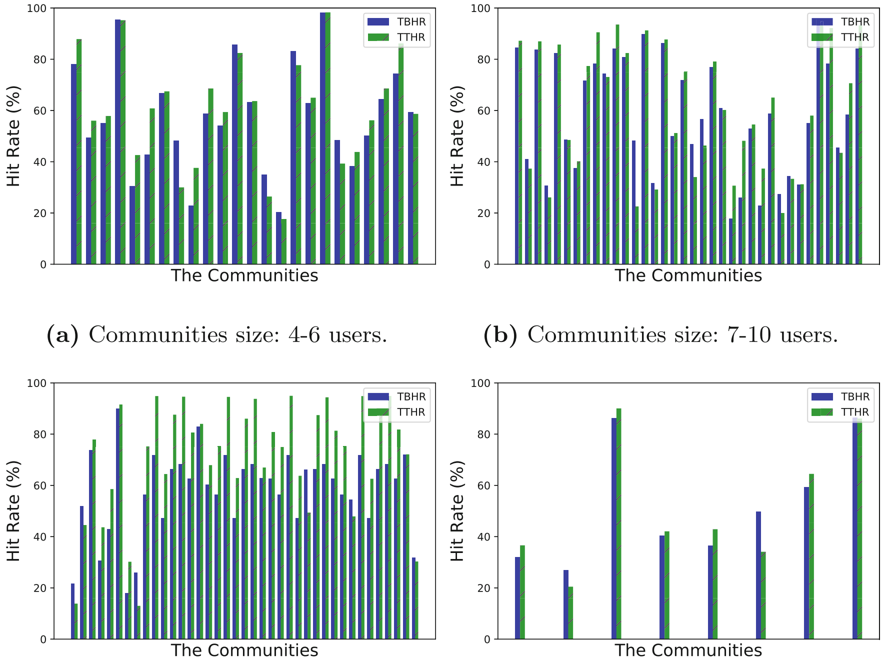
We compare the performance of our Topology-Topic Hashtag Recommendation (TTHR) algorithm with that of the *Topology-Based Hashtag Recommendation* (TBHR) algorithm [1]. In both algorithms, the evaluation measure is the *hit rate* of users on the recommended hashtags. The difference between the two algorithms is that the users in TBHR are socially connected whereas the users in our TTHR are socially and topically connected. Table 1 shows the experimental parameters we use in our experiments. For both experiments, we ran each of the hashtag recommendation model 5 times and the recorded results are the

Table 1. Parameter values used in our experiments.

The retrieved top-n similar tweets	50
Threshold ν for degree of tweet similarity	0.1
The top-y recommended hashtags	10
Number of generated topics $N_{\mathscr{T}}$	1,5,10,15
Number of terms per topic $N_{\mathscr{W}}$	20
The LDA dirichlet priors α and η	0.5
Threshold ξ for degree of community similarity	0.5

averaged score over those five runs. When evaluating TBHR, the hit rate performance ranges between 9.33% and 100%. When evaluating TTHR, there are 42 groups of topical overlapped communities and 111 subgroups within the topical overlapped communities. Through experiments, varying $N_{\mathscr{T}}$ did not affect the generated topic-terms as we are using only nouns when generating the topics. Therefore, we fix $N_{\mathscr{T}}$ to 1 and $N_{\mathscr{W}}$ to 20. Figure 2 is represented by four plots, each demonstrates a comparison between the TBHR and the TTHR hit rate performance on communities with different sizes: 4–6, 7–10, 11–20 and 21–40 users. The blue bars represent the TBHR and the dashed green bars bars represent the TTHR hit rate performance. It is clear that the dashed green bars are the most dominant of the graph. There is an improvement in the hit rate performance of 67% communities, 29% of the communities have a reduced performance and only 4% of the communities have the same performance. In addition, the best performance was shown in plot (c) where the communities size ranges between 11 and 20 users. These results prove the efficiency of our proposed model TTHR.

For deeper analysis, we have studied the hashtagging behaviour of communities considering hashtag popularity. Figure 3 shows the behaviour of some shared hashtags among 6 subcommunities, their corresponding topical community and topical community using TTHR. In the six subcommunities, the hashtag popularity of some hashtags are the same such as 'umci' and 'hcsm'. The user who used these hashtags and who is a member in all these communities is considered an influential user. In addition, the words used by the influential user are mostly dominating the generated topics. On the other hand, hashtags such as 'annarbor', 'poetry' and 'android' are used with different ratios by different communities. The hashtag popularity of the topical community sums up the hashtags used by the six communities but not the hashtags used by the influential user. The hit rate performance does not increase this way. In the topical community using TTHR, the hashtag popularity that are used by the influential users are doubled and the hashtags popularity that are used with different ratios are summed. Our experimental results show that TTHR improves the hit rate performance.



(a) Communities size: 4-6 users.

(b) Communities size: 7-10 users.

(c) Communities size: 11-20 users.

(d) Communities size: 21-40 users.

Fig. 2. A comparison between the TBHR and the TTHR hit rate performance for 111 communities. The blue bars represent TBHR and the dashed green bars represent the TTHR (best viewed in colour). (Color figure online)

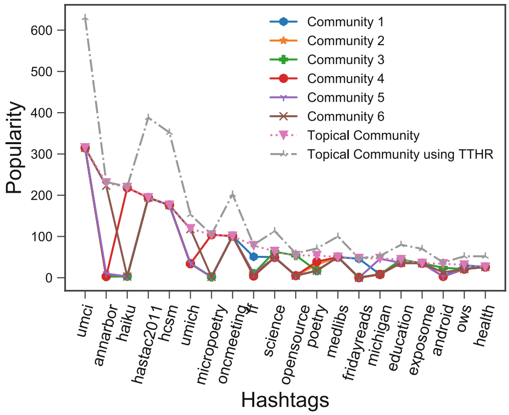


Fig. 3. Community-Based hashtagging behavior.

5 Conclusion and Future Work

The proposed model TTHR integrates community detection algorithms, topics modeling and topics pooling methods in hashtag recommendation. TTHR bridges the gap between the community detection algorithms and the hashtag recommendation models which were treated separately in the literature. This method reveals better understanding of the communities' hashtagging behavior and stresses the importance of addressing influential users in hashtag recommendation. In the future work, we aim to address different features other than topics to find more similarity measures between communities that are suitable for hashtag recommendation.

Acknowledgements. This work was partially supported by the ARC Discovery Project under Grant No. DP160102114 and a Titan Xp GPU from Nvidia Corporation.

References

1. Alsini, A., Datta, A., Li, J., Huynh, D.: Empirical analysis of factors influencing twitter hashtag recommendation on detected communities. In: Cong, G., Peng, W.-C., Zhang, W.E., Li, C., Sun, A. (eds.) ADMA 2017. LNCS (LNAI), vol. 10604, pp. 119–131. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69179-4_9
2. Alvarez-Melis, D., Saveski, M.: Topic modeling in Twitter: aggregating tweets by conversations. In: ICWSM, pp. 519–522. AAAI Press (2016)
3. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
4. Ding, Y.: Community detection: topological vs topical. J. Inf. **5**(4), 498–514 (2011)
5. Dovgopol, R., Nohelty, M.: Twitter Hash Tag Recommendation. CoRR abs/1502.00094 (2015)
6. Godin, F., Slavkovikj, V., De Neve, W., Schrauwen, B., Van de Walle, R.: Using topic models for Twitter hashtag recommendation. In: Proceedings of the 22nd International Conference on World Wide Web, WWW, pp. 593–596. ACM, New York, NY, USA (2013)
7. Guo, R.: Research on information spreading model of social network. In: 2012 Second International Conference on Instrumentation, Measurement, Computer, Communication and Control, pp. 918–921, December 2012
8. Kowald, D., Pujari, S.C., Lex, E.: Temporal effects on hashtag reuse in Twitter: a cognitive-inspired hashtag recommendation approach. In: Proceedings of the 26th International Conference on World Wide Web, WWW 2017, pp. 1401–1410. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland (2017)
9. Kywe, S.M., Hoang, T.-A., Lim, E.-P., Zhu, F.: On recommending hashtags in Twitter networks. In: Aberer, K., Flache, A., Jager, W., Liu, L., Tang, J., Guéret, C. (eds.) SocInfo 2012. LNCS, vol. 7710, pp. 337–350. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-35386-4_25
10. Li, D., et al.: Modeling topic and community structure in social tagging: the TTR-LDA-community model. J. Assoc. Inf. Sci. Technol. **62**(9), 1849–1866 (2011)
11. Li, R., Wang, S., Deng, H., Wang, R., Chang, K.: Towards social user profiling: unified and discriminative influence model for inferring home locations. In: KDD, pp. 1023–1031 (2012)

12. Mazzia, A., Juett, J.: Suggesting hashtags on Twitter. In: EECS 545 Project, Winter Term (2011)
13. Mehrotra, R., Sanner, S., Buntine, W.L., Xie, L.: Improving LDA topic models for microblogs via Tweet pooling and automatic labeling. In: SIGIR, pp. 889–892. ACM (2013)
14. Palla, G.: Uncovering the overlapping community structure of complex networks in nature and society. Nature **435**(814), 814–818 (2005)
15. Reihanian, A., Minaei-Bidgoli, B., Alizadeh, H.: Topic-oriented community detection of rating-based social networks. J. King Saud Univ. Comput. Inf. Sci. **28**(3), 303–310 (2016)
16. Richardson, M., Domingos, P.: Mining knowledge-sharing sites for viral marketing. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2002, pp. 61–70. ACM, New York, NY, USA (2002)
17. Riquelme, F., Cantergiani, P.G.: Measuring user influence on Twitter: a survey. Inf. Process. Manag. **52**(5), 949–975 (2016)
18. Sarkar, D. (ed.): Text Analytics with Python. Apress, Bangalore (2016)
19. Tran, V.C., Hwang, D., Nguyen, N.T.: Hashtag recommendation approach based on content and user characteristics. Cybern. Syst., 1–16 (2018), https://doi.org/10.1080/01969722.2017.1418724
20. Weng, J., Lim, E.P., Jiang, J., He, Q.: Twitterrank: finding topic-sensitive influential Twitterers. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining, WSDM, pp. 261–270. ACM, New York, NY, USA (2010)
21. Yan, X., Guo, J., Lan, Y., Cheng, X.: A biterm topic model for short texts. In: Proceedings of the 22nd International Conference on World Wide Web, WWW, pp. 1445–1456. ACM, New York, NY, USA (2013)
22. Yin, J., Wang, J.: A Dirichlet multinomial mixture model-based approach for short text clustering. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD, pp. 233–242. ACM, New York, NY, USA (2014)
23. Yu, J., Zhu, T.: Combining long-term and short-term user interest for personalized hashtag recommendation. Front. Comput. Sci. **9**(4), 608–622 (2015)
24. Zangerle, E., Gassler, W., Specht, G.: On the impact of text similarity functions on hashtag recommendations in microblogging environments. Soc. Netw. Anal. Min. **3**(4), 889–898 (2013)
25. Zhao, F., Zhu, Y., Jin, H., Yang, L.T.: A personalized hashtag recommendation approach using LDA-based topic model in microblog environment. Future Gener. Comput. Syst. **65**(Supplement C), 196–206 (2016). special Issue on Big Data in the Cloud
26. Zhou, D., Manavoglu, E., Li, J., Giles, C.L., Zha, H.: Probabilistic models for discovering e-Communities. In: Proceedings of the 15th International Conference on World Wide Web, WWW, pp. 173–182. ACM, New York, NY, USA (2006)

Behavioral Analysis of Users for Spammer Detection in a Multiplex Social Network

Tahereh Pourhabibi[1](✉), Yee Ling Boo[1], Kok-Leong Ong[2],
Booi Kam[1], and Xiuzhen Zhang[1]

[1] RMIT University, Melbourne, Australia
{tahereh.pourhabibi,yeeling.boo,
b.kam,xiuzhen.zhang}@rmit.edu.au
[2] Latrobe University, Melbourne, Australia
kok-leong.ong@latrobe.edu.au

Abstract. There are now a growing number of social networking websites with millions of users, creating a fertile ground for "spammers" to abuse opportunities in these websites for their own gain through constant exposure of malicious communications to other users. The variety of interactions afforded by these social networks has resulted in a Multiplex Network of interactions. In these networks, malicious users evade detection by frequently changing the nature of their activities. This makes it challenging to analyse users' interactions to capture anomalous behaviours. In this paper, we aimed to detect spammers in a large time-evolving multiplex social network called Tagged.com. For this purpose, we used four different sets of features: (i) a set of light-weight behavioural features to capture the structural behaviour of users in their neighbourhood network; (ii) a set of bursty features and (iii) sequence-based features for capturing the temporal behaviour of users; and (iv) a set of profile-based features which was used as a side information. In addition, we also employed an unsupervised Laplacian Score based approach for feature selection and space dimensionality reduction. The experimental results showed an accuracy of over 88% in spammer detection with a lower empirical time complexity for feature extraction. Implementing behavioural and bursty features in a relational data management system makes the proposed approach more practical since most of the real-world networks store their data in relational databases.

Keywords: Spammer detection · Multiplex network · Feature extraction · Behavioural feature · Temporal · Laplacian score · Un-supervised feature selection

1 Introduction

The increasing popularity of social networks has brought about the opportunities to collect a huge amount of information about their users such as, their characteristics, habits and friends [1]. This valuable information and ease of access to other users within social networks provide a platform for illegitimate users, called "spammers", to behave maliciously and cause inconvenience to others. Particularly, spammers frequently try to reach new victims by sending unsolicited messages.

R. Islam et al. (Eds.): AusDM 2018, CCIS 996, pp. 228–240, 2019.
https://doi.org/10.1007/978-981-13-6661-1_18

As online social networks provide their users with different types of interactions (e.g., tweets, mention, like, dislike, wink, poke, etc.), they are a specific type of heterogeneous networks with different types of links known as Multiplex Networks [2]. In these networks, spammers are more likely to evade the filtering security measures by frequently changing the nature of their interactions [3].

Many studies, e.g., [4–9], do not consider the intrinsic multiplex nature of human interactions. They tend to investigate users' behaviour in simplex social networks by focusing on one type of activity and the content of their messages to detect and filter spammers. Sophisticated spammers can cleverly manipulate their spam messages to bypass traditional content-based spam filters. In contrast, it is a lot more difficult for spammers to conceal their interactions in a network to avoid detection [2, 10]. For example, [2] is one of the very few studies that analysed spammers in multiplex social networks by analysing the structural characteristics of social networks using graph-based features and used sequence-based features to capture the temporal behaviour of users.

In this study, we developed an approach to detect spammers in a time-evolving multiplex social network based on users' interactions in the network independent of the message contents. We extracted structural features based on the analysis of spammers' behaviour in popular social media such as Twitter. Inspired by gene sequence analysis, we introduced new sequence-based features to analyse users' temporal behaviour. Rather than using a sequence-based feature set to capture changes in users' activities, we used a bursty feature set to capture bursts of spamming activities over time. Like [2], profile-based feature was also included. To control dimensionality with our approach, we exploited an unsupervised feature selection approach called Unsupervised Laplacian Score (LS) [11] to select the most relevant features.

To analyse these feature sets, we designed several experiments over a very large dataset collected from Tagged.Com. Finally, we compared the results of our proposed set of features with those introduced in [2] and evaluated the two approaches in a number of ways. Consequently, this study is among the first to analyse spammers' behaviour in a multiple network setup through a different combination of features that is determined from an unsupervised feature selection process. Practically, we show that this approach is a lot more productive from the reduction of time complexity while achieving a good set of features that result in good spammer detection compared to the other feature sets from other comparable studies.

2 Related Works

A recent study by [4] used both behavioural and linguistic features for users and reviews to identify spammers in heterogeneous information networks. [5–7] used a combination of user behaviour and content base features to detect spammers in different social networks namely Facebook, Twitter, Myspace, and SinaWeibo.

Table 1. Summary of related works

Reference	*Type of features*					*Type of Network*			Adaption of feature selection
	Content	Structure		Temporal		Heterogeneous	Multiplex	Simplex	
		Graph	Behaviour	Bursty	Sequence				
[4]	✓		✓			✓		✓	
[5]	✓		✓					✓	
[6]	✓		✓	✓				✓	
[7]	✓		✓					✓	
[8]	✓		✓	✓				✓	
[9]		✓	✓	✓	✓			✓	
[2]		✓				✓	✓		
This study			✓	✓	✓	✓	✓		✓

Rather than detecting individual spammers, [8] explored spam campaigns in Facebook's wall posts through similarity graphs. They discussed the temporal behaviour of malicious activities using two key features, the distributed coverage and bursty nature [12]. In [9] four different sets of features, behavioural, graph, automation, and timing were used whereas in [2] three different feature sets were used to analyse spammers in a multiplex network: graph-based, user demographic, and sequence-based features.

In this study, we analysed spammers' behaviour in a multiplex social network regardless of their message contents. Instead, we focused on their interaction within the network, which is harder to mask as reported in [2, 10]. This is a more robust approach than content-based approaches on a simplex network. Despite [2, 9] that used graph theory concepts to extract structural features, we used behavioural features. Then to capture the temporal behaviour of users, we used a set of features to estimate the bursty nature of spammers' activities that was not considered in [2]. We further drew inspiration from gene sequence analysis to capture a set of sequence-based features. The dimensionality of these features were then reduced through feature selection using an un-supervised Laplacian scoring. Table 1 summarizes the characteristics of networks and features used in this study in relations to current works in the area.

3 Multiplex Network Modelling and Problem Definition

Multiplex networks are usually defined in different distinct layers such that each layer shows a different kind of relationship between a common set of network nodes [13]. In each layer, underlying nodes share the same type of relationship [13]. Therefore, in order to distinguish different types of edges between each pair of nodes and simulate different layers in a network, a "relation type" label is assigned to each edge. This label denotes the type of relationship between each pair of nodes [13]. Figure 1 shows a sketch of a multiplex network with two different type of relations shown by two layers a and b, while the two layers share the same sets of nodes.

We denote any directed time-stamped multiplex social network as a graph G where $G = \mathrm{U}_{i=1}^{R} G_i$, where $i \in R$ is a relation type and $G_i = (V_i, E_i, R_i, T_i)$ is a sub-graph of G. For each sub-graph G_i, E_i denotes a directed relation of type R_i which is created between each pair of vertexes from vertex set V_i at time stamp T_i [2, 14].

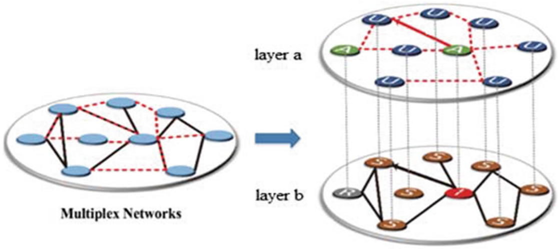


Fig. 1. Sketch of a Multiplex Network with two different types of links [13].

The problem is to predict if any user in multiplex social network denoted by vertexes in graph *G*, is a normal user or a spammer considering his interactions in multiple layers of the network. For this purpose, we investigated each users' behaviour in his neighbourhood graph by extracting some features and then mapped all users to a class from [Spammer, Normal] set.

4 Proposed Approach

Our approach to detect spammers in a large time-evolving multiplex social network relied on the interactions of users in the network. We defined four sets of easy to calculate features: profile-based features, behaviour-based features, bursty features, and sequence-based features. Profile-based features are user demographic information usually obtainable from their registration with a social network. The other three feature sets are described below.

4.1 Behaviour Based Features

To extract the behavioural characteristics of users in their interactions network, we relied on premise that spammers can usually control their own activities to easily evade detection systems [15]. But it would be much more difficult to control their neighbours' activities and therefore, the quality of the neighbourhood is considered as a key factor for identifying spammers and non-spammers [15].

For this purpose, we relied on the analysis of spammers behaviour in the popular social media mainly Twitter and extracted a set of behaviour-based features to capture the community structure (collusive behaviour) of spammers. These features were inspired by follower-followee and friendship relationship in Twitter. Mapping the follower-followee context into our problem, we extracted a set of easy to calculate behaviour-based feature as follows:

- Bidirectional link ratio [9, 16]: Total number of users who are in a bidirectional relation with a user, to his total friends: $\frac{N_{bilink}}{N_{friends}}$
- Average Neighbours' Followers [9]: The total number of friends followers to the total number of friends: $\frac{N_{friends\ followers}}{N_{friends}}$
- Followings to Median Neighbours' Followers [9, 17]: The number of total friends to the median of friends followers:$\frac{N_{friends}}{Median\left(N_{friends\ followers}\right)}$
- Average tweets of friends [18]: The total number of friends to the total number of friends relations:$\frac{N_{friends\ relations}}{N_{friends}}$
- Follower to Following Ratio (fof) [18]: The total number of a user's followers to the total number of his friends:$\frac{N_{followers}}{N_{friends}}$
- Total follow in/out ratio [17]: The number of total relations made with a user to the total relations a user makes:$\frac{N_{in-relations}}{N_{out-relations}}$

- Spamicity [15, 19]:$\frac{1+N_{out-relations}}{1+(N_{in-relations}+N_{out-relations})}$
- Reputation [7]: $\frac{1+N_{followers}}{1+\left(N_{followers}+N_{friends}\right)}$
- Average Neighbour Reputation [20]: $Avg\left(Reputation_{friends}\right)$

4.2 Bursty Features

Bursty behaviour or burstiness is defined as intermittent increase or decrease in the frequency of events [21]. Unlike human perception which tends to accept regularity in the happening of various events, the pieces of evidence have shown that events happen in a burst in the form of a large number of very rapidly occurring events in a short period of time [21]. To capture the burst of users' activities we used three different measures:

- B-measure [22]: $B=\frac{\delta-\mu}{\delta+\mu}$, where μ is the average number of user activities per day and δ is the standard deviation of them. $B\in[-1,1]$ and correlates the burstiness as B = 1 is the most burst signal, B = 0 is neutral, and B = –1 shows a completely regular signal.
- Median of activities (median activity rate) [9, 16, 23]: Median number of users activities per day.
- Median of inter-time activity (median activity time) [9, 16, 23]: Median of each user's inter-time activities

4.3 Sequence-Based Features

In multiplex social networks, spammers are frequently switching between different relation types and sending spamming messages to different users [24]. To capture these temporal behaviour, we defined two different sets of sequence-based features.

- Relative Abundance: Inspired from gene sequence analysis [25], for each user μ_i who is creating a discrete sequence of n relations in time stamps $T_i(i=1,2,\ldots n)$, we define Relative Abundance as $RA_{ij}\frac{P_{ij}}{P_iP_j}$, $(i,j)\in n$, where P_i or P_j is the frequency of the occurrence of activity i or j, and P_{ij} is the joint probabilities of activity i and j. If one sequence is completely stochastic and the activities are mutually independent, then theoretically $P_{ij}=P_iP_j$ and the value of RA_{ij} is 1. Therefore, the deviation of RA_{ij} to 0 could evaluate the bias from the normal behaviour. Therefore, for spammers RA_{ij} tends to 1.
- Distinct Neighbours Ratio: For each user u_i who is sending messages to a discrete sequence of n users in time stamps $T_i(i=1,2,\ldots,n)$, we define Distinct Neighbours Ration as $DN_{u_i}=\frac{N_{Ds}}{S}$, where N_{Ds} is the number of distinct users in user u_i neighboring sequence and S is the total number of users in this sequence. If the neighbouring sequence of a user is completely stochastic and the users who a user is related with are mutually independent, then theoretically $N_{Ds}=S$ and the value of DN_{u_i} is 1. Therefore, the deviation of DN_{u_i} to 0 could evaluate the bias from the normal behaviour. Therefore, for spammers DN_{u_i} tends to 1.

5 Unsupervised Feature Selection

In many learning domains, a human defines sets of potential features. However, not all of these features may be relevant [26]. Unnecessary features increase the size of the search space, making generalization difficult. This large dimensionality is a major problem in machine learning and data mining [27]. In such cases, choosing a subset of the original features will often lead to better performance in convergence time and accuracy. In scenarios that data labels are given, we choose the features relevant to the class labels. But in most real-world scenarios we are not given the data labels.

Therefore, using unsupervised feature selection approach can be very helpful for selecting the most relevant features and reducing the dimensionality, removing irrelevant and redundant features [28]. However, feature selection approaches do not always improve classification rate but they would be very helpful to highlight the importance of different features in feature set [29].

To identify key features and reduce dimensionality, we used an unsupervised Laplacian Score (LS) based approach to select features. The main advantage of LS is that despite many feature selection approaches, it evaluates the features according to their locality preserving power. This means LS considers the local structure of data space rather than the global structure. If two different data points are close two each other, LS considers that they are probably related to the same topic [11].

LS based feature selection makes the nearest neighbour graph over the data samples in order to model local structure of the data [28]. Then, it uses the Graph Laplacian of the K-NN graph for calculating the Laplacian score of each feature. Finally, top k features with the lowest score are removed (Due to space limitation we do not include the LS feature selection algorithm but refer to [11] instead.).

6 Experimental Study

6.1 Dataset Description

The experimental data is described in Table 2. This dataset is a labelled data collected from a social networking website called Tagged.com. It is a dating website for meeting new people and provides its users with various methods to make new connections with other users. The original dataset was first used by Fakhraei et al. [2], who published their data for research studies. However, due to security concerns, parts of relations from the original data has been removed. Besides, the spammer labels have been updated with the release of the published data. The dataset includes over 5.6 million of users of which 336,953 users are labelled as spammers. This imbalance distribution of fraudulent users compared to legitimate ones makes the process of classification quite challenging as common classification algorithms tend to produce more errors when the class distribution is imbalanced [30].

Table 2. Dataset Description

Data Features	Description
Users	Over 5.6 million labelled users 5,270,494 normal users and 336,953 spammers
Relations	858,247,099 relations from 7 different type of relations anonymized to $r_i, i \in [1, 7]$

6.2 Compared Baseline

We considered the proposed approach by [2] as a baseline. [2] used a set of graph-based features, profile-based features, and sequence-based features to detect spammers in a multiplex social network. Their graph-based features included PageRank [31], Graph Coloring [32], the number of Connected Components [33], the number of Triangles [34], k-core centrality measure [35], in-degrees, and out-degrees.

In their sequence-based feature set, [2] used two different types of features. The first sequence-based feature was called sequential k-gram. To calculate this feature, they considered $k = 2$ and calculated the occurrence of any two different relation types in the sequence of users' activities. The second sequence-based feature was defined as a probability of spamicity for each users' activity sequence. This spamicity was extracted using Tree-Augmented Naive Bayes [36] relying on the label of the existing data. However, [2] found that this spamicity did not improve the accuracy of classification. Moreover, this feature relied on the availability of data labels. In our analysis, we aimed to introduce features which do not rely on predefined data labels.

Therefore, we omitted this spamicity feature from sequence-based feature set and extracted the rest of features, namely graph-based features, profile-based features and sequence k-gram features as baseline features.

6.3 Proposed Process Framework

To speed up the process of feature extraction, data were collected in a cumulative manner (e.g. in a daily manner and during night runs) to form different layers of the multiplex network. We simulated the process of behaviour and bursty feature sets extraction in Microsoft SQLServer 2016. Since most of the massive real-world network data are stored in relational data management systems databases [37, 38] (e.g. SQL, Oracle, Vertica) for easy updating and accessing, implementing these features in SQLServer will be very important in productivity and performance of the proposed approach [38].

Then behaviour-based features (f_{BH}) and bursty features (f_B) were extracted for each layer $l \in L$ separately. The sequence-based features (f_S) were also extracted in parallel with bursty and behaviour-based features using designed Python engine. User demographic profile features (f_P) were extracted from user profile information. Finally, all features were combined to form a feature vector F:

$$F = [f_{B1}, f_{B2}, \ldots, f_{Bl}, f_{BH1}, f_{BH2}, \ldots, f_{BHl}, f_S, f_P]$$

Figure 2 shows our proposed framework for data storage, feature extraction, and classification.

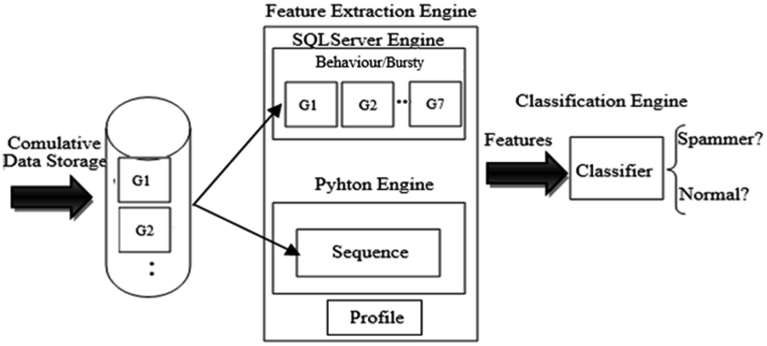


Fig. 2. The Process Framework for Proposed Approach

6.4 Experimental Setup

We performed various sets of experiments on an Azure server with 112 GB RAM and 2.40 GHz CPU (16 processor). In order to compare the proposed approach with the baseline [2], we extracted the baseline features in Python using GraphLabCreate^{TM} which is a distributed graph database.

We performed our experiments using three different classifiers: Support Vector Machine (SVM), Random Forest Regression and Gradient-Boosted Decision Trees with default parameter values in GraphLabCreate^{TM} package. First each introduced feature set was separately evaluated and then all features were modeled into one framework to be evaluated. In next experiment we compared the result of this study with baseline [2].

The next step in this experiment was the selection of the best performance metric and the preventive mechanisms that should be taken on identified spammer accounts. The most appropriate metric for measuring the performance of proposed features on this highly imbalanced dataset is the Precision-Recall curve [39]. Spammer accounts with high precision are automatically blocked in order to prevent any harmful effect on the legitimate users. In contrast, users with high recall, are given the chance to prove their legitimacy by deploying security actions such as Completely Automated Public Turing Tests to Tell Computers and Humans Apart (CAPTCHA). Therefore, we report the accuracy, area under Precision-Recall curve (AUPR) for measuring the performance. Area Under the Receiver Operating Characteristics Curve (AUROC) can be a useful measure, however due to the high class-imbalance, it does not give much insight about the data [39].

6.5 Result and Discussion

Feature Extraction Time: Regardless of the difference between the type of features in this study and baseline, implementing features in relational databases is highly efficient as features in this study took 2.25 (h) to be extracted. Compared to the baseline features which were implemented in a graph database and took 5.27 (h) to be extracted in our experimental setup, the proposed approach in this study has a higher productivity. Sequence-based features in both studies also require one pass through the whole relations.

Accuracy: Fig. 3 (a–d) compares the average result of AUPR, AUROC, and accuracy using 10-fold cross-validation and different feature sets in this study. It shows that Boosted Tree classifier has the best results of AUPR and AUROC over a combination of all features. This classifier is a collection of decision trees combined through a technique called gradient boosting.

Our Analysis also shows that the features in our study outperform the baseline. Since Boosted Tree classifier had the best results in this data, we used this classifier to compare our study with the baseline (Fig. 3 (c)).

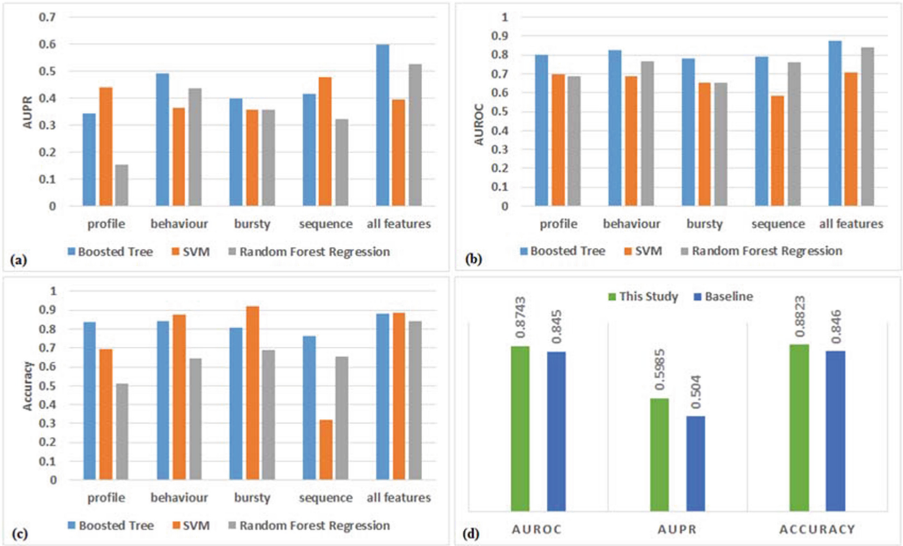


Fig. 3. (a–c) AUPR, AUC and Accuracy of Different Features in Our Study in Three Classifiers. (d) Average AUPR, AUC and Accuracy of All Features in Our Study Compared with Baseline Features using Boosted Tree Classifier.

Feature Importance: Finally, we applied feature selection to the full framework of features in this study. This experiment did not have much improvement in the AUPR and AUROC. But it highlighted the importance of different sets of features in our study (Fig. 4 (a)) and helped to select the best features to reduce the dimensionality of feature space and training time of the classifier. As it is shown, the proposed sequence-based features have the highest share of laplacian score and bursty features are the second important sets of features while behaviour and profile features have the least importance in this data. Figure 4 (b) compares the training time and AUPR for the selected features with different laplacian score. Features with laplacian score larger than 0.005 have the best training time while keeping the near maximum AUPR.

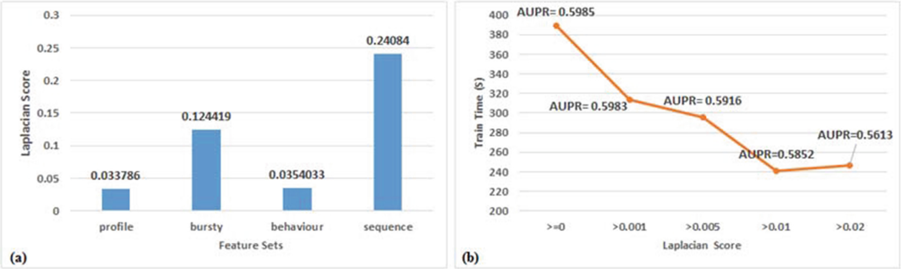


Fig. 4. (a) Overall Average Laplacian Score of Different Feature Sets in This Study. (b) Overall Average AUPR and Training Time Over Selected Features With Various Laplacian Score Threshold.

7 Conclusion and Future Work

We analysed users' behaviour in a time-evolving multiplex social network with four different sets of easy to calculate features as a way to detect spammers. These features included a set of profile-based features, a set of light-weight behaviour-based features, a set of bursty features, and a set of sequence-based features inspired by gene sequence analysis. We also introduced the use of Laplacian feature selection and as shown, our set of sequence-based features achieved the highest score compared to existing combination of features proposed. Our experimental results further showed that our proposed feature sets had empirically less time-complexity while achieving higher AUPR and AUROC and Accuracy compared to the baseline approach. The behaviour-based and bursty features can be easily captured in relational databases, which is their raw form used by many real-world systems. Lastly, our experimental results showed an accuracy of over 88% in spammer detection even on an incomplete dataset, where part of data was obfuscated due to privacy concerns. Nevertheless, the results paved the way for further work as we seek to employ relational learning to analyse the influence of predefined labelled users on the users with unknown label.

References

1. Stringhini, G., Kruegel, C., Vigna, G.: Detecting spammers on social networks. In: Proceedings of ACSAC10, USA (2010)
2. Fakhraei, S., Foulds, J., Shashanka, M., Getoor, L.: Collective spammer detection in evolving multi-relational social networks. In: Proceedings of KDD15, Australia, pp 1769–1778. ACM (2015)
3. Agrawal, D., Budak, C., El Abbadi, A., Georgiou, T., Yan, X.: Big data in online social networks: user interaction analysis to model user behavior in social networks. In: Madaan, A., Kikuchi, S., Bhalla, S. (eds.) DNIS 2014. LNCS, vol. 8381, pp. 1–16. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-05693-7_1
4. Shehnepoor, S., Salehi, M., Farahbakhsh, R., Crespi, N.: NetSpam: a network-based spam detection framework for reviews in online social media. IEEE Trans. Inf. Forensics Secur. **12**, 1585–1595 (2017)
5. Zheng, X., Zeng, Z., Chen, Z., Yu, Y., Rong, C.: Detecting spammers on social networks. Neurocomputing **159**, 27–34 (2015)
6. Benevenuto, F., Magno, G., Rodrigues, T., Almeida, V.: Detecting spammers on Twitter. In: 7th Annual Collaboration, Electronic Messaging, AntiAbuse and Spam, USA (2010)
7. Wang, A.H.: Don't follow me: spam detection in Twitter. In: International Conference on Security and Cryptography, Greece (2010)
8. Gao, H., Hu, J., Wilson, C., Li, Z., Chen, Y., Zhao, B.Y.: Detecting and characterizing social spam campaigns. In: Proceedings of IMC 2010, Australia, pp 35–47. ACM (2010)
9. Yang, C., Harkreader, R., Gu, G.: Empirical evaluation and new design for fighting evolving Twitter spammers. IEEE Trans. Inf. Forensics Secur. **8**, 1280–1293 (2013)
10. Hooi, B., Shin, K., Song, H.A., Beutel, A., Shah, N., Faloutsos, C.: Graph-based fraud detection in the face of camouflage. ACM Trans. Knowl. Discov. Data **11**, 1–26 (2017)
11. He, X., Cai, D., Niyogi, P.: Laplacian score for feature selection. In: Proceedings of NIPS 2005, Canada, pp 507–514. MIT Press (2005)
12. Xie, Y., Yu, F., Achan, K., Panigrahy, R., Hulten, G., Osipkov, I.: Spamming botnets: signatures and characteristics. In: Proceedings of SIGCOMM 2008, USA. vol. 38, pp. 171–182. ACM (2008)
13. Liu, T., Li, P., Chen, Y., Zhang, J.: Community size effects on epidemic spreading in multiplex social networks. PLoS One **11**, e0152021 (2016)
14. Schlichtkrull, M., Kipf, T.N., Bloem, P., Berg, R.v.d., Titov, I., Welling, M.: Modeling Relational Data with Graph Convolutional Networks. arXiv preprint arXiv:170306103 (2017)
15. Karim, M.R., Zilles, S.: Robust features for detecting evasive spammers in Twitter. In: Sokolova, M., van Beek, P. (eds.) AI 2014. LNCS (LNAI), vol. 8436, pp. 295–300. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-06483-3_28
16. Bhat, S.Y., Abulaish, M.: Community-based features for identifying spammers in online social networks. In: IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining Canada (2013)
17. Yang, C., Harkreader, R.C., Gu, G.: Die free or live hard? empirical evaluation and new design for fighting evolving twitter spammers. In: Sommer, R., Balzarotti, D., Maier, G. (eds.) RAID 2011. LNCS, vol. 6961, pp. 318–337. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23644-0_17
18. Chu, Z., Gianvecchio, S., Wang, H., Jajodia, S.: Detecting automation of Twitter accounts: are you a Human, Bot, or Cyborg? IEEE Trans. Dependable Secur. Comput. **9**, 811–824 (2012)

19. Eom, C.S.-H., Lee, W., Lee, J.J.-H.: Spammer detection for real-time big data graphs. In: Proceedings of EDB 2016, Korea, pp 51–60. ACM (2016)
20. Karsai, M., Jo, H.-H., Kaski, K.: Bursty Human Dynamics. SC. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-68540-3
21. García-Pérez, G., Boguñá, M., Serrano, M.Á.: Regulation of burstiness by network-driven activation. Sci. Rep. **5**, 9714 (2015)
22. Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., Tesconi, M.: Fame for sale: Efficient detection of fake Twitter followers. Decis. Support Syst. **80**, 56–71 (2015)
23. Bindu, P.V., Mishra, R., Thilagam, P.S.: Discovering spammer communities in Twitter. J. Intell. Inf. Syst. 1–25 (2018)
24. Jiang, M., Cui, P., Beutel, A., Faloutsos, C., Yang, S.: Catching synchronized behaviors in large networks: a graph mining approach. ACM Trans. Knowl. Discov. Data **10**, 1–27 (2016)
25. Kariin, S., Burge, C.: Dinucleotide relative abundance extremes: a genomic signature. Trends Genet. **11**, 283–290 (1995)
26. Dy, J.G., Brodley, C.E.: Feature selection for unsupervised learning. J. Mach. Learn. Res. **5**, 845–889 (2004)
27. Pourhabibi, T., Imani, M.B., Haratizadeh, S.: Feature selection on Persian fonts: a comparative analysis on GAA, GESA and GA. Procedia Comput. Sci. **3**, 1249–1255 (2011)
28. Zhu, L., Miao, L., Zhang, D.: Iterative laplacian score for feature selection. In: Liu, C.-L., Zhang, C., Wang, L. (eds.) CCPR 2012. CCIS, vol. 321, pp. 80–87. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33506-8_11
29. Enache, A.-C., Sgârciu, V.: An improved bat algorithm driven by support vector machines for intrusion detection. In: Herrero Á., Baruque B., Sedano J., Quintián H., Corchado, E. (eds.) International Joint Conference. CISIS 2015. Advances in Intelligent Systems and Computing. International Joint Conference, vol. 369, pp. 41–51. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19713-5_4
30. Perera, B.K.: A class imbalance learning approach to fraud detection in online advertising. Masdar Institute of Science and Technology (2013)
31. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank Citation Ranking: Bringing Order to the Web. Stanford InfoLab (1999)
32. Jensen, T.R., Toft, B.: Graph coloring problems. Wiley, New York (2011)
33. Pemmaraju, S., Skiena, S.: Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica. Addison-Wesley Longman, Boston (1990)
34. Polak, A.: Counting triangles in large graphs on GPU. In: IEEE International Parallel and Distributed Processing Symposium Workshops (2016)
35. Alvarez-Hamelin, J.I., Dall'Asta, L., Barrat, A., Vespignani, A.: Large scale networks fingerprinting and visualization using the k-core decomposition. In: Proceedings of NIPS 2005 Canada, pp 41–50. MIT Press (2005)
36. Zheng, F., Webb, G.I.: Tree augmented naive bayes. In: Sammut, C., Webb, G.I. (eds.) Encyclopedia of Machine Learning, pp. 990–991. Springer, USA (2010). https://doi.org/10.1007/978-0-387-30164-8
37. Liu, Z., Wang, C., Zou, Q., Wang, H.: Clustering coefficient queries on massive dynamic social networks. In: Chen, L., Tang, C., Yang, J., Gao, Y. (eds.) WAIM 2010. LNCS, vol. 6184, pp. 115–126. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14246-8_14
38. Jindal, A., Madden, S., Castellanos, M., Hsu, M.: Graph analytics using vertica relational database. In: IEEE International Conference on Big Data, pp 1191–1200 (2015)
39. Saito, T., Rehmsmeier, M.: The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLoS One **10**, e0118432 (2015)

Taxonomy-Augmented Features for Document Clustering

Sattar Seifollahi[1,2,3](✉), Massimo Piccardi[1], Ehsan Zare Borzeshi[2], and Bernie Kruger[3]

[1] University of Technology Sydney, Sydney, NSW, Australia
[2] Capital Markets Cooperative Research Centre (CMCRC), Sydney, NSW, Australia
sseif@cmcrc.com
[3] Transport Accident Commission (TAC), Geelong, VIC, Australia

Abstract. In document clustering, individual documents are typically represented by feature vectors based on term-frequency or bag-of-word models. However, such feature vectors intrinsically dismiss the order of the words in the document and suffer from very high dimensionality. For these reasons, in this paper we present novel taxonomy-augmented features that enjoy two promising characteristics: (1) they leverage semantic word embeddings to take the word order into account, and (2) they reduce the feature dimensionality to a very manageable size. Our feature extraction approach consists of three main steps: first, we apply a word embedding technique to represent the words in a word embedding space. Second, we partition the word vocabulary into a hierarchy of clusters by using *k*-means hierarchically. Lastly, the individual documents are projected to the hierarchy and a compact feature vector is extracted. We propose two methods for generating the features: the first uses all the clusters in the hierarchy and results in a feature vector whose dimensionality is equal to the number of the clusters. The second uses a small set of user-defined words and results in an even smaller feature vector whose dimensionality is equal to the size of the set. Numerical experiments on document clustering show that the proposed approach is capable of achieving comparable or even higher accuracy than conventional feature vectors with a much more compact representation.

1 Introduction

With the rapid growth of the world wide web and online services, more and more unstructured textual data - in the form of blog postings, emails, review forums, discussion board messages and speech-to-text transcriptions - are becoming widely available. However, a common complaint is that this increase in data size stresses our ability to understand their contents. Cluster analysis is one of the most frequently used data mining technique to provide structure to large amounts of textual data and organize document collections over meaningful topics.

The most popular clustering algorithms are undoubtedly the *k*-means algorithm and its variants, including hierarchical clustering; for detailed reviews,

R. Islam et al. (Eds.): AusDM 2018, CCIS 996, pp. 241–252, 2019.
https://doi.org/10.1007/978-981-13-6661-1_19

see [1,2,6,18]. These algorithms generally start with creating a vector-space model, known as the bag-of-word (BoW) model. The distinct words in the vocabulary are considered as the basic features of the text, and each document is represented by the vector of the word frequencies in that document. Given that most documents only utilize a small subset of the available words, their feature vectors tend to be very sparse and unnecessarily high-dimensional. Such a high dimensionality can be regarded as an instance of the *curse of dimensionality* [10], making it difficult for clustering algorithms to perform effectively.

Exploiting an ontology is a promising approach to improve the effectiveness of document clustering [7,9,11]. Not only can it help reduce the dimensionality of the feature vector, but also incorporate semantic knowledge in the process. However, how to best exploit ontologies for document clustering is still an open problem [9]. Most of the proposed techniques only employ existing lexical databases such as WordNet to organize the documents' words and find similarities. However, if the ontology does not suit the text corpus, it may lead to poor clusters that fail to reflect its main topics. An example is the case when the text at hand is complex and the existing databases do not cover its vocabulary due to language diversity and/or terminology used in a specific domain. This is the case when the documents are written by many different hands, each with their own writing styles and lexical preferences. The lexical differences and diverse nomenclature used, even among documents with similar topics, can unfavorably affect the effectiveness of clustering algorithms. We tackle one such case in our experiments, where the textual data are phone call transcripts written by the various operators of an accident compensation agency, and many abbreviations and unknown words frequently appear in the text.

Recently, document clustering and classification algorithms based on *word embeddings* [17,19] have been proposed. A word embedding transforms the words from categorical variables taking values in large vocabularies to fixed-dimensional, real-valued vectors. Word embeddings overcome the typical drawbacks of the BoW model, including the dismissal of word ordering and the curse of dimensionality [16]. In the same vein, in this paper we propose an approach based on the combination of two contemporary techniques: (1) the use of taxonomy-augmented features to mollify the curse of dimensionality and embed semantic, and (2) the use of word embeddings to incorporate word ordering and co occurrence. Using an ontology (here, more simply, a taxonomy) can greatly reduce the number of features required by document clustering. At their turn, word embeddings have proven to be effective at capturing semantic regularities, since words with similar meaning and syntactic attributes are projected into the same region in the vector space. Therefore, our goal is to develop a document clustering technique with a strong focus on feature reduction that can improve clustering and identification of topics in a given document collection. To this aim, we use the three-step process described hereafter. First, a word embedding technique is used to convert each distinct word to a vector of $|W|$ dimensions. Second, the word vectors are partitioned into a hierarchy of clusters, simply referred to as "word clusters", via a hierarchical clustering algorithm. Third, the individual documents are projected onto the word clusters to produce their taxonomy-based feature vectors. We use two different algorithms to construct the feature vectors; one is based on projecting the documents directly to the word clusters and results in a feature vector whose size is equal to the number of word clusters. The other is based on a set of user-defined words and results in a feature vector whose size is equal to the size of this set.

Throughout this paper we use the following notations: symbols V, M, N and K are used to denote the number of distinct words (i.e., the vocabulary size), the number of documents, the number of features per document, and the number of clusters, respectively. The dimensionality of the word embedding space is noted as $|W|$. Notations w_ν and y_ν are used for representing the ν-th word and its word embedding, respectively. The word embedding matrix for the entire vocabulary is noted as Y, where $Y = [y_{\nu w}]$; $\nu = 1, \ldots, V$ and $w = 1, \ldots, |W|$. Notation $X = [x_1, \ldots, x_M]$ is used to denote the tf-idf feature vectors of a collection of documents, with $x_m \in \mathbb{R}^V$. Conversely, we use $D = [d_1, \ldots, d_M]$ to note the document vectors in word vector space, i.e. $d_m \in \mathbb{R}^N$ is the m-th document. Notation C_w is used to note the hierarchy of word clusters.

2 Related Works and Motivation

Among different techniques for document representation, techniques using the BoW model are the most widely used [1]. The entire vocabulary of the text corpus is considered as the feature space. In this model, each document is represented by a vector whose elements are the frequency of the distinct words in that document. The document-term matrix of these vectors for the entire collection is very sparse, due to the large number of features. It is also very common to reweigh these vectors to somehow reflect the “discriminative power” of each word. The term frequency-inverse document frequency (tf-idf) approach is the most popular reweighting scheme for the BoW model. The term frequency (tf) is the raw count of the word’s appearances in a document, and the inverse document frequency (idf) increases the importance of words that appear rarely in the collection, assuming that they would be more discriminative in any ensuing clustering and classification tasks [2,8,20].

The BoW model has inherent flaws such as ignoring the word ordering [4] and the curse of dimensionality [10]. Using an ontology, derived from existing databases, is a promising solution to diminish such flaws [7,9,11]. In [11], the authors integrated the WordNet ontology with document clustering. WordNet organizes words into sets of synonyms called “synsets”. Using WordNet, the authors mollified the limitations of the BoW representation by capturing relationships between terms which do not co-occur literally. The work of [21] presents a clustering technique based on an information extraction system, ANNIE, and WordNet that finds the lexical category of each term and uses it to replace the term. The document vectors were reduced to 41 dimensions given that this is the number of lexical categories for nouns and verbs in Wordnet. At its turn, ANNIE is an information extraction system which helps understand whether two words or compound words refer to the same entity [21]. In [9], the authors claim that the drawback of merely augmenting the original words with WordNet synsets is the corresponding increase in the dimensionality of the terms. Therefore, they used the ontology to reduce the number of the features. In addition, they showed that the clustering can be improved by identifying and using "noun features" alone. In turn, the work of [7] leveraged WordNet to decrease the document features to just 26 features representing the WordNet lexical noun categories. They integrated the WordNet ontology with bisecting k-means using the MapReduce parallel programming model.

More recently, various text clustering techniques based on word embeddings [17] have emerged as capable of overcoming the flaws of the BoW model. More precisely, these techniques embed each distinct word in a vector space of dimensionality ($\approx 10^2 - 10^3$) typically much smaller than that of the BoW model ($\approx 10^5 - 10^6$). The word embeddings from [17] have been widely used in the literature for, among other: sentiment analysis [26,29], document distance measurement [13], topic modelling [5,28], and document clustering and classification [12,15,16,27].

Representing entire documents using word embeddings is still a partially unresolved challenge. A simple weighted average of the word embeddings ignores the word ordering, while a parse tree-based combination of the embeddings [24] can only work for whole sentences. Le and Mikolov [14] train word and paragraph vectors to predict context, but sharing the word embeddings across paragraphs. Kim et al. [12] introduce a convolutional sentence kernel based on word embeddings which overcomes the sparsity issue that arises when classifying short documents or in the case of limited training data.

3 Methodology

In this section, we present the proposed approach for extracting taxonomy-based features for document clustering. The approach employs a word embedding method, a hierarchical procedure for word partitioning as well as document vectors based on the word taxonomy.

3.1 Hierarchy of Word Clusters

Cluster analysis based on partitional algorithms and hierarchical algorithms has been extensively studied in the literature (see for example [18] and references therein). The k-means algorithm and its variants are the most broadly used partitional algorithms [1,6,22,25]. Hierarchical algorithms can be divided into two main categories, namely agglomerative and divisive. Agglomerative algorithms are a bottom-up approach, i.e. the clustering produced at each layer of the hierarchy merges similar clusters from the previous layer. Conversely, divisive algorithms sub-divide clusters incrementally, starting from the initial data set. Agglomerative algorithms generally perform better than divisive algorithms, and often "better" than single-layer algorithms such as k-means [18]. However,

[25] found that "bisecting" k-means can produce clusters that are both better than those of standard k-means and as good as (or better than) those produced by agglomerative hierarchical clustering. In plain terms, bisecting k-means joins k-means with divisive hierarchical algorithms and has been successfully applied to document clustering [25].

For these reasons, in this paper we employ a method similar to bisecting k-means with two modifications; (1) we use spherical k-means instead of standard k-means and (2) each cluster is split into two sub-clusters only if the number of its elements exceeds a predefined threshold. Spherical k-means projects data onto the unit sphere and has been shown to be an effective method for document clustering [6]. Building clusters in a hierarchical fashion has the advantages of reducing time complexity, increasing performance as well as implicitly producing a taxonomy of words, where words with similar semantics are expected to appear in the same branch or close branches of the hierarchy. Figure 1 illustrates our hierarchical word clusters. All data (word vectors) are, first, partitioned into two clusters, and from the first level down each cluster is iteratively partitioned into two new clusters until the level limit is reached.

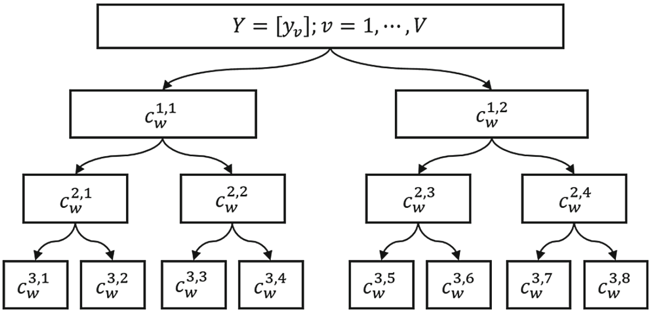


Fig. 1. Three layers of the hierarchy of word clusters.

Let us note the hierarchy of word clusters as C_w. This expands as $C_w = [C_w^1, \ldots, C_w^L]$, where $C_w^l = \{C_w^{l,1}, \ldots, C_w^{l,2^l}\}$ is the set of clusters in the l-th level. The process of building the hierarchy of clusters is illustrated in Algorithm 1.

We have two stopping criteria in Step 4; either when l exceeds an integer value, L (the maximum number of levels in the hierarchy) or when the number of words in a cluster is less than a predefined threshold, δ. Increasing the number of levels may increase the quality of the word clusters, hence enhancing the quality of the ensuing document vectors; however, it also increases the complexity and therefore a heuristic trade-off is needed. Some of the clusters at the lower levels may have no entities (i.e., they are empty) due to a small number of words ($< \delta$) in the corresponding upper layer.

Algorithm 1. Computing the hierarchy of clusters in word space

```
Input: Word vectors $Y = [y_\nu]$; $\nu = 1, \dots, V$; $y_\nu \in \mathbb{R}^{|W|}$
Output: Hierarchy of word clusters $C_w$.

1. Set level number $l = 0$, a threshold $\delta$ as lower bound for splitting a cluster and $C_w = \emptyset$.
2. For $i = 1, \dots, 2^l$ :
   2.1 Partition $C_w^{l,i}$ into two new clusters as $C_w^{l+1,2i-1}$ and $C_w^{l+1,2i}$.
3. Set $C_w = C_w \cup \{C_w^{l+1,1}, \dots, C_w^{l+1,2^l}\}$.
4. If the stopping criteria are satisfied, exit; otherwise, set $l = l + 1$ and repeat from 2.1..
```

3.2 Taxonomy-Augmented Features Given the Hierarchy of Word Clusters

A taxonomy can play a key role in document clustering by reducing the large number of features from thousands to a few tens only. The feature reduction process benefits from the taxonomy's semantic relations between words. Our utilization of the taxonomy differs from previous work, since we construct a hierarchy of word clusters and utilize it directly for document representation. In other terms, the documents are projected to a word cluster space by matching their contents to the hierarchy of word clusters. Let us assume that $D = [d_1, \dots, d_M]$ are M documents in word space. Each document can be further noted as $d_i = [d_i^1, d_i^2, ..., d_i^L]$, where L is the number of levels in the hierarchy of word clusters and $d_i^l = [d_i^{l,1}, d_i^{l,2}, ..., d_i^{l,2^l}]$. For simplicity, we store D in a two-dimensional matrix of size $M \times N$, where N is the overall number of clusters in all levels, i.e. $N = \sum_{l=1}^{L} 2^l$.

Algorithm 2. Taxonomy-augmented features

```
Input: Hierarchy of word clusters $C_w$
Output: Document matrix $D \in \mathbb{R}^{M \times N}$

1. Set $D = [d_1, \dots, d_M] = 0$ where $d_i \in \mathbb{R}^N$, $N = \sum_{l=1}^{L} 2^l$, and $L$ is the number of layers in the hierarchy of word clusters.
2. For each document $d_i, i = 1, \dots, M$:
   2.1. For each word $w$ in document $d_i$:
      2.1.1. For level $l = 1, \dots, L$:
         2.1.1.1. Find cluster index in $C_w^l$, $j$, such that $w \in C_w^{l,j}$.
         2.1.1.2. Set $d_i^{l,j} = d_i^{l,j} + x_{iw}$.
   2.2. For $l = 1 \dots L$:
      2.2.1. Normalize $d_i^l$ to one, i.e. $\|d_i^l\| = 1$.
3. Remove any features from $D$ that have zero value across all documents.
```

If some of the clusters in the hierarchy never appear in any of the documents in the collection, the corresponding features in D will all be zero. Therefore, we remove those features in Step 3 of Algorithm 2. In Step 2, $x_{i,w}$ corresponds to the document-term weight of the i-th document and word w, which is based on tf-idf weighting.

3.3 Taxonomy-Augmented Features Given a Set of Words

In this section, we propose an alternative approach for representing the documents for clustering. In this approach, the documents are projected to a space of predefined words, and the feature dimensionality is the same as the size of this set of words. This approach is particularly suitable for short documents, where the total number of words per document is limited; typically, say, less than 100 words. For such short documents, the BoW features result in an extremely sparse matrix. In order to alleviate this issue, the representation of short documents must be enriched with information from the semantics of the words. The correlation between words and its use for measuring short-text similarities have been studied in the literature, for example, in [23]. Based on [23], if two short segments do not have any common words, but words from the first segment appear frequently with words from the second segment in other documents, this implies that these segments are semantically related, and any measure of their similarity should be high.

Our proposal differs from previous works, including [23], since we build features by combining the documents' content, the hierarchy of word clusters, and a set of predefined words. An intuition of the approach can be provided in these terms: for every word in a given document and for every level in the hierarchy, we find the cluster where the word belongs. We then retrieve all the predefined words that belong to that same cluster, and we increment their counters. The final counters of the predefined words are used as the feature vector to represent the document. In other terms, the predefined words can be seen as the "representative elements" of the clusters they belong to, and their counters are incremented every time a document's word falls in their cluster. To describe the process precisely, let us assume that $D = [d_1, \ldots, d_M]$ is the document matrix, where $d_i \in \mathbb{R}^N$ and N is the number of the predefined words. With these positions, the steps for generating the document features are described by the following Algorithm 3.

The number of the indices in $I(l, j)$, Step 3, depends on how many of the predefined words are in the cluster of word w (from zero to, potentially, N). Rather than simple counters, the document features d_i^j add up the tf-idf weight of word w, $x_{i,w}$.

4 Experiments

4.1 Datasets

To test and compare the proposed models, we have carried out experiments with three, diverse datasets. The first (*PCalls*) consists of phone call transcripts

Algorithm 3. Taxonomy-augmented features given a set of predefined words

```
Input: Hierarchy of word clusters $C_w$, set of predefined words $S = \{w_1, \dots, w_N\}$
Output: Documents $D \in \mathbb{R}^{M \times N}$

1. Set $D = [d_1, \dots, d_M] = 0$, where $d_i \in \mathbb{R}^N$ and $N$ is the size of the set of predefined
   words. Set $W_s = [w_s^1, \dots, w_s^L] = \big[[w_s^{1,1}, w_s^{1,2}], \dots, [w_s^{L,1}, \dots, w_s^{L,2^L}]\big]$, and $w_s^{l,j} = \emptyset$.
2. For each word $w$ in $S$ :
   2.1 For level $l = 1, \dots, L$ :
       2.11 Find index of cluster in $C_w^l$, j, such that $w \in C_w^{l,j}$.
       2.12 Set $w_s^{l,j} = w_s^{l,j} \cup w$.
3. For each document $d_i, i = 1, \dots, M$ :
   3.1 For each word $w$ in document $d_i$ :
       3.11 For level $l = 1, \dots, L$ :
            3.111 Find cluster index in $C_w^l$, i.e. $j$ such that $w \in C_w^{l,j}$.
            3.112 Retrieve all words of $w_s^{l,j}$ with corresponding indices in $S$, $I(l,j)$.
            3.113 Set $d_i^{I(l,j)} = d_i^{I(l,j)} + x_{i,w}$.
```

recorded by the Transport Accident Commission (TAC), the accident compensation agency in Victoria, Australia. It contains a total of 59,048 transcripts from phone calls from 8,000 single clients, transcribed by various operators. The second (*WebKB*) is a classic clustering benchmark consisting of web pages from computer science departments of universities. The last (*Reuters*) is a dataset of news stories published on the Reuters newswire in 1987. The main statistics of these datasets are given in Table 1. For more details, please refer to [2] and references therein.

Table 1. Dataset summary.

Datasets	Dataset name	M	Total number of words after pre-processing
D1	PCalls	8,000	6,684
D2	WebKB	4,199	2,153
D3	Reuters	12.902	1,313

The following preprocessing steps have been applied to each dataset before its use in the experiments: (1) removal of numbers, punctuation, symbols and “stopwords”; (2) replacement of synonyms and misspelled words with the base and actual words (only for the PCalls dataset); (3) removal of sparse terms (keeping 95 % sparsity or less) and infrequently occurring words; (4) removal of uninformative words such as people names and addresses.

4.2 Experimental Set-Up

We have employed two document vector methods as the baselines for comparison; (1) the well-known tf-idf and (2) doc2vec which is based on the average of word embeddings. As word embeddings, we have used GloVe for its reported strong performance in a variety of tasks [19]. To learn the embeddings, we have used the following settings: the dimensionality was set to 200; the context window size was set to 12; and the number of training epochs was set to 1,000. For clustering, we have used two very popular algorithms, k-means++ and PAM (partitioning around medoids). In Algorithm 3, we set $N = 100$.

4.3 Experimental Results on Document Clustering

In this section, we compare the effectiveness of our model based on two complementary measures: connectivity and silhouette. The connectivity captures the "degree of connectedness" of the clusters and it is measured in terms of how many nearest neighbors of any given sample belong to other clusters. The connectivity value ranges between 0 and infinity and should be minimized. The silhouette measures the compactness and separation of the clusters and it ranges between −1 (poorly clustered observations) and 1 (well clustered observations). To compute these measures, we have used clValid, the R package for cluster validity [3].

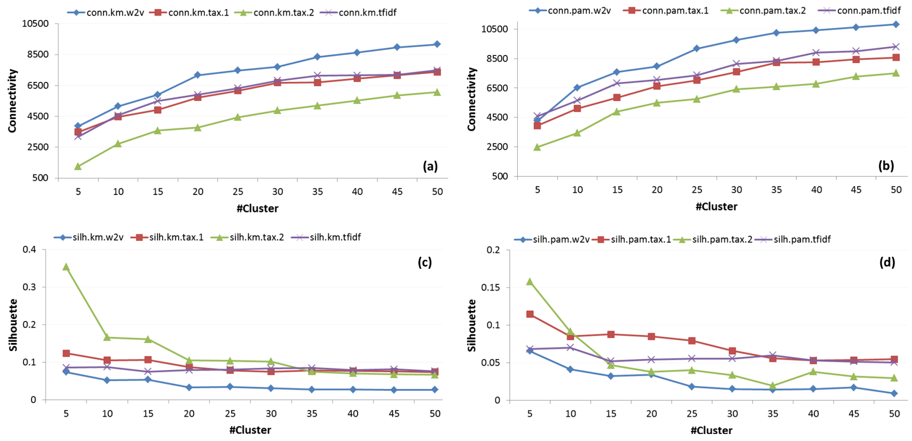


Fig. 2. Connectivity and silhouette measures of all models for the PCalls dataset.

Figures 2, 3 and 4 show the connectivity and silhouette measures for PCalls, WebKB and Reuters as a function of the number of the clusters. In these figures, "tfidf" stands for the BoW model, "w2v" for doc2vec, and "tax.1" and "tax.2" for our proposed models from Algorithms 2 and 3, respectively. In turn, "km" and "pam" stand for k-means++ and PAM, respectively. These results show that

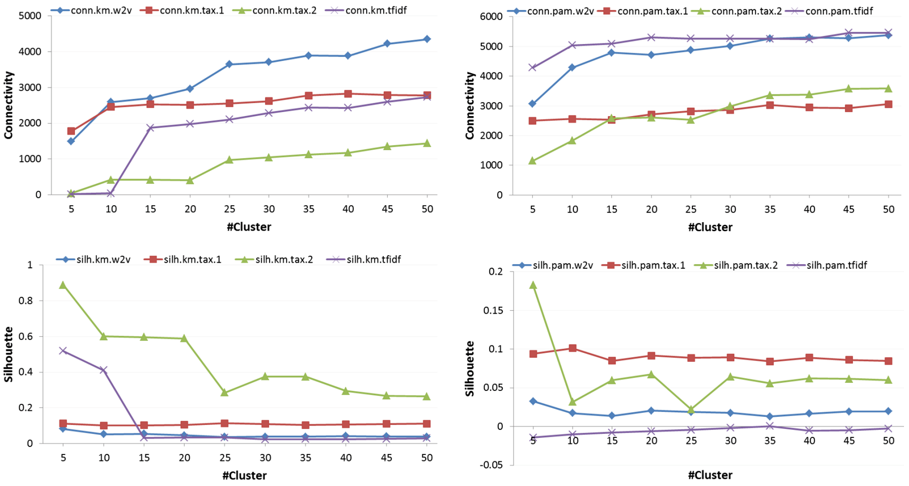


Fig. 3. Connectivity and silhouette measures of all models for the WebKB dataset.

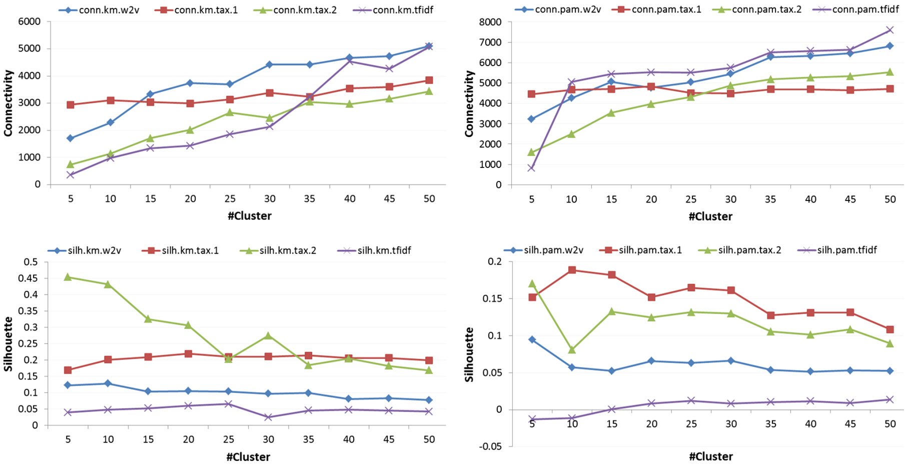


Fig. 4. Connectivity and silhouette measures of all models for the Reuters dataset.

the proposed taxonomy-augmented models, “tax.1” and “tax.2”, have performed better than the other two models over all datasets for a large majority of cluster numbers and for both clustering algorithms. Among the conventional models, the performance of BoW is generally better than that of doc2vec. We do not formally compare the time complexity of the models, but we note that BoW is the most time-consuming due to its large number of features. Based on our experiments, it is approximately 10 times slower than the other models.

5 Conclusion

In this paper, we have presented two original taxonomy-augmented models for document clustering. These models address two urgent challenges of conventional document representations, namely (1) their large number of features, and (2) the dismissal of the word ordering in the formation of the features. By amending these two shortcomings, the proposed models are able to provide a more compact and semantically-meaningful document representation and improve the effectiveness of clustering.

In a set of experiments, we have compared the proposed models with two well-known methods, the BoW/tf-idf model and doc2vec, over three diverse datasets. The results have shown that the proposed models have achieved better connectivity and silhouette measures in the large majority of cases. In the near future, we plan to explore the performance of the proposed models in other important NLP tasks such as document classification and sentiment analysis.

Acknowledgement. This project was funded by the Capital Markets Cooperative Research Centre in combination with the Transport Accident Commission of Victoria. Acknowledgements and thanks to industry partner David Attwood (Lead Operational Management and Data Research). This research has received ethics approval from the University of Technology Sydney (UTS HREC REF NO. ETH16-0968).

References

1. Arthur, D., Vassilvitskii, S.: k-means++: the advantages of careful seeding. In: Gabow, H. (Ed.) Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms [SODA07], pp. 1027–1035. Society for Industrial and Applied Mathematics (2007)
2. Bagirov, A., Seifollahi, S., Piccardi, M., Zare, E., Kruger, B.: SMGKM: an efficient incremental algorithm for clustering document collections. CICLing 2018 (2018)
3. Brock, G., Pihur, V., Datta, S., Datta, S.: clValid: an R package for cluster validation. J. Stat. Softw. **25**, 1–22 (2008)
4. Y. Cheng. Ontology-based fuzzy semantic clustering. In Proceedings – 3rd International Conference on Convergence and Hybrid Information Technology, ICCIT 2008, vol. 2, pp. 128–133 (2008)
5. Das, R., Zaheer, M., Dyer, C.: Gaussian LDA for topic models with word embeddings. In: Proceedings ACL 2015, pp. 795–804 (2015)
6. Dhillon, S., Fan, J., Guan, Y.: Efficient clustering of very large document collections. In: Kamath, C., Kumar, V., Grossman, R., Namburu, R. (eds.) Data Mining for Scientific and Engineering Applications. Kluwer Academic Publishers, Oxford (2001)
7. Elsayed, A., Mokhtar, H.M.O., Ismail, O.: Ontology based document clustering using Mapreduce. Int. J. Database Manag. Syst. **7**(2), 1–12 (2015)
8. Erra, U., Senatore, S., Minnella, F., Caggianese, G.: Approximate tf-idf based on topic extraction from massive message stream using the gpu. Inf. Sci. **292**, 143–161 (2015)
9. Fodeh, S., Punch, B., Tan, P.-N.: On ontology-driven document clustering using core semantic features. Knowl. Inf. Syst. **28**(2), 395–421 (2011)

10. Friedman, J.H.: On bias, variance, 0/1-loss, and the curse-of-dimensionality. Data Min. Knowl. Discov. **1**(1), 55–77 (1997)
11. Hotho, A., Staab, S., Stumme, G.: Ontologies improve text document clustering. In: Third IEEE International Conference on Data Mining, pp. 541–544 (2003)
12. Kim, J., Rousseau, F., Vazirgiannis, M.: Convolutional sentence kernel from word embeddings for short text categorization. In: Proceedings EMNLP 2015, pp. 775–780, September 2015
13. Kusner, M.J., Sun, Y., Kolkin, N.I., Weinberger, K.Q.: From word embeddings to document distances. In: Proceedings - ICML, vol. 37, pp. 957–966 (2015)
14. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: Proceedings of the 31st International Conference on Machine Learning, vol. 32, pp. 1188–1196 (2014)
15. Lenc, L., Král, P.: Word embeddings for multi-label document classification. In: Proceedings of Recent Advances in Natural Language Processing, pp. 431–437 (2017)
16. Lilleberg, J., Zhu, Y., Zhang, Y.: Support vector machines and word2vec for text classification with semantic features. In: Proceedings of IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC), pp. 136–140 (2015)
17. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. Arxiv, pp. 1–12 (2013)
18. Moseley, B., Wang, J.R.: Approximation bounds for hierarchical clustering: average linkage, bisecting K-means, and local search. Number Nips, pp. 3097–3106 (2017)
19. Pennington, J., Socher, R., Manning, C.: Glove: global vectors for word representation. In: Proceedings EMNLP 2014, pp. 1532–1543 (2014)
20. Qimin, C., Qiao, G., Yongliang, W., Xianghua, W.: Text clustering using vsm with feature clusters. Neural Comput. Appl. **26**(4), 995–1003 (2015)
21. Recupero, D.R.: A new unsupervised method for document clustering by using WordNet lexical and conceptual relations. Inf. Retr. **10**(6), 563–579 (2007)
22. Seifollahi, S., Bagirov, A., Layton, R., Gondal, I.: Optimization based clustering algorithms for authorship analysis of phishing emails. Neural Process. Lett. **46**(2), 411–425 (2017)
23. Seifzadeh, S., Farahat, A.K., Kamel, M.S., Karray, F.: Short-text clustering using statistical semantics. In: Proceedings WWW 2015, pp. 805–810 (2015)
24. Socher, R., Bauer, J., Manning, C.D., Ng, A.Y.: Parsing with compositional vector grammars. In: Proceedings ACL 2013, vol. 1, pp. 455–465 (2013)
25. Steinbach, M., Karypis, G., Kumar, V.: A comparison of document clustering techniques. In: KDD workshop on text mining, vol. 400, pp. 1–2 (2000)
26. Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., Qin, B.: Learning sentiment-specific word embedding for twitter sentiment classification. In: Proceedings ACL, pp. 1555–1565 (2014)
27. Wang, P., Xu, B., Xu, J., Tian, G., Liu, C.L., Hao, H.: Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification. Neurocomputing **174**, 806–814 (2016)
28. Xun, G., Gopalakrishnan, V., Li, F.M.Y., Gao, J., Zhang, A.: Topic discovery for short texts using word embeddings. In: Proceedings - IEEE International Conference on Data Mining, ICDM, pp. 1299–1304 (2017)
29. Zhang, D., Xu, H., Su, Z., Xu, Y.: Chinese comments sentiment classification based on word2vec and SVMperf. Expert Syst. Appl. **42**(4), 1857–1863 (2015)

A Modularity-Based Measure for Cluster Selection from Clustering Hierarchies

Francisco de Assis Rodrigues dos Anjos[1], Jadson Castro Gertrudes[1], Jörg Sander[2], and Ricardo J. G. B. Campello[3](✉)

[1] University of São Paulo, São Carlos, SP, Brazil
{anjosfar,jadsoncastro}@usp.br
[2] University of Alberta, Edmonton, AB, Canada
jsander@ualberta.ca
[3] University of Newcastle, Callaghan, NSW, Australia
ricardo.campello@newcastle.edu.au

Abstract. Extracting a flat solution from a clustering hierarchy, as opposed to deriving it directly from data using a partitional clustering algorithm, is advantageous as it allows the hierarchical relationships between clusters and sub-clusters as well their stability across different hierarchical levels to be revealed before any decision on what clusters are more relevant is made. Traditionally, flat solutions are obtained by performing a global, horizontal cut through a clustering hierarchy (e.g. a dendrogram). This problem has gained special importance in the context of density-based hierarchical algorithms, because only sophisticated cutting strategies, in particular non-horizontal local cuts, are able to select clusters at different density levels. In this paper, we propose an adaptation of a variant of the *Modularity Q* measure, widely used in the realm of community detection in complex networks, so that it can be applied as an optimization criterion to the problem of optimal local cuts through clustering hierarchies. Our results suggest that the proposed measure is a competitive alternative, especially for high-dimensional data.

Keywords: Hierarchical clustering · Cluster evaluation and selection

1 Introduction

Clustering [1,2] is a fundamental unsupervised learning technique that has become increasingly important for exploratory data analysis, partially because, as opposed to labeled data that can be difficult and expensive to obtain, cheap unlabeled data have been automatically acquired at an unprecedented scale in various fields. However, the lack of labels makes clustering particularly challenging. In particular, model selection is far from trivial because the evaluation of different clustering candidates must be performed without a ground truth.

In this paper we focus on a particular yet very important model selection problem in clustering, which is the problem of assessing and selecting the most

R. Islam et al. (Eds.): AusDM 2018, CCIS 996, pp. 253–265, 2019.
https://doi.org/10.1007/978-981-13-6661-1_20

prominent clusters from a clustering hierarchy. This selection corresponds to extracting a flat clustering solution that is globally optimal according to a certain optimization criterion, subject to non-overlapping clustering constraints.

One of the advantages of hierarchical clustering algorithms is to be able to represent clusters at different levels of granularity or density, from which a flat clustering solution, a so-called *data partition*, can be extracted, if desired. Extracting a flat solution from a clustering hierarchy, as opposed to deriving it directly from data using a partitioning clustering algorithm, such as k-means or EM [3], is advantageous as it allows the hierarchical relationships between clusters and sub-clusters as well their stability across different hierarchical levels to be revealed before any decision is made on what clusters are more relevant. This way, several candidate clusters and clustering solutions can be explored to guide decisions such as how many clusters to select, which otherwise would be required, implicitly or explicitly, as some sort of critical user-defined parameter.

The usual approach to extract a flat solution from a clustering hierarchy is the well-known "horizontal cut", which essentially corresponds to choosing one of the levels of the hierarchy and, as a byproduct, the number of clusters [4]. Choosing the best cut level is a classic problem in cluster analysis, which is traditionally addressed by applying some statistical test or index to quantify the suitability of each candidate cut level [5]. The horizontal cut approach has prevailed for decades as the single standard approach for this problem, even though it has a major limitation, historically overlooked until very recently: it cannot extract clusters from different levels along different branches of the hierarchy. In practice, this means that clusters in different regions of the data cannot be looked into at different levels of specificity [6]. From a density estimation perspective, this means that clusters cannot be extracted using different density thresholds, and hence clustering solutions containing clusters with varied densities cannot be obtained in general, as they will hardly correspond to a single level of a hierarchy [7]. This is also a well-known limitation of partitioning-like density-based clustering algorithms, such as DBSCAN, DENCLU, and others [8].

As opposed to partitioning-like algorithms, HDBSCAN* [7,9] is a state-of-the-art hierarchical clustering technique that has rapidly become a popular choice among density-based methods, with very efficient third-party implementations already available in major open-source software distributions such as R/CRAN [10] and Python/SciKit-learn [11]. The algorithm does not require any critical parameter and builds a complete hierarchy of density-based clusters from which a flat solution, possibly consisting of clusters with varied densities, can optionally be extracted. For cluster extraction, HDBSCAN* uses a generic *Framework for Optimal Selection of Clusters* (FOSC), originally introduced in [6], which performs local cuts through clustering hierarchies and is guaranteed to maximize a given, suitable measure of clustering quality.

In this paper, we propose a new measure of cluster quality and show that it satisfies the properties required by FOSC. We show that the proposed measure, which is based on a variant of the *Modularity Q* measure widely used in the realm of community detection in complex networks [12], is a competitive alternative to the original measure of cluster *Stability* used by FOSC in HDBSCAN*.

2 Related Work

A few methods in the literature can explicitly or implicitly perform local cuts through a clustering hierarchy. These methods typically exhibit one or more major limitations. For example, the visualization-based interactive tools to select clusters from hierarchies proposed in [13] require human intervention to make decisions. The greedy search methods proposed in [14,15] are fully automatic, but they are merely heuristic and provide no optimality guarantees. This is also the problem with methods that are integrated as part of the clustering algorithm itself, such as [16,17]; in these cases, the obtained solution can be shown to be equivalent to performing local cuts through a given type of hierarchy, but in an implicitly and heuristic way that critically depends on a user-defined threshold and is limited to a particular type of clustering algorithm. The method in [18] is also limited to a particular type of clustering algorithm (OPTICS) and, besides, the only ad-hoc approach to perform local cuts through the corresponding cluster tree is to arbitrarily take all the leaf clusters and discard the others.

In [6], FOSC was introduced as the first general method in the literature able to perform local cuts through clustering hierarchies, such that: (a) it can be applied to any clustering hierarchy, provided as input to the framework; (b) it is guaranteed to find (in linear time with a bottom-up pass through the hierarchy) a globally optimal solution that maximizes a user-defined measure of clustering quality, provided that this measure satisfies the properties of *additivity* and *locality*. To date, the original measure proposed in [6], called *Stability*, as well as its variant for density-based hierarchies that is used by the state-of-the-art algorithm HDBSCAN* [7,9], still represent the standard choice in the literature.

The development of supplementary quality measures that satisfy the properties required by FOSC is particularly relevant because, in the context of cluster analysis, due to the intrinsic subjectivity of the notion of cluster itself, assessing cluster quality is particularly challenging. Indeed, it is often argued that the analyst should not rely on a single measure to perform this task [19,20].

3 Background

3.1 HDBSCAN*

HDBSCAN* [7,9] is a hierarchical density-based clustering algorithm that represents an improvement over its predecessors, DBSCAN and OPTICS, mainly in that it produces an explicit hierarchy of nested density-based clustering structures with respect to a single parameter, m_{pts}, which is a smoothing factor of a non-parametric density estimate. A specific level at some height ε in this hierarchy represents a density threshold that determines whether data objects are part of density-based clusters or noise: dense objects (those above the threshold) are part of some cluster at the corresponding level, while those below the threshold are deemed noise. Dense objects, so-called *core points* w.r.t. ε and m_{pts}, are those that have at least m_{pts} many points in their ε-neighborhood, i.e., within a ball of radius ε centered at the point; they are called a *noise point* otherwise. Two core points are directly *ε-reachable* if they are within each other's ε-neighborhood, and they are *density-connected* if they are directly or transitively ε-reachable. A *cluster* is a non-empty maximal subset of density connected points.

The above definitions of cluster and noise are the same as those used by DBSCAN [21], except for the so-called *border points*. Unlike DBSCAN, however, HDBSCAN* does not need ε as a parameter. Instead, it efficiently produces a complete hierarchy for all possible values of $\varepsilon \in [0, \infty)$. Each hierarchical level corresponds to a DBSCAN* solution (DBSCAN* stands for DBSCAN without the border points, which can be added in a post-processing step if desired) associated with the range of ε values that produce that unique solution.

To determine the *nested* structure of density-based clusters in a dataset $\mathbf{X}$, w.r.t. m_{pts}, HDBSCAN* works implicitly, i.e., conceptually only, with a complete weighted graph, called *Mutual Reachability Graph*, $\mathbb{G}_{m_{pts}}$, whose vertices represent the data objects in $\mathbf{X}$, and the edge weight between two objects $\mathbf{x}_1$ and $\mathbf{x}_2$ (called *mutual reachability distance*) is the smallest value of ε such that $\mathbf{x}_1$ and $\mathbf{x}_2$ are directly ε-reachable w.r.t. m_{pts}. For a specific density level associated with ε, removing all edges from $\mathbb{G}_{m_{pts}}$ with weights greater than ε reveals the maximal connected components of that level as clusters. The density-based clustering hierarchy can thus be easily extracted by removing edges in decreasing order from an augmented Minimum Spanning Tree of $\mathbb{G}_{m_{pts}}$, MST_r [7].

As an illustrative example, consider the toy dataset in Fig. 1, with 15 objects. The complete clustering hierarchy produced by HDBSCAN* with $m_{pts} = 3$ is shown in Table 1, where rows correspond to hierarchical levels (density thresholds for varied ε), columns correspond to data objects, and entries contain cluster labels ("0" stands for noise). Notice that, unlike traditional dendrograms, clusters can shrink and yet retain the same label when individual objects are disconnected from them becoming noise, as the density threshold increases for decreasing values of ε, when going top-down the hierarchy. Only when a cluster is divided into two subsets of density connected objects the resulting subsets are deemed new clusters. This way, the complete hierarchy in Table 1 can be represented as a simplified cluster tree where the root ($\mathbf{C}_1$) is the "cluster" containing the whole dataset, which subdivides into two child nodes corresponding to clusters $\mathbf{C}_2$ and $\mathbf{C}_3$, and $\mathbf{C}_2$ further subdivides into two sub-clusters $\mathbf{C}_4$ and $\mathbf{C}_5$. Figure 2 illustrates the notion of a cluster tree with a more sophisticated example, where cluster $\mathbf{C}_3$ is further subdivided into sub-clusters. If these clusters can be properly assessed according to a suitable cluster quality measure, an optimal flat solution in which objects are guaranteed not to belong to more than one cluster can be extracted, as discussed in the next section.

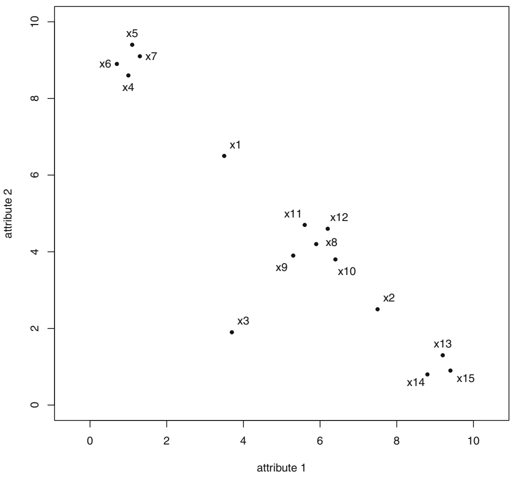


Fig. 1. Illustrative dataset

Table 1. Complete HDBSCAN* hierarchy for the dataset in Fig. 1 ($m_{pts} = 3$).

ε	$\mathbf{x}_3$	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_9$	$\mathbf{x}_{10}$	$\mathbf{x}_{11}$	$\mathbf{x}_{12}$	$\mathbf{x}_8$	$\mathbf{x}_{13}$	$\mathbf{x}_{14}$	$\mathbf{x}_{15}$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_4$	$\mathbf{x}_7$
3.2650	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3.1828	2	2	2	2	2	2	2	2	2	2	2	3	3	3	3
3.1623	0	2	2	2	2	2	2	2	2	2	2	3	3	3	3
2.0809	0	0	2	2	2	2	2	2	2	2	2	3	3	3	3
0.8544	0	0	0	4	4	4	4	4	5	5	5	3	3	3	3
0.8246	0	0	0	0	4	4	4	4	5	5	5	3	3	3	3
0.6403	0	0	0	0	0	4	4	4	5	5	5	3	3	3	3
0.6325	0	0	0	0	0	4	4	4	0	0	5	0	3	3	3
0.6083	0	0	0	0	0	4	4	4	0	0	5	0	0	3	3
0.5831	0	0	0	0	0	0	0	4	0	0	0	0	0	3	3
0.0000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

3.2 FOSC

Let $\{\mathbf{C}_1, ..., \mathbf{C}_k\}$ be the set of all clusters in a hierarchy of $\mathbf{X}$ from which we want to extract a flat solution, $\mathbf{P}$. Assume that the hierarchy is represented by a cluster tree such as the one in Fig. 2, where each node represents a cluster and the root of the tree ($\mathbf{C}_1$) represents the dataset, i.e. $\mathbf{C}_1 = \mathbf{X}$. In addition, assume that there is an objective function $Q_T(\mathbf{P})$ that can quantitatively assess the quality of every valid candidate solution $\mathbf{P}$, and which we want to maximize. For FOSC to work properly, the criterion $Q_T(\mathbf{P})$ must be decomposable according to two properties: (i) **Additivity:** $Q_T(\mathbf{P})$ must be written as the sum of individual components $Q(\mathbf{C}_i)$, each of which is associated with a single cluster $\mathbf{C}_i$ of $\mathbf{P}$;

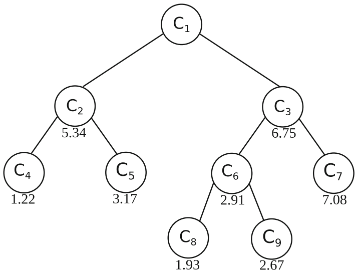


Fig. 2. Example of cluster tree: values below the nodes indicate cluster quality.

(ii) **Locality:** Every component $Q(\mathbf{C}_i)$ must be computable locally to $\mathbf{C}_i$, regardless of what the other clusters that compose the candidate solution $\mathbf{P}$ are.

Due to the property of *locality*, the value $Q(\mathbf{C}_i)$ associated with every cluster in the cluster tree can be computed prior to the decision on what clusters will compose the final solution to be extracted. These are the values illustrated below the nodes in Fig. 2. Due to the property of *additivity*, the objective function can be written as $Q_T(\mathbf{P}) = \sum_{\mathbf{C}_i \in \mathbf{P}} Q(\mathbf{C}_i)$, and the problem we want to solve is to choose a collection of clusters $\mathbf{P}$ such that: (a) $Q_T(\mathbf{P})$ is maximized; and (b) $\mathbf{P}$ is a valid flat solution, i.e., clusters and their sub-clusters are mutually exclusive (no data object belongs to more than one cluster).

FOSC [6] solves this problem using the simple observation that, due to the property of locality, the partial selections made inside any subtree remain optimal in the context of larger trees containing that subtree. This allows the application of a very efficient, globally optimal dynamic programming method that traverses the cluster tree bottom-up starting from the leaves, comparing the quality of parent clusters against the aggregated quality of the respective children, carrying the optimal choices upwards until the root is reached. In Fig. 2, notice that the sum of $Q(\mathbf{C}_8)$ and $Q(\mathbf{C}_9)$ (4.6) is larger than $Q(\mathbf{C}_6)$ (2.91), hence $\mathbf{C}_6$ is discarded while $\mathbf{C}_8$ and $\mathbf{C}_9$ are provisionally selected. Next, the aggregated value of $Q(\mathbf{C}_8)$, $Q(\mathbf{C}_9)$ and $Q(\mathbf{C}_7)$ (11.68) is compared against $Q(\mathbf{C}_3)$ (6.75), which is smaller, hence $\mathbf{C}_3$ is discarded whereas $\mathbf{C}_7$, $\mathbf{C}_8$ and $\mathbf{C}_9$ are retained. Similarly, $Q(\mathbf{C}_2)$ (5.34) is larger than the aggregated value of $Q(\mathbf{C}_4)$ and $Q(\mathbf{C}_5)$ (4.39), hence $\mathbf{C}_2$ is selected whereas $\mathbf{C}_4$ and $\mathbf{C}_5$ are discarded. Since the root has been reached, the final solution is $\mathbf{P} = \{\mathbf{C}_2, \mathbf{C}_7, \mathbf{C}_8, \mathbf{C}_9\}$, with $Q_T(\mathbf{P}) = 17.02$.

While many clustering quality criteria from the literature satisfy the additive property, it is not easy to find criteria that satisfy the locality property required by FOSC. In the original publication [6], the authors proposed a criterion that is based on the classic notion of *cluster lifetime*. The lifetime of a cluster in a clustering dendrogram is basically the length of the dendrogram scale along which the cluster exists [1]. This concept has been adapted in [6] to account for the fact that in certain hierarchies, including density-based hierarchies such as the one in Table 1, not all data objects stay in the cluster during its whole lifetime, because some objects become noise on the way. The measure of *Stability* of a cluster as proposed in [6] is the sum of the lifetimes of every object in that cluster, $Q(\mathbf{C}_i) = \sum_{\mathbf{x}_j \in \mathbf{C}_i} lifetime(\mathbf{x}_j)$. For example, in Table 1, cluster $\mathbf{C}_4$ appears bottom-up at level 0.5831 (formed by object $\mathbf{x}_8$ alone) and disappears when it gets merged with cluster $\mathbf{C}_5$, giving rise to $\mathbf{C}_2$ at level 2.0809. Along this interval, another four objects join this cluster, at levels 0.8544 ($\mathbf{x}_9$), 0.8246 ($\mathbf{x}_{10}$), and 0.6083 ($\mathbf{x}_{11}$ and $\mathbf{x}_{12}$). Hence, its Stability is given by $Q(\mathbf{C}_4) = (2.0809 - 0.5831) + (2.0809 - 0.8544) + (2.0809 - 0.8246) + (2.0809 - 0.6083) * 2 = 6.9258$.

To perform flat cluster extraction, HDBSCAN* uses FOSC with the Stability criterion as described above, except that it replaces ε with $\frac{1}{\varepsilon}$ and flips the scale for the computation of lifetime. This makes Stability more statistically sound in the density-based context as it becomes equivalent to the concept of relative *excess of mass* of a cluster [7]. In this case, the Stability of cluster $\mathbf{C}_4$ in our example would be computed as $Q(\mathbf{C}_4) = (1/0.5831 - 1/2.0809) + (1/0.8544 - 1/2.0809) + (1/0.8246 - 1/2.0809) + (1/0.6083 - 1/2.0809) * 2 = 4.9831$.

3.3 Modularity

Community detection in networks, so-called graph/network clustering or community mining, is essentially the problem of partitioning the vertices of a graph into groups so that the vertices in each group are more interrelated to each other than they are to vertices in other groups [22]. The most widely used quality measure for graph partitioning is the *Modularity Q* proposed in [12]. The intuition behind this measure is that strong communities in a graph tend to have a much larger number of internal connections than the number that would be expected if the connections were randomly distributed according to a given null model of interest. The original Modularity subsumes unweighted graphs. In the literature of community detection, the following weighted version is commonly used [23]:

$$Q_T(\mathbf{P}) = \sum_{i=1}^{k} \left[\frac{IS(\mathbf{C}_i)}{TS} - \left(\frac{DS(\mathbf{C}_i)}{TS} \right)^2 \right] \tag{1}$$

where $\mathbf{P} = \{\mathbf{C}_1, \cdots, \mathbf{C}_k\}$ is a partition of the vertices into k disjoint groups $\mathbf{C}_i$, TS is the total sum of edge weights in the graph, $IS(\mathbf{C}_i)$ is the sum of edge weights within cluster $\mathbf{C}_i$ (i.e., edges for which both end vertices belong to $\mathbf{C}_i$) and $DS(\mathbf{C}_i)$ is the sum of edge weights over the edges that have at least one end vertex in $\mathbf{C}_i$ (i.e., the sum of the *strengths* of the vertices in $\mathbf{C}_i$). In the next section we show how this measure can be adapted for use in FOSC/HDBSCAN*.

4 Modularity-Based Measure for Cluster Extraction

Equation (1) satisfies the properties of *additivity* and *locality* required by FOSC. In fact, notice that it can be written as $Q_T(\mathbf{P}) = \sum_{i=1}^{k} Q(\mathbf{C}_i)$, where $Q(\mathbf{C}_i) = \frac{IS(\mathbf{C}_i)}{TS} - (\frac{DS(\mathbf{C}_i)}{TS})^2$, so it is clearly *additive*. In addition, since TS is a constant, $Q(\mathbf{C}_i)$ depends solely on information available from the vertices in $\mathbf{C}_i$ and their edges, regardless of who the other clusters are, so it is *local*. Therefore, if we represent a dataset $\mathbf{X}$ using a suitable graph, we can use the Weighted Modularity in (1) to extract clusters from a clustering hierarchy of the data using FOSC.

A first, naive idea is to take the complete graph of similarities (G_s) where every pair of vertices is adjacent with edge weight given by the similarity between the corresponding objects in the original feature space (if a distance d, e.g. Euclidean, is used for clustering, it can be converted into similarity as $1-d/d_{max}$, where d_{max} is the largest distance value in the data set). This approach is naive for two different reasons. First, the intuition behind the Modularity is compromised in a complete graph where every pair of vertices is adjacent. Second, the similarity between vertices in G_s does not capture the notion of structural similarity, which is important when assessing community structure [23,24].

To tackle the first problem, we can use a spanning subgraph of the similarity graph, $G_{KNN} \subset G_s$, such that only those edges connecting each vertex to its K nearest neighbors are retained, while edges that do not belong to the set of K largest edges incident to any vertex are discarded (the choice of K will be discussed later). As for the second problem, we adopt a usual approach in the literature of community detection, which is adjusting the edge weights of the resulting graph according to the measure of *structural similarity* [23,24]. Since G_{KNN} is a weighted graph, a weighted version of the measure is used [25]:

$$\sigma(u,v) = \frac{\sum_{x \in \Gamma_u \cap \Gamma_v} w(u,x)w(v,x)}{\sqrt{\sum_{x \in \Gamma_u} w^2(u,x)}\sqrt{\sum_{x \in \Gamma_v} w^2(v,x)}} \tag{2}$$

where Γ_i is the set of vertices adjacent to vertex i, $w(i,j)$ is the original edge weight between any two adjacent vertices i and j in the original graph (G_{KNN}), and $\sigma(i,j) \in [0,1]$ is the new, adjusted weight that takes structural similarity into account. Notice that the topology of the original graph is unchanged.

Proposed Method: In summary, the proposed method is as follows:

1. Take the pairwise similarity matrix used for hierarchical clustering of the dataset $\mathbf{X}$ and interpret it as a complete weighted similarity graph, G_s.
2. Build a spanning subgraph of G_s, G_{KNN}, keeping only the K largest edges incident to each vertex of G_s.
3. Adjust the edge weights of this graph (G_{KNN}) using the structural similarity measure in Equation (2), and call the resulting graph G_σ.
4. For every candidate cluster $\mathbf{C}_i$ in the cluster tree from which we want to extract a flat clustering solution of the data, compute the corresponding component of the modularity, $Q(\mathbf{C}_i)$, using G_σ as the reference graph.
5. Apply FOSC to the cluster tree using $Q(\mathbf{C}_i)$ as measure of cluster quality.

Density-Based Case — HDBSCAN*

Similarity Graph: HDBSCAN* works in the space of mutual reachability distances, so the similarity graph G_s in this case is the *mutual reachability graph*

$\mathbb{G}_{m_{pts}}$ used by the algorithm (see Sect. 3.1), however with its edges weights (distances) transformed into similarity (i.e., $w = 1 - d/d_{max}$).

Choice of K: For HDBSCAN*, the natural choice is $K = m_{pts}$, as there is no clear reason to argue in favor of the use of different neighborhood sizes during the clustering and cluster extraction stages. Notice that, with this setting, no extra computational burden is imposed to compute G_{KNN}, as HDBSCAN* internally computes the m_{pts} nearest neighbors of every object as part of the algorithm.

Noise as Singletons: Notice in Table 1 that some objects in a parent cluster may not be present in any of the children as they are noise at the density levels at which the children exist. Noise cannot be ignored when assessing the quality of extracted clusters, otherwise undesirable clustering solutions in which a large amount of objects are left unclustered as noise may be favored. A usual and effective solution to this problem is to represent each noise object as a *singleton* (a cluster on its own). The modularity naturally penalizes singletons as they are components with no internal connections, so the term $IS(\mathbf{C}_i)$ in (1) is zero.

Complexity: The spanning subgraph G_{KNN} can be computed simultaneously to HDBSCAN*, whose overall complexity is $O(n^2)$. This graph is sparse because $K = m_{pts}$ is a constant ($<< n$). Hence, G_σ can be computed in linear time. We assume that this sparse graph is implemented using an adjacency list, which allows any vertex and its adjacent vertices to be accessed in $O(1)$. We also assume the use of an auxiliary $n \times k$ binary matrix to check in $O(1)$ time whether a data object belongs to any given cluster $\mathbf{C}_i$ $(i = 1, \cdots, k)$ in the cluster tree. Based on these assumptions and the fact that, in the worst case, the HDBSCAN* complete hierarchy has at most n levels, each of which contains n labeled objects, it can be shown that the modularity component $Q(\mathbf{C}_i)$ for every cluster in the cluster tree, which is the information required to run FOSC, can be computed efficiently with a single bottom-up pass through the hierarchy, which is $O(n^2)$ in the worst case, i.e., it does not change HDBSCAN*'s complexity.

5 Experimental Results

Data Sets: For controlled experiments where we know the true probability distribution of clusters, we use the well-known benchmark collection from [26],[1] which contains 160 synthetic datasets with Gaussian and Ellipsoidal clusters. There are two sub-collections, one containing 80 low dimensional datasets (2D and 10D, called *Synthetic 1* here) and the other containing 80 high dimensional datasets (50D and 100D, called *Synthetic 2* here). The datasets have 4, 10, 20 and 40 clusters with cluster sizes varying from 10 to 500, depending on the number of clusters. For each combination of dimensionality and number of clusters there are 10 different datasets. We also used real data sets from different domains: "Articles-1442-5" and "Articles-1442-80" [27] as well as "Cbrilpirivson" and "Cbrilpirson" [28] consist of very high dimensional (4636D, 388D, 1431D,

[1] http://personalpages.manchester.ac.uk/staff/Julia.Handl/generators.html.

1423D resp.) representations of text documents (n varying from 253 to 945) belonging to 5 or 4 known categories. Cosine similarity has been used to cluster these datasets. “CellCycle-237” and “CellCycle-384” [29] as well as “YeastGalactose” [30] represent gene-expression data and contain the expression of genes ($n = 237, 384, 205$ resp.), belonging to 4 known classes, across 17 (CellCycle) or 20 (Yeast) conditions/dimensions. For these data sets we used Pearson distance. “Wine”, “Ecoli”, Glass, Iris and WDBC are from the UCI Repository [31]. For these data sets we used Euclidean distance. Finally, we have also used the image datasets “ALOI-TS88” and “ALOI-PCA” from [32], which contain, respectively, 88D and 6D descriptors of $n = 50$ to 125 images belonging to 2, 3, 4 or 5 known categories. For these data sets we also used Euclidean distance.

Parameters and Evaluation: We compare the performance of the proposed measure (hereafter referred to as *Mod-Knn*) against *Stability* when extracting clusters from HDBSCAN* with its parameter m_{pts} varying from 4 to 40 in steps of 4.[2] We evaluate the results against the ground truth using the *Adjusted Rand Index* — ARI [33], a standard measure in the clustering literature.

Summary of Findings: The first important observation is that, in the vast majority of the datasets, the relative performance between the two measures for a given dataset is stable across different values of m_{pts}, so looking only at the best result obtained by each measure in each dataset is a good representative of their relative performance for that dataset in general. For the **synthetic datasets**, Mod-Knn systematically obtained the best result for collection Synthetic 2 (high dimensional), and also for those datasets from collection Synthetic 1 (low dimensional) containing fewer clusters (4 or 10). The only scenario where Stability outperformed Mod-Knn is in lower dimensional datasets with larger number of clusters (20 or 40). Table 2 shows the mean ARI values for both measures and data collections across multiple m_{pts} values, where it is clear that Stability dominates the aggregated results for the low dimensional collection,[3] whereas Mod-Knn does so for the high dimensional one. We summarize the overall best results in Table 3 in terms of numbers of wins, losses and ties of each measure for each collection, alongside with the (non-parametric, paired) *Wilcoxon Signed-Rank Test*, which supports our findings above.

Table 2. Aggregated ARI values over the datasets of collections Synthetic 1 and 2.

Coll.	Measure	m_{pts}									
		4	8	12	16	20	24	28	32	36	40
Syn. 1	Stability	0.737	**0.797**	**0.777**	**0.764**	**0.744**	**0.722**	**0.704**	**0.682**	**0.671**	**0.662**
	Mod-Knn	**0.814**	0.751	0.717	0.689	0.662	0.645	0.634	0.616	0.605	0.601
Syn. 2	Stability	0.770	0.768	0.729	0.658	0.604	0.558	0.483	0.418	0.377	0.340
	Mod-Knn	**0.837**	**0.794**	**0.745**	**0.689**	**0.640**	**0.586**	**0.534**	**0.469**	**0.412**	**0.384**

[2] HDBSCAN* has an *optional* parameter m_{ClSize} that has not been used ($m_{\text{ClSize}} = 1$).

[3] An exception is $m_{pts} = 4$, where Mod-Knn has outperformed Stability in most cases.

Table 3. Summary of Results and Statistical Test.

Mod-Knn × Stability		
Dataset collection	Wins/Ties/Losses	p-value (Wilcoxon)
Real Datasets	8 / 2 / 4	0.1165
Synthetic 1	32 / 7 / 41	0.0207
Synthetic 2	60 / 19 / 1	< 0.01

As for **real data**, Mod-Knn has obtained the best overall result in 10 out of 14 datasets, 2 of which were also reached by Stability (a tie for the two "Articles" datasets). These include all the high dimensional (> 50D) ones. The only real datasets for which Stability outperformed Mod-Knn were CellCycle-237, Wine, Glass, and WDBC, of which the highest dimensional one is WDBC (32D).

Overall, our findings suggest that, in high dimensional spaces, the measure of Stability, which is strongly based on the density profile of clusters, becomes less reliable. The use of an auxiliary similarity graph and a similarity measure that also accounts for structural similarities, rather than relying solely on the similarities as measured in the original data space, has shown more robust performance, especially in high dimensional spaces where data typically lacks contrast both in terms of distances as well as in terms of density.

6 Conclusions

We proposed to adapt a variant of the *Modularity Q* measure, widely used in the realm of community detection in complex networks, so that it can be applied as an optimization criterion to the problem of optimal local cuts through clustering hierarchies. The results show that the proposed measure is a competitive alternative to the *Stability* measure, which has been successfully used by the state-of-the-art algorithms HDBSCAN* and FOSC. In particular, the proposed method systematically outperformed Stability in high dimensional datasets, where contrast in density is low and Stability has exhibited less robust behavior.

Acknowledgements. CNPq and CAPES (Brazil), NSERC (Canada).

References

1. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice-Hall Inc., Englewood Cliffs (1988)
2. Aggarwal, C.C., Reddy, C.K.: Data Clustering: Algorithms and Applications, 1st edn. Chapman & Hall/CRC, Boca Raton (2013)
3. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer, New York (2006)
4. Everitt, B.S., Landau, S., Leese, M.: Cluster Analysis. Oxford University Press, Oxford (2001)

5. Milligan, G.W., Cooper, M.C.: An examination of procedures for determining the number of clusters in a data set. Psychometrika **50**(2), 159–179 (1985)
6. Campello, R.J.G.B., Moulavi, D., Zimek, A., Sander, J.: A framework for semi-supervised and unsupervised optimal extraction of clusters from hierarchies. Data Min. Knowl. Discov. **27**(3), 344–371 (2013)
7. Campello, R.J.G.B., Moulavi, D., Zimek, A., Sander, J.: Hierarchical density estimates for data clustering, visualization, and outlier detection. ACM Trans. Knowl. Discov. Data **10**(1), 1–51 (2015)
8. Kriegel, H.P., Kröger, P., Sander, J., Zimek, A.: Density-based clustering. WIREs: Data Min. Knowl. Discov. **1**(3), 231–240 (2011)
9. Campello, R.J.G.B., Moulavi, D., Sander, J.: Density-based clustering based on hierarchical density estimates. In: PAKDD, pp. 160–172 (2013)
10. Piekenbrock, M., Hahsler, M.: HDBSCAN with the 'dbscan' package. https://cran.r-project.org/web/packages/dbscan/vignettes/hdbscan.html (nd)
11. McInnes, L., Healy, J., Astels, S.: The 'hdbscan' clustering library (Python Scikit-learn docs). http://hdbscan.readthedocs.io/en/latest/index.html (nd)
12. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E **69**(2), 026113 (2004)
13. Boudaillier, E., Hébrail, G.: Interactive interpretation of hierarchical clustering. Intell. Data Anal. **2**, 229–244 (1998)
14. Ferraretti, D., Gamberoni, G., Lamma, E.: Automatic cluster selection using index driven search strategy. In: AI*IA, pp. 172–181 (2009)
15. Gupta, G., Liu, A., Ghosh, J.: Automated hierarchical density shaving: a robust automated clustering and visualization framework for large biological data sets. IEEE/ACM Trans. Comp. Biol. Bioinform. **7**(2), 223–237 (2010)
16. Stuetzle, W.: Estimating the cluster tree of a density by analyzing the minimal spanning tree of a sample. J. Classif. **20**, 25–47 (2003)
17. Stuetzle, W., Nugent, R.: A generalized single linkage method for estimating the cluster tree of a density. J. Comp. Graph. Stat. **19**(2), 397–418 (2010)
18. Sander, J., Qin, X., Lu, Z., Niu, N., Kovarsky, A.: Automatic extraction of clusters from hierarchical clustering representations. In: PAKDD, pp. 75–87 (2003)
19. Bezdek, J.C., Pal, N.R.: Some new indexes of cluster validity. IEEE Trans. Syst. Man Cybern. Part B (Cybern.) **28**(3), 301–315 (1998)
20. Jaskowiak, P.A., Moulavi, D., Furtado, A.C., Campello, R.J., Zimek, A., Sander, J.: On strategies for building effective ensembles of relative clustering validity criteria. Knowl. Inf. Syst. **47**(2), 329–354 (2016)
21. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD, vol. 96, pp. 226–231 (1996)
22. Fortunato, S.: Community detection in graphs. Phys. Rep. **486**, 75–174 (2010)
23. Feng, Z., Xu, X., Yuruk, N., Schweiger, T.A.J.: A novel similarity-based modularity function for graph partitioning. In: Song, I.Y., Eder, J., Nguyen, T.M. (eds.) DaWaK 2007. LNCS, vol. 4654, pp. 385–396. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74553-2_36
24. Xu, X., Yuruk, N., Feng, Z., Schweiger, T.A.: Scan: a structural clustering algorithm for networks. In: KDD, pp. 824–833 (2007)
25. Huang, J., Sun, H., Song, Q., Deng, H., Han, J.: Revealing density-based clustering structure from the core-connected tree of a network. IEEE Trans. Knowl. Data Eng. **25**(8), 1876–1889 (2013)
26. Handl, J., Knowles, J.: An evolutionary approach to multiobjective clustering. IEEE Trans. Evol. Comput. **11**(1), 56–76 (2007)

27. Naldi, M.C., Campello, R.J.G.B., Hruschka, E.R., Carvalho, A.C.P.L.F.: Efficiency issues of evolutionary k-means. Appl. Soft Comput. **11**(2), 1938–1952 (2011)
28. Paulovich, F.V., Nonato, L.G., Minghim, R., Levkowitz, H.: Least square projection: a fast high-precision multidimensional projection technique and its application to document mapping. IEEE Trans. Vis. Comput. Graph. **14**, 564–575 (2008)
29. Yeung, K., Fraley, C., Murua, A., Raftery, A., Ruzzo, W.: Model-based clustering and data transformations for gene expression data. Bioinf. **17**(10), 977–987 (2001)
30. Yeung, K.Y., Medvedovic, M., Bumgarner, R.E.: Clustering gene-expression data with repeated measurements. Genome Biol. **4**(5), R34 (2003)
31. Lichman, M.: UCI machine learn. Repository (2013). http://archive.ics.uci.edu/ml
32. Horta, D., Campello, R.J.G.B.: Automatic aspect discrimination in data clustering. Pattern Recognit. **45**(12), 4370–4388 (2012)
33. Hubert, L., Arabie, P.: Comparing partitions. J. Classif. **2**, 193–218 (1985)

Statistics in Data Science

Positive Data Kernel Density Estimation via the LogKDE Package for R

Hien D. Nguyen[1](✉), Andrew T. Jones[2], and Geoffrey J. McLachlan[2]

[1] La Trobe University, Bundoora, VIC 3086, Australia
h.nguyen5@latrobe.edu.au
[2] University of Queensland, St. Lucia, QLD 4072, Australia

Abstract. Kernel density estimators (KDEs) are ubiquitous tools for nonparametric estimation of probability density functions (PDFs), when data are obtained from unknown data generating processes. The KDEs that are typically available in software packages are defined, and designed, to estimate real-valued data. When applied to positive data, these typical KDEs do not yield *bona fide* PDFs. A log-transformation methodology can be applied to produce a nonparametric estimator that is appropriate and yields proper PDFs over positive supports. We call the KDEs obtained via this transformation log-KDEs. We derive expressions for the pointwise biases, variances, and mean-squared errors of the log-KDEs that are obtained via various kernel functions. Mean integrated squared error (MISE) and asymptotic MISE results are also provided and a plug-in rule for log-KDE bandwidths is derived. We demonstrate the log-KDEs methodology via our *R* package, `logKDE`. Real data case studies are provided to demonstrate the log-KDE approach.

Keywords: Kernel density estimator · Log-transformation · Nonparametric · Plug-in rule · Positive data

1 Introduction

Let X be a random variable that arises from a distribution that can be characterized by an unknown density function $f_X(x)$. Assume that (A1) X is supported on $\mathbb{R}$, and (A2) $f_X(x)$ is sufficiently continuously differentiable (i.e. $\int_{\mathbb{R}} \left| f^{(m)}(x) \right| \mathrm{d}x < \infty$, where $f^{(m)}(x)$ is the mth derivative of $f(x)$, for $m \le M \in \mathbb{N}$).

Let $\{X_i\}_{i=1}^{n}$ be an independent and identically distributed (IID) sample of random variables, where each X_i is identically distributed to X ($i \in [n] = \{1, \dots, n\}$). Under conditions (A1) and (A2), a common approach to estimating $f_X(x)$ is via the kernel density estimator (KDE)

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right), \tag{1}$$

R. Islam et al. (Eds.): AusDM 2018, CCIS 996, pp. 269–280, 2019.
https://doi.org/10.1007/978-981-13-6661-1_21

which is constructed from the sample $\{X_i\}_{i=1}^n$. Here, $K(x)$ is a probability density function on $\mathbb{R}$ and is called the kernel function, and $h > 0$ is a tuning parameter that is referred to as the bandwidth. This approach was first proposed in the seminal works of [13] and [15].

Assume (B1) $\int_{\mathbb{R}} K(x)\,\mathrm{d}x = 1$, (B2) $\int_{\mathbb{R}} xK(x)\,\mathrm{d}x = 0$, (B3) $\int_{\mathbb{R}} x^2K(x)\,\mathrm{d}x = 1$, and (B4) $\int_{\mathbb{R}} K^2(x)\,\mathrm{d}x < \infty$, regarding the kernel function $K(x)$. Under conditions (A1) and (A2), [13] showed that (B1)–(B4) allowed for useful derivations of expressions for the mean squared error (MSE) and mean integrated squared error (MISE) between $f_X(x)$ and (1); see for example [18, Ch. 3] and [19, Ch. 24]. Furthermore, simple conditions can be derived for ensuring pointwise asymptotic unbiasedness and consistency of (1) (cf. [5, Sect. 32.7]). See [20] for further exposition regarding kernel density estimation (KDE).

The estimation of $f_X(x)$ by (1) has become a ubiquitous part of modern data analysis, data mining, and visualization (see e.g., [1, Sect. 8.6] and [25, Sect. 9.3]). The popularity of the methodology has made its implementation a staple in most data analytic software packages. For example, in the *R* statistical programming environment [14], KDE can be conducted using the core-package function `density`.

Unfortunately, when (A1) is not satisfied and is instead replaced by (A1*) X is supported on $(0,\infty)$, using a KDE of form (1) that is constructed, from a kernel that satisfies (B1)–(B4), no longer provides a reasonable estimator of $f_X(x)$. That is, if $K(x) > 0$ for all $x \in \mathbb{R}$, and (B1)–(B4) are satisfied, then $\int_0^\infty \hat{f}_X(x)\,\mathrm{d}x < 1$ and thus (1) is no longer a proper *bona fide* (PDF) over $(0,\infty)$. For example, this occurs when $K(x)$ is taken to be the popular Gaussian (normal) kernel function. Furthermore, expressions for MSE and MISE between $f_X(x)$ and (1) are no longer correct under (A1*) and (A2).

In [4], the authors proposed a simple and elegant solution to the problem of estimating $f_X(x)$ under (A1*) and (A2). First, let $Y = \log X$, $Y_i = \log X_i$ $(i \in [n])$, and $f_Y(y)$ be the PDF of Y. Note that if X is supported on $(0,\infty)$ then the support of $f_Y(y)$ satisfies (A1). If we wish to estimate $f_Y(y)$, we can utilize a KDE of form (1), constructed from $\{Y_i\}_{i=1}^n$, with a kernel that satisfies (B1)–(B4). If $f_Y(y)$ also satisfies (A2), then we can calculate the MSE and MISE between $f_Y(y)$ and (1).

Let W be a random variable and $U = G(W)$, where $G(w)$ is a strictly increasing function. If the distribution of U and W can be characterized by the PDFs $f_U(u)$ and $f_W(w)$, respectively, then the change-of-variable formula yields: $f_W(w) = f_U(G(w))\,G^{(1)}(w)$ (cf. [2, Thm. 3.6.1]). Utilizing the fact that $\mathrm{d}\log x/\mathrm{d}x = x^{-1}$, [4] used the aforementioned formula to derive the log-kernel density estimator (log-KDE)

$$\begin{aligned}\hat{f}_{\log}(x) &= x^{-1}\hat{f}_Y(\log x)\\ &= \frac{1}{nh}\sum_{i=1}^n x^{-1}K\left(\frac{\log x - \log X_i}{h}\right)\\ &= \frac{1}{n}\sum_{i=1}^n L(x; X_i, h)\,, \end{aligned} \tag{2}$$

where $L(x;z,h) = (xh)^{-1} K\left(\log\left[(x/z)^{1/h}\right]\right)$ is the log-kernel function with bandwidth $h > 0$, at location parameter z. For any $z \in (0,\infty)$ and $h \in (0,\infty)$, $L(x;z,h)$ has the properties that (C1) $L(x;z,h) \geq 0$ for all $x \in (0,\infty)$ and (C2) $\int_0^\infty L(x;z,h)\,\mathrm{d}x = 1$, when (B1)–(B4) are satisfied.

By property (C2), we observe that $\int_0^\infty \hat{f}_X(x)\,\mathrm{d}x = 1$, thus making (2) a *bona fide* PDF on $(0,\infty)$. Furthermore, using the expressions for the MSE and MISE between $f_Y(y)$ and (1), we can derive the relevant quantities for (2) as well as demonstrate its asymptotic unbiasedness and consistency.

For every kernel function that satisfies (B1)–(B4), there is a log-kernel function that satisfies (C1) and (C2), which generates a log-KDE that is a proper PDF over $(0,\infty)$. We have compiled an array of other potential pairs of kernel and log-kernel functions in Table 1. Throughout Table 1, the function $\mathbb{I}\{A\}$ takes value 1 if statement A is true and 0, otherwise.

Table 1. Pairs of kernel functions $K(y)$ and log-kernel functions $L(x;z,h)$, where $z \in (0,\infty)$ and $h \in (0,\infty)$.

Kernel	$K(y)$
Epanechnikov	$3\left(5-y^2\right)/\left(20\sqrt{5}\right)\mathbb{I}\left\{y \in \left(-\sqrt{5},\sqrt{5}\right)\right\}$
Gaussian (normal)	$(2\pi)^{-1/2}\exp\left(-y^2/2\right)$
Laplace	$\left(\sqrt{2}/2\right)\exp\left(-2^{1/2}\left\lvert y\right\rvert\right)$
Logistic	$\left(\pi/4\sqrt{3}\right)\operatorname{sech}^2\left(\pi y/2\sqrt{3}\right)$
Triangular	$\left(\sqrt{6}-\lvert y\rvert\right)6^{-1}\mathbb{I}\left\{y \in \left(-\sqrt{6},\sqrt{6}\right)\right\}$
Uniform	$\left(2\sqrt{3}\right)^{-1}\mathbb{I}\left\{y \in \left(-\sqrt{3},\sqrt{3}\right)\right\}$
Log-Kernel	$L(x;z,h)$
Log-Epanechnikov	$3\left(5-x^2\right)/\left(20\sqrt{5}xh\right)\mathbb{I}\left\{\log\left[(x/z)^{1/h}\right] \in \left(-\sqrt{5},\sqrt{5}\right)\right\}$
Log-Gaussian	$(2\pi)^{-1/2}(xh)^{-1}\exp\left(-\log^2\left[(x/z)^{1/h}\right]/2\right)$
Log-Laplace	$\left(\sqrt{2}/2\right)(xh)^{-1}\exp\left(-2^{1/2}\left\lvert\log\left[(x/z)^{1/h}\right]\right\rvert\right)$
Log-Logistic	$\left(\pi/4\sqrt{3}\right)(xh)^{-1}\operatorname{sech}^2\left(\pi\log\left[(x/z)^{1/h}\right]/2\sqrt{3}\right)$
Log-Triangular	$(xh)^{-1}\left(\sqrt{6}/6-\lvert x\rvert\right)6^{-1}\mathbb{I}\left\{\log\left[(x/z)^{1/h}\right] \in \left(-\sqrt{6},\sqrt{6}\right)\right\}$
Log-Uniform	$\left(2\sqrt{3}xh\right)^{-1}\mathbb{I}\left\{\log\left[(x/z)^{1/h}\right] \in \left(-\sqrt{3},\sqrt{3}\right)\right\}$

Unfortunately, out of all of the listed function pairs from Table 1, only the normal and log-Gaussian (log-normal) PDFs have been considered for use as kernel and log-kernel functions, respectively, for conducting log-KDE. The log-normal PDF was used explicitly for the construction of log-KDEs in [4], and more generally, for conducting asymmetric KDE on the support $(0,\infty)$, in [9]. Other works that have considered the log-normal PDF for conducting asymmetric KDE include [6–8], and [23]. In the R environment, asymmetric KDE with log-normal PDF as kernels has been implemented through the `dke.fun` function from the package `Ake` [23].

In this paper we firstly expand upon the theoretical results that were reported in [4], who derived expressions for the biases and variances between generic log-KDEs and their estimands. Here, we utilize general results for KDE of transformed data from [11] and [21]. We further derive a plug-in rule for the bandwidth h, which is similar to the famous rule of [18, Sect. 3.4]. Secondly, we introduce the readers to our R package `logKDE` [10, 12], which implements log-KDE in a manner that is familiar to users of the base R function `density`. Thirdly, we provide details regarding a set of simulation studies and demonstrate the use of the log-KDE methodology via a pair of example data sets.

The paper proceeds as follows. Theoretical results for log-KDE are presented in Sect. 2. Use of the `logKDE` package is described in Sect. 3. Numerical studies and examples are detailed in Sect. 4.

2 Theoretical Results

We start by noting that MSE and MISE expressions for the log-KDE with log-normal kernels have been derived by [9]. The authors have also established the conditions for pointwise asymptotic unbiasedness and consistency for the log-normal kernel. In the general case, informal results regarding expressions for the pointwise bias and variance, have been provided by [4]. In this section, we generalize the results of [9] and formalize the results of [4] via some previously known results from [20, Sect. 2.5] and [21]. In the sequel, we shall make assumptions (A1) and (A2) regarding $f_Y(y)$, (A1*) and (A2) regarding $f_X(x)$, and (B1)–(B4) regarding $K(y)$.

2.1 Pointwise Results

The following expressions are taken from [20, Sect. 2.5]. At any $y \in \mathbb{R}$, define the pointwise bias and variance between (1) and $f_Y(y)$ as

$$\begin{aligned}\operatorname{Bias}\left[\hat{f}(y)\right] &= \mathbb{E}\left[\hat{f}(y)\right] - f_Y(y) \\ &= \frac{1}{2}h^2 f_Y^{(2)}(y) + o\left(h^2\right),\end{aligned} \tag{3}$$

and

$$\operatorname{Var}\left[\hat{f}(y)\right] = \frac{1}{nh} f_Y(y) \int_{\mathbb{R}} K^2(z)\,\mathrm{d}z + o\left(\frac{1}{nh}\right), \tag{4}$$

respectively, where $a_n = o(b_n)$ as $n \to \infty$, if and only if $\lim_{n\to\infty} |a_n/b_n| = 0$. From expressions (3) and (4), and the change-of-variable formula, we obtain the following expressions for the bias, variance, and MSE between (2) and $f_X(x)$.

Proposition 1. *For any $x \in (0, \infty)$, the bias, variance, and MSE between (2) and $f_X(x)$ have the forms*

$$\begin{aligned} \mathit{Bias}\left[\hat{f}_{\log}(x)\right] &= \mathbb{E}\left[\hat{f}_{\log}(x)\right] - f_X(x) \\ &= \frac{h^2}{2}\left[f_X(x) + 3x f_X^{(1)}(x) + x^2 f_X^{(2)}(x)\right] + o\left(h^2\right), \end{aligned} \tag{5}$$

$$\mathit{Var}\left[\hat{f}_{\log}(x)\right] = \frac{1}{nhx} f_X(x) \int_{\mathbb{R}} K^2(z)\, dz + o\left(\frac{1}{nh}\right), \tag{6}$$

and

$$\begin{aligned} \mathit{MSE}\left[\hat{f}_{\log}(x)\right] &= \mathit{Var}\left[\hat{f}_{\log}(x)\right] + \mathit{Bias}^2\left[\hat{f}_{\log}(x)\right] \\ &= \frac{1}{nhx} f_X(x) \int_{\mathbb{R}} K^2(z)\, dz \\ &\quad + \frac{h^4}{4}\left[f_X(x) + 3x f_X^{(1)}(x) + x^2 f_X^{(2)}(x)\right]^2 \\ &\quad + o\left(\frac{1}{nh} + h^4\right), \end{aligned} \tag{7}$$

respectively

Consult cran.r-project.org/web/packages/logKDE/vignettes/logKDE.pdf for proofs of all theoretical results.

Let $h = h_n > 0$ be a positive sequence of bandwidths that satisfies the classical assumptions (D1) $\lim_{n\to\infty} h_n = 0$ and (D2) $\lim_{n\to\infty} nh_n = \infty$. That is, h_n approaches zero at a rate that is slower than n^{-1}. Under (D1) and (D2), we have obtain the pointwise unbiasedness and consistency of (2) as an estimator for $f_X(x)$.

Remark 1. *As noted by [4], the performance of the log-KDE method is most hindered by the behavior of the estimand $f_X(x)$, when $x = 0$, because of the $x^{-1} f_X(x)$ term in (6). If this expression is large at $x = 0$, then we can expect that the log-KDE will exhibit high levels of variability and a large number of observations n may be required in order to mitigate such effects. From the bias expressions (5), we also observe influences from expressions of form $x f_X^{(1)}(x)$ and and $x^2 f_X^{(2)}(x)$. This implies that there may be a high amount of bias when estimating $f_X(x)$ at values where x is large and $f_X(x)$ is either rapidly changing or the curvature of $f_X(x)$ is rapidly changing. Fortunately, in the majority of estimating problems over the domain $(0, \infty)$, both $f_X^{(1)}(x)$ and $f_X^{(2)}(x)$ tend to be decreasing in x, hence such effects should not be consequential.*

2.2 Integrated Results

We denote the asymptotic MISE between a density estimator and an estimand as the AMISE. From the general results of [21], we have the identity

$$\begin{aligned}\text{MISE}\left[\hat{f}_{\log}\right] &= \int_0^\infty \text{MSE}\left[\hat{f}_{\log}(x)\right]\mathrm{d}x \\ &= \text{AMISE}\left[\hat{f}_{\log}\right] + o\left(\frac{1}{nh} + h^4\right),\end{aligned}$$

where

$$\begin{aligned}\text{AMISE}\left[\hat{f}_{\log}\right] = &\frac{1}{nh}\mathbb{E}\left[X^{-1}\right]\int_{\mathbb{R}} K^2(z)\,\mathrm{d}z \\ &+\frac{h^4}{4}\int_0^\infty \left[f_X(x) + 3xf_X^{(1)}(x) + x^2 f_X^{(2)}(x)\right]^2 \mathrm{d}x.\end{aligned}$$

By a standard argument

$$\begin{aligned}h^* &= \arg\inf_{h>0}\ \text{AMISE}\left[\hat{f}_{\log}\right] \qquad (8)\\ &= \left[\frac{\mathbb{E}\left[X^{-1}\right]\int_{\mathbb{R}} K^2(z)\,\mathrm{d}z}{\int_0^\infty \left[f_X(x) + 3xf_X^{(1)}(x) + x^2 f_X^{(2)}(x)\right]^2 \mathrm{d}x}\right]^{1/5} n^{-1/5},\end{aligned}$$

and

$$\inf_{h>0}\ \text{AMISE}\left[\hat{f}_{\log}\right] = \frac{5}{4}\left[\int_{\mathbb{R}} K^2(z)\,\mathrm{d}z\right]^{4/5} Jn^{-4/5}, \qquad (9)$$

where

$$J = \left(\mathbb{E}^4\left[X^{-1}\right]\int_0^\infty \left[f_X(x) + 3xf_X^{(1)}(x) + x^2 f_X^{(2)}(x)\right]^2 \mathrm{d}x\right)^{1/5}.$$

Using expression (8), we can derive a plugin bandwidth estimator for common interesting pairs of kernels $K(y)$ beand estimands $f_X(x)$. For example, we may particularly interesting in obtaining an optimal bandwidth h^* for scenario where we take $K(y)$ to be normal and $f_X(x)$ to be log-normal with scale parameter $\sigma^2 > 0$ and location parameter $\mu \in \mathbb{R}$. This scenario is analogous to the famous rule of thumb from [18, Sect. 3.4].

Proposition 2. *Let $K(y)$ be normal, as per Table 1 and let $f_X(x)$ be log-normal, with the form*

$$f_X(x) = \frac{1}{x\sqrt{2\pi\sigma^2}}\exp\left(-\frac{1}{2\sigma^2}\left[\log x - \mu\right]^2\right).$$

If we estimate $f_X(x)$ by a log-KDE of form (2), then the bandwidth that minimizes $AMISE\left[\hat{f}_{\log}\right]$ is

$$h^* = \left[\frac{8\exp\left(\sigma^2/4\right)}{\sigma^4 + 4\sigma^2 + 12}\right]^{1/5} \frac{\sigma}{n^{1/5}}, \tag{10}$$

and

$$\inf_{h>0} \quad AMISE\left[\hat{f}_{\log}\right] = \frac{5}{16}\left(\frac{2}{\pi}\right)^{2/5} n^{-4/5} J,$$

where

$$J = \frac{1}{2}\frac{\exp\left(9\sigma^2/20\right)\left(\sigma^4 + 4\sigma^2 + 12\right)^{1/5}}{\pi^{1/10}\sigma}.$$

Remark 2. *In general, one does not know the true estimand $f_X(x)$, or else the problem of density estimation becomes trivialized. However, as a guideline, the log-normal density function can be taken as reasonably representative with respect to the class of densities over the $(0, \infty)$. As such, the plugin bandwidth estimator (10) can be used in order to obtain a log-KDE with reasonable AMISE value. Considering that the true parameter value σ^2 is also unknown, estimation of this quantity is also required before (10) can be made useful. If $\{X_i\}_{i=1}^n$ is a sample that arises from a log-normal density with parameters σ^2 and μ, then $\{Y_i\}_{i=1}^n$ ($Y_i = \log X_i$, $i \in [n]$) is a sample that arises from a normal density with the same parameters. Thus, faced with $\{X_i\}_{i=1}^n$, one may take the logarithmic transformation of the data and compute the sample variance of the data to use as an estimate for σ^2. Alternatively, upon taking the logarithmic transformation, any estimator for σ^2 with good properties can be used. For example, one can use the interquartile range divided by 1.349^2.*

Rule (10) is by no means the only available technique for setting the bandwidth h when performing log-KDE. An alternative to using rule (10) is to utilize the classic rule from [18, Sect. 3.4], based on minimizing the AMISE with respect to the estimator of form (1) using normal kernels, for estimating normal densities.

Apart from the two aforementioned plugin bandwidth estimators, we can also utilize more computationally intensive methodology for choosing the bandwidth h, such as cross-validation (CV) procedures that are discussed in [18, Ch. 3] or the improved efficiency estimator of [17]. The implementations of each of the mentioned methods for bandwidth selection in the `logKDE` package are discussed in further detail in the following section.

3 The `logKDE` package

The `logKDE` package can be installed from github and loaded into an active *R* session using the following commands:

```
install.packages("logKDE", repos='http://cran.us.r-project.org')
```

The logKDE package seeks replicate the syntax and replicate the functionality of the KDE estimation function, density, built into the base *R* stats package. The two main functions included in the package, logdensity and logdensity_fft, both return Density objects, which are of the same form as those produced by density. This enables the efficient reuse of functions such as plot and print, included in the stats package. Descriptions of all logKDE functions can be found in the manual [12].

3.1 Kernels

All of the kernels described in Table 1 are available in logKDE. They can be chosen via the kernel parameter in logdensity and logdensity_fft. The different options are epanechnikov, gaussian, laplace, logistic, triangular, and uniform. Note that uniform is referred to as rectangular in stats. By default we set kernel='gaussian' (i.e., log-normal). This choice was made to conform with the default settings of the density function.

3.2 Bandwidth Selection

The available Bandwidth selection methods with the logKDE package include all of those from stats as well as two new bandwidth (BW) methods. The available methods are bcv, bw.logCV, bw.logG, bw.SJ, nrd0, nrd, and ucv. The equation of [18] is used to compute the nrd0 bandwidth. The nrd bandwidth is the same as nrd0, except for a multiplicative constant. The bandwidths ucv and bcv are computed as per the descriptions of [16], performed on the log-transformed data. The bw.SJ bandwidth is computed as per the description of [17], and bw.logG bandwidth utilizes Eq. (10). Finally, bw.logCV computes unbiased cross-validation bandwidths using untransformed data, rather than the log-transformed data that are used by ucv; see [4] for details.

3.3 Plotting and Visualization

The reuse of functionality from base *R* packages allows intuitive visualization of the densities estimated using logKDE. This is illustrated in the following simple example (see Fig. 1):

```
> fit1 <- logdensity(rchisq(100,10))
> plot(fit1)

> print(fit1)
Call:
    logdensity(x = chisq10)
Data: chisq10 (100 obs.);   Bandwidth 'bw' = 0.1546
       x                 y
 Min.   : 1.966   Min.   :0.003108
 1st Qu.: 8.007   1st Qu.:0.008810
```

```
Median :14.048   Median :0.034050
Mean   :14.048   Mean   :0.040812
3rd Qu.:20.089   3rd Qu.:0.071367
Max.   :26.131   Max.   :0.094840
```

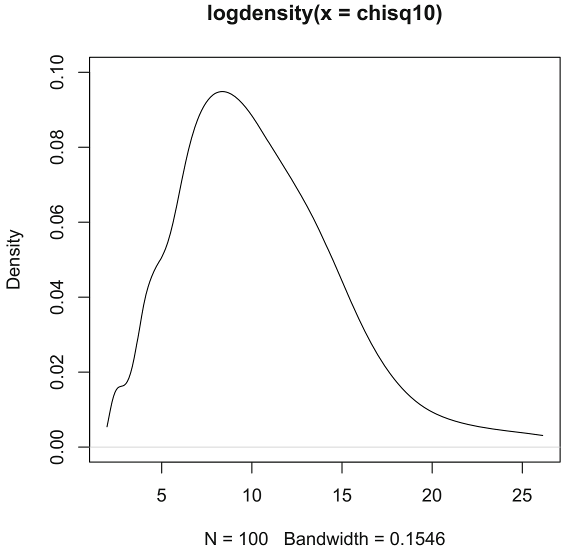


Fig. 1. A basic example of the use of `logdensity` class from *R* package `logKDE`.

The shared syntax and class structure between `logdensity` and `density` allows for the simple creation of more complex graphical objects. Additionally, via a range of settings and options, different bandwidth and kernel preferences can be easily accessed (see Fig. 2):

```
> fit<-logdensity(chisq10, bw =‘logCV’, kernel = ‘triangular’)
> plot(fit, ylim=c(0, 0.1))
> grid(10,10,2)
> x<-(seq(min(chisq10), max(chisq10), 0.01))
> lines(x, dchisq(x,10), col = 4)
```

4 Numerical Results

4.1 Simulation Studies

A comprehensive set of simulation studies was conducted, based on the simulation studies of [4]. The performance of the `logKDE` package was compared with those of the methods from `stats` and `Conake` [22]. The performances of the kernel density estimators from the various packages were compared via the average MISE and MIAE criteria. The results of the simulation studies that have been described are reported in Tables 2–7 of the `logKDE` vignette.

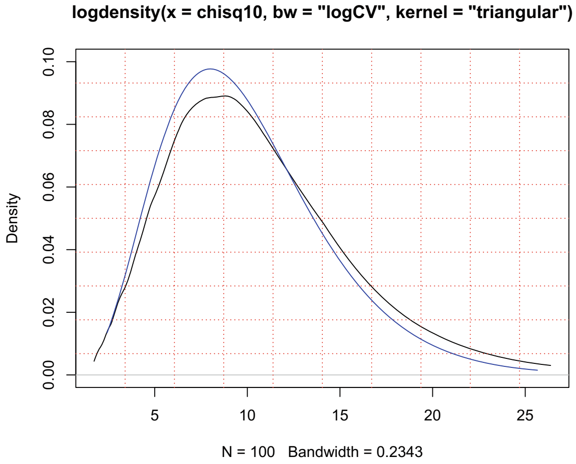


Fig. 2. Another example of the use of `logdensity` class from *R* package `logKDE`. In this case, the bandwidth is selected using the CV method and a triangular kernel is used. The χ^2_{10} reference distribution is marked in blue, whereas the log-KDE is plotted in black.

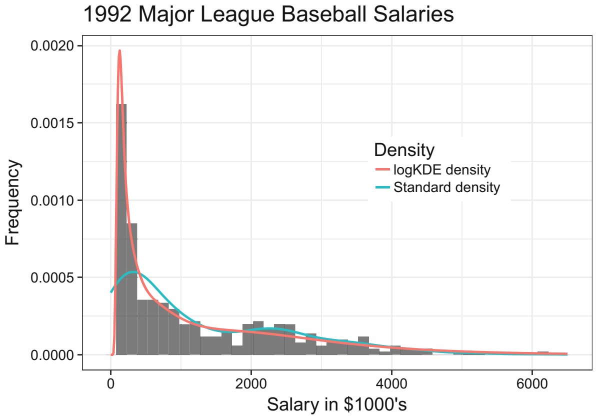


Fig. 3. Histogram and KDEs of the Baseball Salary data from [24].

4.2 Case Studies with Real Data

Our first illustrative example of the relative performance of the log-KDE method compared with standard KDE is provided using data taken from [24]. These data comprise of 331 salaries of Major League Baseball players for the 1992 season. Both densities were constructed using the default settings of the respective packages and both were estimated over the range [0.0001, 6500]. As can be seen in Fig. 3, the log-KDE estimate is qualitatively closer to the histogram of the actual data, particularly for values that are close to the origin.

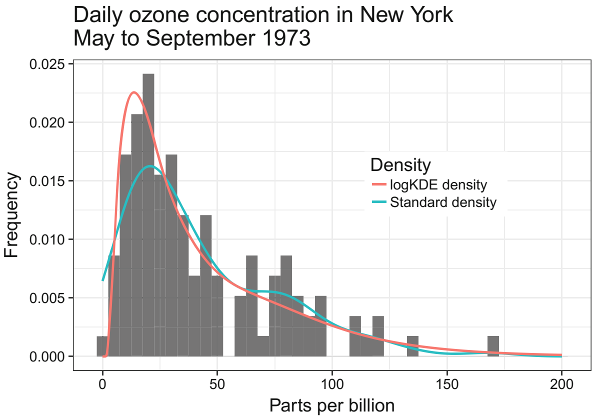


Fig. 4. Histogram and KDEs of the daily ozone concentration data taken from the air quality dataset in [3].

Another famous set of positive data is the daily ozone level data taken from a wider air-quality study [3]. The data consist of 116 daily measurements of ozone concentration in parts per billion taken in New York City, between May and September, 1973. The default settings and log-normal kernels were used for both estimators, which covered the range [0.0001, 200]. As with the baseball data, the fidelity of the kernel density estimate is improved, close to the origin (Fig. 4).

References

1. Aggarwal, C.C.: Data Mining. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-14142-8
2. Amemiya, T.: Introduction to Statistics and Econometrics. Harvard University Press, Cambridge (1994)
3. Chambers, J.M., Cleveland, W.S., Kleiner, B., Tukey, P.A.: Graphical Methods for Data Analysis. Wadsworth, Belmont (1983)
4. Charpentier, A., Flachaire, E.: Log-transform kernel density estimation of income distribution. L'Actualite Economique **91**, 141–159 (2015)
5. DasGupta, A.: Asymptotic Theory Of Statistics And Probability. Springer, New York (2008). https://doi.org/10.1007/978-0-387-75971-5
6. Hirukawa, M., Sakudo, M.: Nonnegative bias reduction methods for density estimation using asymmetric kernels. Comput. Stat. Data Anal. **75**, 112–123 (2014)
7. Igarashi, G.: Weighted log-normal kernel density estimation. Commun. Stat. - Theory Methods **45**, 6670–6687 (2016)
8. Igarashi, G., Kakizawa, Y.: Bias corrections for some asymmetric kernel estimators. J. Stat. Plan. Inference **159**, 37–63 (2015)
9. Jin, X., Kawczak, J.: Birnbaum-Saunders and lognormal kernel estimators for modelling durations in high frequency financial data. Ann. Econ. Financ. **4**, 103–124 (2003)

10. Jones, A.T., Nguyen, H.D., McLachlan, G.J.: logKDE: log-transformed kernel density estimation. J. Open Source Softw. **3**, 870 (2018)
11. Marron, J.S., Ruppert, D.: Transformations to reduce boundary bias in kernel density estimation. J. R. Stat. Soc. B **56**, 653–671 (1994)
12. Nguyen, H.D., Jones, A.T., McLachlan, G.J.: logKDE: computing log-transformed kernel density estimates for postive data (2018). cran.r-project.org/package=logKDE
13. Parzen, E.: On estimation of a probability density function and mode. Ann. Math. Stat. **33**, 1065–1076 (1962)
14. R Core Team: R: a language and environment for statistical computing. R Foundation for Statistical Computing (2016)
15. Rosenblatt, M.: Remarks on some nonparametric estimates of a density function. Ann. Math. Stat. **27**, 832–835 (1956)
16. Scott, D.W., Terrell, G.R.: Biased and unbiased cross-validation in density estimation. J. Am. Stat. Assoc. **82**(400), 1131–1146 (1987)
17. Sheather, S.J., Jones, M.C.: A reliable data-based bandwidth selection method for kernel density estimation. J. R. Stat. Soc. B **53**, 683–690 (1991)
18. Silverman, B.W.: Density Estimation for Statistics and Data Analysis. Chapman and Hall, London (1986)
19. van der Vaart, A.: Asymptotic Statistics. Cambridge University Press, Cambridge (1998)
20. Wand, M.P., Jones, M.C.: Kernel Smoothing. Springer, New York (1995)
21. Wand, M.P., Marron, J.S., Ruppert, D.: Transformations in density estimation. J. Am. Stat. Assoc. **86**, 343–353 (1991)
22. Wansouwé, W.E., Libengué, F.G., Kokonendji, C.C.: Conake: Continuous Associated Kernel Estimation (2015). CRAN.R-project.org/package=Conake
23. Wansouwé, W.E., Some, S.M., Kokonendji, C.C.: Ake: an R package for discrete and continuous associated kernel estimations. R Journal **8**, 258–276 (2016)
24. Watnik, M.R.: Pay for play: are baseball salaries based on performance? J. Stat. Educ. **6**, 1–5 (1998)
25. Witten, I.H., Frank, E., Hall, M.A., Pal, C.J.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, Amsterdam (2017)

The Role of Statistics Education in the Big Data Era

Ryan H. L. Ip(✉)

School of Computing and Mathematics, Charles Sturt University,
Wagga Wagga, NSW 2650, Australia
hoip@csu.edu.au

Abstract. With the increasing availability and trendiness of "big data", data science has become a fast growing discipline. Data analysis techniques are shifting from classical statistical inferences to algorithmic machine learnings. Will the rise of data science lead to the fall of statistics? If education is the key to defend statistics as a discipline, what should statisticians teach to respond to the challenges brought by big data? This paper aims to provide the current situation of data science and statistics programs within the higher education sector in Australia and some personal thoughts on statistics education in this era.

Keywords: Statistics teaching · Data science · Small data

1 Introduction

Since the emergence of "big data", there has been a long debate on the role of statistics in the modern world flooded with data. I guess most people would agree that we are living in the era of big data. The term"big data" is not only trendy within academia, but is also frequently used by the media, companies and even ordinary people. Recently, *Statistics & Probability Letters (SPL)* published a special issue "The role of Statistics in the era of Big Data" (Volume 136). This special issue collects inspiring opinions from statisticians, computer scientists, data miners and other experts from related domains. A key message brought out from the special issue is that - statistics is still a central part of data science (or science, in general), despite the fact that some argue that statistics - especially sampling and modelling - is no longer relevant when sufficient data are obtained [1].

Although there does not seem to have a consensus on the definition of "big data" and the difference between "big data" and "small data", it may be useful to be more specific. Along the lines of [8], the term "big data" is used to refer to datasets with large number of observations (with possibly large number of variables as well) that are usually collected from observational studies, and are even too large to be handled by machines. In contrast, the term "small data" is used to represent datasets that result from designed experiments and are possible to be handled manually, or at most with an ordinary computer. Statistical

R. Islam et al. (Eds.): AusDM 2018, CCIS 996, pp. 281–288, 2019.
https://doi.org/10.1007/978-981-13-6661-1_22

techniques, at least the classical ones, mainly focus on analysing small data while the focus of "data science" is usually on extracting information from big data using data mining or machine learning techniques. Despite saying so, the two disciplines are, and will never be, mutually exclusive. As observed by Reid [14], some statistics departments or schools in colleges or universities have renamed themselves to integrate with "data science" or "data analytics". Such a movement is likely to be strategic to secure funding and student number due to the fact that statistics does not seem to have a good fame [9] while "big data" or "data science" are labelled as "sexy" [4]. In fact, I would not be surprised to see statistics, as a discipline, will be merged with, replaced by, or placed under, data science in the next five to ten years. In order to defend statistics as a discipline, [11,14,16], among others, indicate the importance of statistics teaching. In particular, Piecresare Secchi called a debate on "how we should teach the next generation of statisticians" [16, p.11]. It is of course important to teach statistics to the next generation of statisticians. Yet, we must not ignore the fact that it is equally important to teach statistics to the next generation of scientists, businesspeople, educators, among many other domains, as we must rely on these people to rebuild the reputation of statistics.

This paper attempts to enrich the discuss on statistics teaching in the era of big data by describing what is happening within Australia and suggesting some statistical ideas and concepts that should be emphasised by statistics educators to respond to the opportunities and challenges brought by big data.

2 The Rise of Data Science

As a discipline, data science is a very young one. The growth is, however, rapid. With an aim to know more about the rise of data science, data science programs and statistics programs offered by universities in Australia were compared. In this paper, only Master programs were considered.

In Australia alone, 30 out of 42 universities have offered the Master of Data Science (MDataSc) program or similar (including similar names such as Master of Data Analytics or Master of Predictive Analytics, and specialisations/streams under various Master programs such as Master of Information Technology, Master of Science, Master of Computer Science or Master of Business Administration). Where data are available, all these programs were launched in or after 2015. The full list can be found in Table 1. Even if specialisations/streams are excluded, there are still 21 universities on the list. In the opposite, Table 1 shows that only seven universities are offering Master of Statistics (MStat) or Master of Applied Statistics. Combining also Master of Biostatistics, Master of Medical Statistics and various master programs with statistics as an optional specialisation, there are 17 options. Considering the "ages" of the two disciplines, the difference is even more striking.

Such a phenomenon is also observed in other countries/ areas. For example, being on of the leaders in statistics in Asia, the University of Hong Kong is launching the Master of Data Science program in 2018. James [11] has reported

Table 1. Master programs on statistics and data science offered by universities in Australia.

University[a]	MStats or similar	MDataSc or similar (Year commenced)[b]
Adel.	MBiostatistics	MDataSc (2018)
ANU	MStat	MAppDataAnalyt (2016)
Aust.Cath.	–	–
Bond	–	Under MBA (2017)
Canb.	–	MBusAnalyt (2015)
Carnegie Mellon (Aust.)	–	Under MIT (Unknown)
C.Qld.	–	MDataSc (2017)
C.Darwin	–	MDataSc (2018)
C.Sturt	–	–
Curtin	–	MPredAnalyt (2017)
Deakin	–	MDataAnalyt (2016)
Divinity	–	–
E.Cowan	–	–
FedUni	–	–
Flin.	–	–
Griff.	–	–
James Cook	–	MDataSc (2017)
La Trobe	MStat	MDataSc (2016)
Macq.	MStat	MDataSc (2018)
Melb.	Under MSc	MDataSc (2017)
Monash	MBiostatistics	MDataSc (2016)
Murd.	–	–
Newcastle(NSW)	MMedStat	Under MIT (Unknown)
Notre Dame Aust.	–	–
Qld.	MBiostatistics	MDataSc (2017)
Qld. UT	MBiostatistics	–
RMIT	MStat	MDataSc (2017)
S.Aust.	–	MDataSc (2018)
S.Cross	–	–
S.Qld.	Under MSc	Under MSc (2015)
Sunshine Coast	–	MInfCommunTech (2017)
Swinburne UT	MStat	Under MIT (2015)
Syd.	MBiostatistics	MDataSc (2016)
Tas.	–	Under MIT (2017)
Technol.Syd.	Under MSc	MDataScInn (2015)
Torrens (Aust.)	–	–
UNE	–	MDataSc (2018)
UNSW	MStat	Under MIT (2015)
Vic.(Melb)	–	–
W.Aust.	MMathStatSc	MDataSc (2017)
W'gong	MStat	Under MCompSc (2018)
W.Syd.	–	MDataSc (2016)

[a] Where available, the abbreviations are based on the list of Institution & Qualification Abbreviations from The University of Queensland. url: http://qual.app.uq.edu.au/institutions/. Last assessed: 25 July 2018.

[b] The year of commencement is based on the handbooks, new presses and/or brochures of the university.

a similar trend in the United States. While the supply may not necessarily reflect the demand, the abundance of the MDataSc programs indicates, at least, the confidence of student intakes (and hence revenues) from the universities' points of view.

In terms of the contents, MStat programs often focus on advanced modelling techniques and probability theories. Typical subjects cover gerenalised linear models, time series models, longitudinal analysis, measure theory, multivariate analysis, survival analysis, non-parametric statistics and Bayesian techniques. Besides, MStat programs often involve some mathematics subjects such as partial differential equations and real analysis. Notably, data mining is also a commonly seen subject in MStat programs. On the other hand, MDataSc programs often focus on algorithmic learning. Typical subjects cover database management, programming, cloud computing, data mining, machine learning, visual or graphical analytics and artificial intelligence. In addition, MDataSc programs usually involve one or more statistics subjects.

Although the contents of MDataSc and MStat are different, it is likely that an interested potential student will not enrol in both programs. With a larger number of options and a "sexier" title, perhaps data science has won the battle if we consider the market as a battlefield. Thus, the rise of data science may in turn lead to the shrinkage of statistics as a discipline.

Is it a threat? Should statisticians, especially the young ones who have no plan to retire in the next ten years, be worried? The answer is perhaps "yes, but not now". As aforementioned, most MDataSc programs involve one or more statistics subjects at the moment, which indicates that statistical thinking and techniques are still a central part, or a foundation, of many techniques in data science. Statistics subjects may still serve as "service subjects" in data science programs like in all other disciplines such as education and science. Moreover, a statistician may still claim to be a data scientist until there is a proper definition and a clear distinction. At the moment, statistics can still sail along the data science boat.

Nonetheless, statisticians should step up and act fast to defend the discipline and rebuild the brand name, before it is too late. Apart from research, which is the main perspective of the papers in the special issue in *SPL*, statisticians should grasp the opportunity of teaching statistics, even at the introductory levels, to stress the importance of statistics in this era. Quality teaching is certainly required [22] but the context is also important.

3 What Should We Teach?

In a world full of big data, what should statisticians offer to the students, especially the undergraduates who do not major in statistics, to help them appreciate statistics? Admittedly, this is a challenging task. This section is devoted to list several important statistical concepts or topics that shall be emphasised. The list is far from being complete but hopefully it can shed some light and promote further discussions. Moreover, some of the topics may be more suitable for introductory level subjects while some are more suitable for advanced subjects.

3.1 Small Data Are Still Prevalent

First and foremost, educators should let students know that small data are still commonly seen and analysed. Despite the fact that many people are talking about big data and big data are becoming more and more available, such datasets are not as prevalent as one may perceived.

Articles in the first issues in 2018 for four journals: *Crop and Pasture Science*, *Australian Education Researchers*, *Rural and Remote Health* (only original research articles were reviewed) and *Journal of Diabetes Investigation* were reviewed. These are leading journals in their respective subject areas in Australia according to *Scimago Journal & Country Rank* [15]. The review focused on the sample sizes considered in these articles. Purely qualitative studies, case studies, letters to editors and editorials were excluded. Following [12], sample sizes over 10,000 were considered to be "large". Out of the 44 relevant articles reviewed, 38 (84%) of them considered samples of sizes less than 10,000. Among the four journals, articles published in *Crop and Pasture Science* considered large datasets the most, which mainly came from genomic studies. Other large datasets considered in other journals arose from meta-analyses. In the opposite, none of the articles in *Australian Education Researchers* considered large datasets as the student numbers in the studies were often limited. Although the list in Table 2 is by no means complete and representative, it shows that small data are still commonly analysed in the literature even though we are living the era of big data.

Table 2. Number of articles with small ($n \leq 10000$) or large ($n > 10000$) sample sizes in the first issues of four journals in 2018.

Subject area	Journal	$n \leq 10,000$	$n > 10,000$
Agriculture	*Crop and Pasture Sci.*	5	3
Education	*Aust. Educ. Res.*	2	0
Health	*Rural Remote Health*	7	2
Medicine	*J. Diabetes Invest.*	23	2

It is not hard to understand why small data are still prevalent in the literature. The existence of big data does not mean the data are available to everyone (scholars included) since the collection of big data often involve a huge cost. Even if the data can be collected with a low cost, various reasons such as privacy issues limit the availability. In fact, around two-third of the datasets (280 out of 410 quantitative datasets) in the UCI Machine Learning Repository [5], which are often used to test various data mining algorithms, contain less than 10,000 observations. Since small data are still widespread, students from various disciplines need to learn the right tool – statistics. I am not denying the importance of machine learning techniques, but statistical techniques are simply more relevant in analysing small data.

3.2 Quality Beats Quantity

Given that small data are still prevalent, educators should emphasize the quality of data rather than the quantity while teaching statistics. It is always better to have small data with high quality than big data with low quality [8]. Quality datasets should at least be representative to the desired target population. As aforementioned mentioned, big data often come from observational studies rather than designed experiments. Without a properly designed collection process, there is a risk that the sample does not represent the target population [6]. The analysis results may thus be biased and cannot be generalised to the target population. Worse still, such a bias may not be obvious and is often hard to quantify. From this angle, collection of data is of high importance. Experimental design and sampling would be the most relevant topics for students.

3.3 Estimations Rather Than Hypothesis Tests

Classical statistics always focus on hypothesis tests. From my personal experience, statistics educators (including myself) tend to teach students to follow a routine (often a five- to seven-step process) in conducting hypothesis tests and to compare the p-value with a pre-specified threshold (usually 5%). In this way, students, especially whose mathematical abilities are not strong, can at least get some marks in the assessments. However, such a practice of teaching is likely to lead to confused ideas about hypothesis tests and various bad consequences such as "p-hacking" [18] and misinterpretation and over-reliance of p-values [20]. Moreover, in the big data era, when the sample size is large enough, everything will be statistically significant [12].

In fact, the usage of hypothesis tests and p-values have long been doubted by many scholars (see, e.g. [2]). Various remediations have been suggested, ranging from lowering the significance level [10] to declaring an end for p-values [7]. Completely moving away from hypothesis tests will not come to a success unless there is a radical change in the way how statistics is taught [19]. The need to focus on estimation and practical significance rather than hypothesis tests and statistical significance while teaching statistics has never been greater due to the emergence of big data.

3.4 Importance of Assumptions

Almost all classical statistical techniques were built upon a set of assumptions. Some assumptions are strict while some are relatively weaker. Even for small datasets, diagnostic checks are rarely mentioned or reported in the literature. Statistical results may simply become invalid when one or more assumptions are not met. The situation escalates when big data are analysed [3]. The omittance of assumption checking in research reports and even journal articles also prohibits readers to assess the validity of the results.

"Assume assumptions are met" is a common phrase in various statistics texts and lecture notes. Such an assumption is too optimistic to be true in practice, especially when big data are involved. Statistics educators should instead encourage students to critically challenge the assumptions more often. Examples where assumptions were not satisfied and the consequences should appear more frequently in texts and teaching materials.

3.5 Dependency Structures

Related to the previous point, independence is often an important assumption in classical statistical techniques. Yet, observations in large datasets are rarely truly independent [13]. The effect of dependencies among observations would be substantial. For example, in spatial analysis, ignoring the positive correlation among observations often leads to underestimation of the standard error and thus narrowing any confidence intervals formed [17]. In practice, the dependency structure may be far less obvious and far more complicated. For example, a set of medical record may contain individuals who belong to the same family. However, such an important piece of information may be masked when the identities were removed due to privacy issues. As described by Wit, while we have never been closer to the long-fancied situation, namely $n \to \infty$, the "bigness" of data actually only leads to the "death" of classical asymptotic results [21].

Thus, when time and resources permit, statistical techniques which handle dependent variables should be included in the syllabus. Topics may include time series, spatial statistics and multivariate statistics. It does not mean results relying on the assumption of independence should not be taught. Undoubtedly these results form the foundation of more complicated theories. However, it is perhaps time to rethink the curriculum and spare time for the above topics.

4 Conclusion

To defend the shrinking discipline, education is the key. Quality teaching and state-of-the-art contents that respond to the needs of the era are needed. Hopefully this short paper would motivate more discussions on what should be taught and how should we teach.

Acknowledgement. Helpful comments from three reviewers are greatly appreciated.

References

1. Anderson, C.: The End of Theory: The Data Deluge Makes the Scientific Method Obsolete (2008). https://www.wired.com/2008/06/pb-theory/. Accessed 26 July 2018
2. Berger, J.O., Sellke, T.: Testing a point null hypothesis: the irreconcilability of P value and evidence. J. Am. Stat. Assoc. **82**, 112–122 (1987)
3. Cox, D.R.: Big data and precision. Biometrika **102**, 712–716 (2015)
4. Davenport, T., Patil, D.: Data scientist: the sexiest job of the 21st century. Harvard Bus. Rev. **90**, 70–76 (2012)

5. Dua, D., Karra Taniskidou, E.: UCI machine learning repository (2017). http://archive.ics.uci.edu/ml. Accessed 6 Oct 2018
6. Dunson, D.B.: Statistics in the big data era: failures of the machine. Stat. Probabil. Lett. **136**, 4–9 (2018)
7. Evans, S.J., Mills, P., Dawson, J.: The end of the *p* value? Brit. Heart J. **60**, 177–180 (1988)
8. Faraway, J., Augustin, N.: When small data beats big data. Stat. Probabil. Lett. **136**, 142–145 (2018)
9. Gal, I., Ginsburg, L.: The role of beliefs and attitudes in learning statistics: towards an assessment framework. J. Stat. Educ. **2**, 2 (1994). https://doi.org/10.1080/10691898.1994.11910471
10. Ioannidis, J.: The proposal to lower *p* value threshold to.005. J. Am. Med. Assoc. **319**, 1429–1430 (2018)
11. James, G.: Statistics within business in the era of big data. Stat. Probabil. Lett. **136**, 155–159 (2018)
12. Lin, M., Lucas Jr., H., Shmueli, G.: Too big to fail: large samples and the *p*-value problem. Inform. Syst. Res. **24**, 906–917 (2013)
13. Meinshausen, N., Bühlmann, P.: Maximin effects in inhomogeneous large-scale data. Ann. Statist. **43**, 1801–1830 (2015)
14. Reid, N.: Statistical science in the world of big data. Stat. Probabil. Lett. **136**, 42–45 (2018)
15. SCImago: SJR-SCImago Journal & Country Rank. http://www.scimagojr.com. Accessed 26 July 2018
16. Secchi, P.: On the role of statistics in the era of big data: a call for a debate. Stat. Probabil. Lett. **136**, 10–14 (2018)
17. Sherman, M.: Spatial Statistics and Spatio-Temporal Data: Covariance Functions and Directional Properties. Wiley, New York (2011)
18. Simmons, J.P., Nelson, L.D., Simonsohn, U.: False-positive psychology: undisclosed flexibility in data collection and analysis allows presenting anything as significant. Psychol. Sci. **22**, 1359–1366 (2011)
19. Sterne, J.: Teaching hypothesis tests - time for significant change? Statist. Med. **21**, 985–994 (2002)
20. Wasserstein, R., Lazar, N.: The ASA's statement on *p*-values: context, process, and purpose. Am. Stat. **70**, 129–133 (2016)
21. Wit, E.C.: Big data and biostatistics: the death of the asymptotic Valhalla. Stat. Probabil. Lett. **136**, 30–33 (2018)
22. Zieffler, A., Garfield, J., Alt, S., Dupuis, D., Holleque, K., Chang, B.: What does research suggest about the teaching and learning of introductory statistics at the college level? A review of the literature. J. Stat. Educ. **16**, 2 (2017). https://doi.org/10.1080/10691898.2008.11889566

Hierarchical Word Mover Distance for Collaboration Recommender System

Chao Sun[1](✉), King Tao Jason Ng[2], Philip Henville[3], and Roman Marchant[2]

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

[2] Centre for Translational Data Science, The University of Sydney, Sydney, Australia

[3] Faculty of Engineering and Information Technologies, The University of Sydney, Sydney, Australia

Abstract. *Natural Language Processing* (NLP) techniques have enabled automated analysis over a large collection of documents, which makes it possible to quantitatively compare researcher profiles based on their publications. This paper proposes a novel researcher similarity measuring system which combines a variety of techniques, including topic modelling, Word2vec and word mover distance calculations on publication abstracts. The proposed method, implemented in python, matches researchers based upon a document's texts by evaluating the semantic meanings of words and topics. The distances between researchers are calculated over various text features in an hierarchical structure. Results show that the system is successful in identifying existing co-authorships from sample data despite co-authorship properties having been removed, as well as suggesting valid potential academic collaboration links from related research areas irrespective of previous collaboration activity.

1 Introduction

Finding a person with the requisite expertise in the job market is often challenging, because resumes contain limited information and are often simply a list of keywords for knowledge and skills that the person claims to be competent at. In the academic world, this is less of an issue, as most researchers have publicly accessible track records, including peer reviewed publications, which reflects their true expertise and research interests. However, in large and diverse academic institutions, where both the amount of information can quickly become overwhelming and researchers often have limited networks outside their own discipline, the process of seeking potential research collaborators is often via traditional off-line networking, such as conferences and seminars. This paper outlines how machine learning can provide a productive augment to help an academic or business effectively find and join or construct multi-disciplinary teams that extend beyond their known networks.

R. Islam et al. (Eds.): AusDM 2018, CCIS 996, pp. 289–302, 2019.
https://doi.org/10.1007/978-981-13-6661-1_23

Recent advances in *Natural Language Processing* (NLP) based data science techniques, coupled with on-line availability of global publication databases, allows exploration of new ways of introducing potential collaborators by matching researchers based on their track records. An efficient and accurate recommender system for researchers allows academic institutions to foster internal collaboration and external engagements that result in beneficial research developments.

Many universities have implemented network visualisations that show collaborations for each researcher [11] on their official University profile webpage. However, many of these graphs are merely illustrations of known co-authorship, and are of limited use for suggesting future collaborations. Some research-focused networking platforms (e.g. Academia.edu and ResearchGate.net) identify papers, people and projects that might be of interest to each user. However, whilst these systems focus on predicting potential collaboration, they are merely based upon the existing networks of citations, areas of research and co-authorships. The proposed method is able to predict collaboration opportunities based on the text contents of all author publications.

For the purpose of recommending similar researchers, Xu et al. [17] proposed a two-layer network based approach that combines semantic concept analysis and social networks. Another closely related work is the 'ScholarSearch' system developed by Gollapalli et al. [4], in which researcher profiles are sorted based on their personal websites, research profiles, publication catalogues and course pages. Several techniques were evaluated as a baseline for estimating similarity between researcher profiles, including: vectorising and ranking profiles with the word *Inverse Document Frequency* (IDF) using Okapi BM25 [7]; *Kullback-Leibler Divergence* (KLD) [8] after converting researcher profile into *Term Frequency* (TF); *Topic Modelling* on the topic distributions of researcher profiles and Trace-based Similarity for finding relevant documents [1]. Surprisingly, the findings of this preliminary analysis suggested that simple models based on term vectors performed better than more complex models such as the KLD and PM over topic modelling. Gollapalli et al. [4] highlights that for such recommender systems, there is an imprecise definition of "correctness", and moreover performance evaluation is not well defined. In the authors' opinion, existing work can be improved by considering more complex NLP techniques which take into account similarity between word meanings.

In this paper, we propose an hierarchical method to recommend researcher collaboration by combining several NLP algorithms, which cumulatively analyse a researcher's publications. The fundamental aim of this work is to develop a distance metric that measures the similarity between objects at different levels of the hierarchy (e.g. researchers, articles). For instance, articles are deemed to have high similarity if they discuss topics from closely related areas. This idea is applied at each hierarchy level to estimate the similarity between objects per level, i.e. topic level, article level and researcher level. The features per level are the words, topics and articles respectively. At the bottom of this hierarchy, the word level distances are determined by the pre-trained Word2vec model [10], so that the distances on upper levels can be built recursively using the same distance metric (Fig. 1).

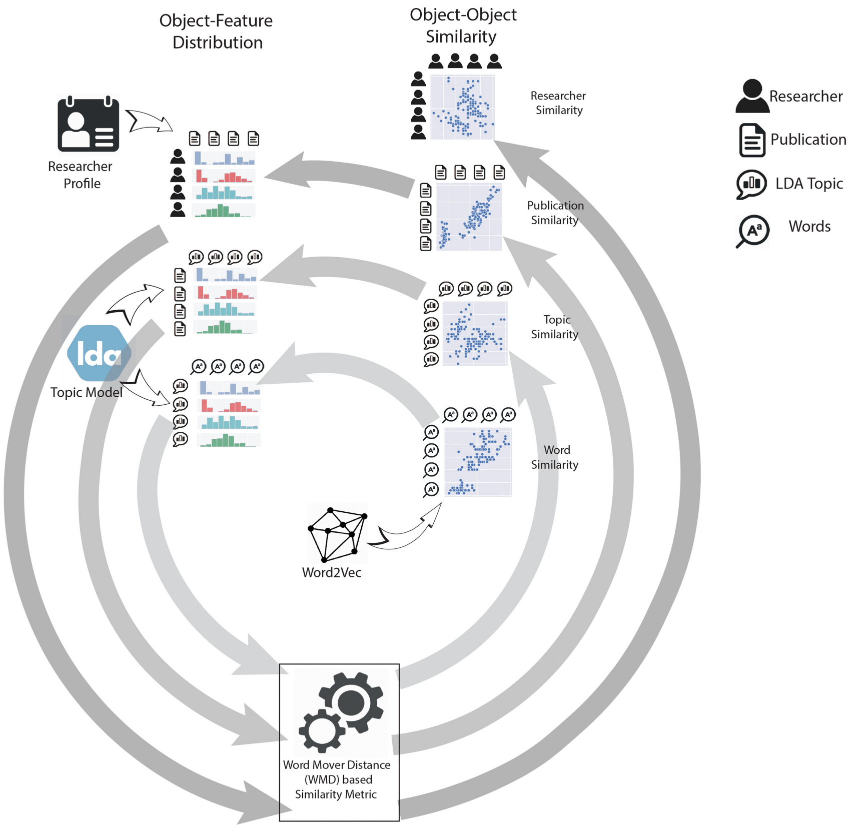


Fig. 1. An overall diagram of the proposed hierarchical similarity metric.

The rest of this paper is structured as follows. Section 2 briefly explains the theoretical background and algorithms that provide the key foundation for our proposed methodology. Section 3 describes the proposed method. Section 4 validates the proposed methodology by evaluating the performance of the method. Section 5 summarises the findings from this work and proposes future improvements and applications.

2 Theoretical Background

This section describes the required theoretical background used as a foundation for the proposed methodology.

In order to apply machine learning techniques and computation over text, documents - and the words they contain - need to be transformed into a numeric representation. For instance, one of the most common ways for creating a vectorised representation of the document is a *Bag-of-Words* (BOW). The full vocabulary is used to construct a high-dimensional space in which each document

Table 1. Term Frequency of example phrases

	at	chicago	dinosaur	exhibition	greet	illinois	in	media	museum	obama	president	press	speak	to
1.	0	0	0	0	0	1	0	1	0	1	0	0	0	1
2.	0	1	0	0	1	0	1	0	0	0	1	0	0	0
3.	1	1	1	1	0	0	1	0	1	0	0	0	0	0

Table 2. Cosine distance matrix among vectorised phrases

	Vec_1	Vec_2	Vec_3
Vec_1	0	1	1
Vec_2	1	0	0.592
Vec_3	1	0.592	0

is transformed into a *Term Frequency* (TF) vector through counting the occurrence of terms (words) within the document. Text vectorisation is the basis for many advanced text-mining techniques (e.g. topic modelling), where the terms act as the fundamental elements for text analysis. However, using individual terms as elemental units for analysis has its own shortcomings.

To exemplify the limitations of word-to-word comparison, we build on the example provided by [9] by comparing the similarity between the following three short sentences using the TF vectorisation method, and a numeric distance representation:

1. Obama speaks to the media in Illinois;
2. The President greets the press in Chicago;
3. A dinosaur is in exhibition at Chicago museum;

In this example, a reader with general contextual knowledge can easily understand that the first two sentences have a close meaning to each other even though different words were used for describing possibly the same event, whilst the third sentence has a very different meaning. However, the raw euclidean distance between the TF vector will have trouble identifying this because similarity of the meaning of the words is not taken into account. The lower case, transformed and lemmatised sentences are shown as TF vectors in Table 1, and the cosine distances between these three vectors are shown in Table 2.

According to Table 2, vectors 2 and 3 are closer than vectors 1 and 2. This shows the importance of considering the meaning of words whilst measuring the similarity between text pieces. Common distance metrics and vectorisation techniques cannot solve this problem at the document level, therefore distance estimation at the word level is a fundamental prerequisite for properly measuring similarity. The following subsection presents the solution for this problem.

Word2vec

The Word2vec technique, developed by Mikolov et al. [10], is a popular solution for representing words in a continuous vector space where semantically similar words are mapped to nearby points. Figure 2 illustrates how the Word2vec model could help improve the previously described situation.

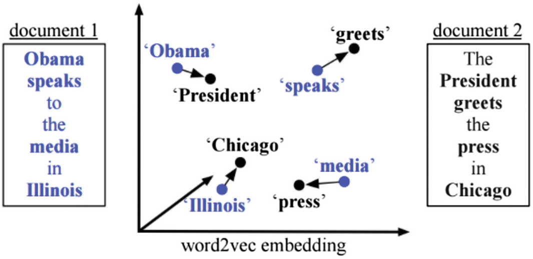


Fig. 2. Semantically similar words in a Word2vec space [9].

Topic Modelling

Topic Modelling is a method that represents a document in a lower dimension space than TF by estimating a vector in the space of topics; where topics are determined and learned from the occurrence of words in the documents. Topic Modelling is often used as a text-mining tool for finding hidden semantic structures in a corpus, however, the dimensions and components of each vector, represented as "topics", do not necessarily correlate to a meaningful interpretation that a human would expect. Existing Topic Modelling algorithms include probabilistic *Latent Semantic Analysis* (pLSA) [6], *Non-negative Matrix Factorization* (NMF) [2], *Latent Dirichlet allocation* (LDA) [3], among others, with LDA being the most widely used Topic Modelling algorithm currently. Within Topic Modelling, each document is represented as a probability distribution over all topics, and each topic is a probability distribution over the whole vocabulary of the corpus. In this paper and associated work, LDA topic modelling has been used to build an intermediate level between the document and word levels, in order to reduce the calculation burden and to increase scalability.

Word Mover Distance

The Word2vec model computes the distance between individual words, however the general distance-like functions, such as the cosine distance, Kullback-Leibler Divergence and Shannon-Jason Divergence, are not capable of accommodating additional distance information from the sub-level feature space. We therefore use the Word Mover Distance (WMD), as proposed by Kusner et al. [9], in order to solve the exact problem demonstrated here.

WMD is a simple idea based on the earth mover distance [5], wherein the distance between vectors is calculated by comparing values across dimensions whilst taking into account the similarity between the features on each dimension. Figure 3 illustrates the distance calculation between two vectors. The shape and

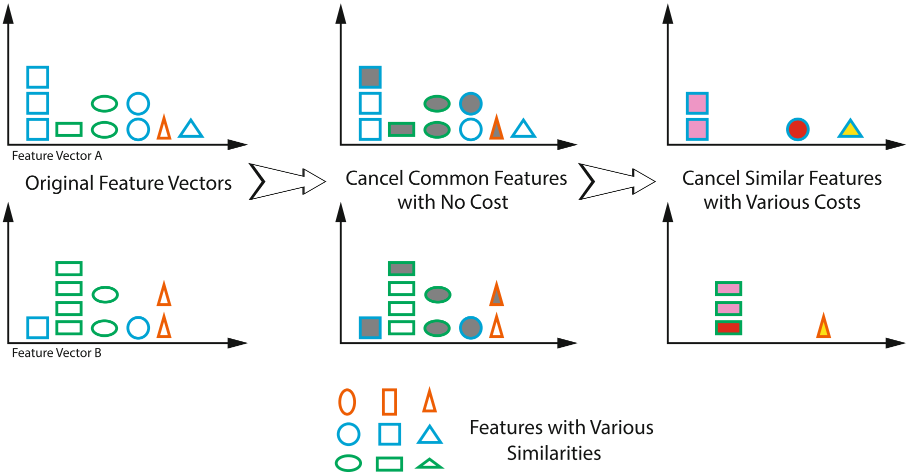

Feature Vector A
Original Feature Vectors
Feature Vector B
Cancel Common Features with No Cost
Cancel Similar Features with Various Costs
Features with Various Similarities

Fig. 3. Using WMD for calculating the similarity between two feature vectors. (Color figure online)

colour of the objects are used to represent features with various similarities, where the count of objects is the value of the vector at that dimension. In order to calculate the distance, firstly, corresponding features between the two vectors are neutralised without any cost (shaded in gray in the middle of the diagram). Then, if the similarities among features are known, closest features can be used to neutralise each other at a cost proportional to their dissimilarity. When the distance calculation involves neutralising objects of a different shape/color, the most similar features are compared against each other and the filling color on the right hand side indicates the cost of neutralising these values between vectors.

Considering the vectorised document as a distribution over the full vocabulary, where the words correspond to features, the similarity (or distance) between two documents can be measured by the WMD with respect to the distance between words provided by the Word2vec model. This similarity metric can be generalised, as long as both the feature distance matrix and feature distribution are available. In this work, the modified WMD is used as the main similarity function in the proposed hierarchical researcher similarity measuring system.

3 Methodology and Approach

As demonstrated in the previous sections, the core idea of this paper and associated work is the usage of the WMD based similarity metric over an hierarchical multi-level structure comprising the words up to and including the similarity between researchers. Under this structure, the similarity between researchers corresponds to the similarity between their publications, and the similarity between articles corresponds to their common similar topics. Finally, the similarity between topics is determined by weight on similarly semantically close words. The proposed method is implemented using Python 3.6.5 with current open source NLP toolkits, e.g. NLTK [15], Gensim [14] and pyemd [12,13]. The rest of this section clarifies the key settings on all three levels of the proposed method.

Level of Researchers
The sub-level feature for researcher comparison is the publications, however, in order to simplify the procedure at the prototyping stage of this work, only article abstracts are used for matching researchers. The researcher-abstract distribution is based on the publication authorship without considering the ordering of authors. Therefore, a researcher is either 100% or 0% associated to an abstract, regardless of his/her potential contribution to the actual contents. Other relevant research documentations/outputs, such as grant applications, are not included in this study. All researcher-abstract distributions are normalised to sum 1 so that researchers with differing number(s) of publications can be directly compared.

Level of Publications
All selected articles are written in English so that the abstracts can be easily tokenised, stemmed and modelled using prevailing functions. NLTK is used to tokenise the abstract(s), remove English stopwords and perform stemming before vectorising the document collection into a standard TF document term matrix (DTM). In the original WMD paper [9], Kusner et al. selected words as the feature of documents, however the WMD is not very efficient at processing large datasets because it has to match all word-pairs from a large vocabulary set for every two documents. Considering the scalability of this system, a level of LDA topic models is inserted between the documents and words in order reduce the overall computational complexity. Through using the topic model, all articles regardless of their actual word count are transformed into uniform distributions over a given number of topics.

Gensim 3.5.0 is used for the LDA topic modelling over all DTMs, and grid search evaluates the optimal topic number with the perplexity across topics [16]. The abstract-topic distribution is drawn from the topic model, which is prenormalised, and shows the weights on each topic as a percentage.

Level of LDA Topics
Every LDA topic is naturally a distribution over the whole vocabulary, and the word level distance matrix is initialised from the Word2vec model trained on English Wikipedia dumps[1]. Except for the topic number, all parameters are left to default values in the Gensim LDA function.

4 Experiment and Outcomes

A prototype has been developed for modelling the abstracts of 17 selected researchers, all from the Faculty of Engineering and Information Technologies, the University of Sydney. 1800 abstracts have been collected online using the

[1] Downloaded from https://github.com/idio/wiki2vec/.

Scopus API[2], and each abstract is labelled with existing co-authorship. For the evaluation purpose, existing co-authorship relations are used as the ground truth of successful research collaboration, and this information is eliminated from the data by allowing only one author per publication.

The TF document term matrix has a shape of 1,800 by 9,482, meaning there are 1800 documents with a vocabulary set comprising some 9,482 terms. The number of topics has been arbitrarily determined as 41 according to a local optimal perplexity gully shown in Fig. 4.

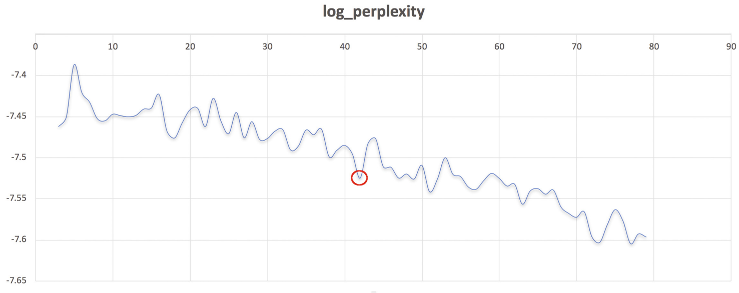


Fig. 4. Perplexities of LDA models with 3-80 topics.

All tests have been executed on a Macbook Pro 15 with 7th gen i7 CPU, 16 GB Ram using Python 3.6.5 64 bits under single thread mode. The time efficiency improvement by inserting a level of topic models is noticeable on a small dataset with only 1,800 abstracts (150 words in average abstract). A test run without the topic level took more than 17 h to finish, whilst the running time for the same process was reduced to about 50 min using 42 topics as an intermediate level between the documents and the words (including an once-off LDA training process which can be reused in the future). Accordingly, it is easy to consider that as one scales both the number of articles and length of each article, the time savings become more significant.

Impressions

Evaluating whether such an approach works for the prediction-oriented business case can be quite subjective, as the main target is to create a recommendation system for seeking new potential collaboration opportunities rather than to verify existing collaborations. A baseline "rule of thumb" for this particular work must result in the top recommendations (of similar researchers) including most of the known existing collaborators. On this basis, researchers with close distance but no previous collaboration should be highlighted for further investigation for collaboration suitability.

[2] Check https://dev.elsevier.com/sc_apis.html for more details.

Table 3. Top 20 pairs of researchers sorted by the profile similarity (ascending order in distance).

	Researcher_1	Researcher_2	Distance	Co-authorship
1	Dacheng Tao	Dong Xu	0.07751413	8
2	Simon Ringer	Julie Cairney	0.12305378	48
3	Xiangyuan Carl Cui	Simon Ringer	0.13905546	40
4	Luming Shen	Itai Einav	0.154071352	7
5	Simon Ringer	Zongwen Liu	0.156210797	32
6	Xiangyuan Carl Cui	Julie Cairney	0.163963329	1
7	Yuan Chen	Jun Huang	0.163996119	0
8	Jinman Kim	Dong Xu	0.166429949	0
9	Xiangyuan Carl Cui	Zongwen Liu	0.170656669	5
10	Jinman Kim	Dacheng Tao	0.175723115	0
11	Itai Einav	Daniel Dias Da Costa	0.178341795	0
12	Xiaozhou Liao	Simon Ringer	0.184834333	31
13	Zongwen Liu	Yuan Chen	0.185263255	0
14	Qing Li	Itai Einav	0.188056821	0
15	Zongwen Liu	Jun Huang	0.192172466	5
16	Luming Shen	Daniel Dias Da Costa	0.192986325	2
17	Julie Cairney	Zongwen Liu	0.194614062	0
18	Judy Kay	Jinman Kim	0.19528421	0
19	Simon Ringer	Luming Shen	0.198239671	2
20	Xiaozhou Liao	Zongwen Liu	0.20194694	0

The top 20 researcher pairs sorted by the WMD distance are shown in Table 3. The number of co-authored papers are listed in the last column for each pair of researchers. Within the total 136 pairwise combinations, 17 pairs of researcher have co-authored at least once (in the sample data). 11 out of these 17 collaborations are listed in the top-20 recommended researcher pairs. Also, as expected, researchers who have co-authored many times are ranked quite highly in this table.

A researcher network visualisation is also generated for visual verification, in which all researchers in Table 3 are represented as nodes connected in a network, where the distances between linked nodes are specified to be the distance produced by the proposed system.

One limitation is that the WMD based researcher distance does not satisfy the triangle inequality, therefore a 2-dimensional network is not able to correctly place all nodes corresponding to the given distance matrix. In order to obtain a layout that maximises the accuracy of displayed distances, the bottom half of the links are broken where the distances are greater than the average value. According to the table, these removed weak researcher pairs contain only 2 real co-authorships out of the total 189 past collaborations - meaning the link removal does not adversely impact the overall performance.

The network illustrated in Fig. 5 is a combination of computed and existing collaborations. Closeness implies greater collaborative potential. Grey links are predictive, green links are confirmed actual co-authorships from the sample data. The thickness and darkness of the green links represent the number of past co-authorship(s). As stated prior, we removed co-authorship from the training data prior to generating the network map, which produced the initial grey layout, over which we layered the green actual co-authorships. In this simple graph, the proposed method has correctly predicted many pairs of researchers that may and have collaborated. Further, of the researchers with much greater separation, only 2 pairs have actually co-authored (with consideration to our rule above whereby below average distances were removed during the link breaking process). Thus, the authors of this paper believe that a quite satisfying outcome is produced for the purpose of a personnel recommending system.

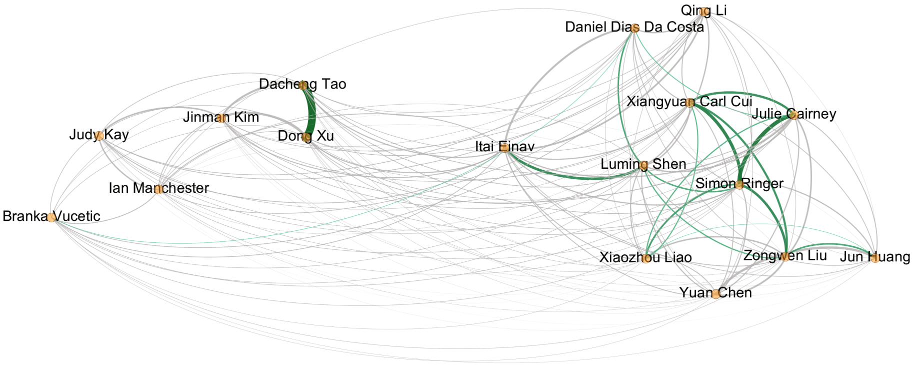
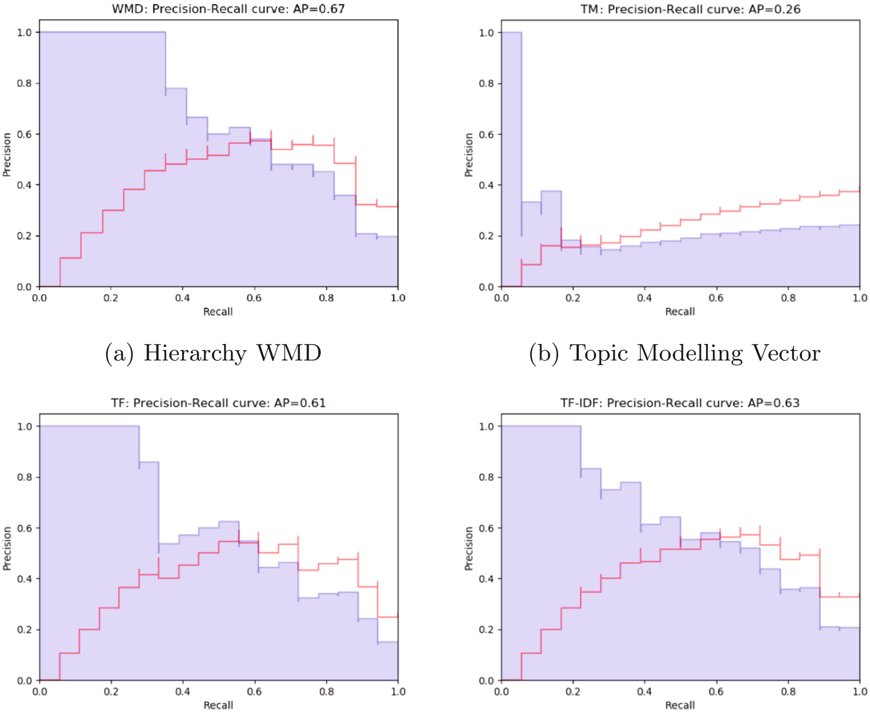


Fig. 5. Researcher similarity network with links of known co-authorships. (Color figure online)

On top of the visual evaluation, the researcher similarity outputs have been qualitatively reviewed by a data analytics officer from the Faculty of Engineering and Information Technologies, who is familiar with the research areas of the selected researchers. Notwithstanding the limited sample size, the computed distances satisfactorily both display known collaborations and usefully predict (feasible) cross-disciplinary opportunities. More specifically, the distance between the researchers, the sub-groupings and the ordering (or placement) are realistic, especially in terms of faculty organisational structure. Not surprisingly from a business perspective, the network visualisation confirms researcher recruitment activity (and capability strategy by inference).

Evaluation. Although the existing co-authorship is not a perfect indicator for measuring the suitableness among researchers, it is still useful as the ground truth for evaluating precision against a few baseline approaches. The researcher profiles are constructed by combining all publications into a single document, which are then vectorised into three types, e.g. the TF vectors, TFIDF vectors, and vectors of LDA topic signatures. Cosine distance is employed to calculate the distances between these baseline researcher vectors without the use of Word2vec, WMD and the hierarchical structure.



(a) Hierarchy WMD

(b) Topic Modelling Vector

(c) TF Vector

(d) TF-IDF Vector

Fig. 6. F1-score and Precision Recall Curve comparison. The red line stands for the F1-score and the blue shade stands for the precision change over recall. (Color figure online)

The precision-recall-curve (PRC) and F1-score are used for quantitatively evaluating the outputs of the proposed method against the three baseline methods. At each cut-off distance threshold, a decision is made so that all researcher pairs with distance less than the threshold are classified as collaborated and the rest as non-collaborative, from which the precision, recall and F1-score are then calculated using Formula 3.

$$precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{1}$$

$$recall = \frac{TruePositive}{TruePositive + TrueNegative} \tag{2}$$

$$F1\_score = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{3}$$

The precision indicates how many classed pairs are relevant to the ground truth, while the recall indicates how many relevant ground truth have been classed at a given distance threshold. The F1-score is an harmonic average of both the precision and recall. Figure 6 includes the PRC and F1-score from all four methods evaluated in this work, where the blue shade stands for the precision change over recall and the red line is the F1-score over recall.

The comparison in Fig. 6 has shown that the WMD model performs (albeit slightly) better than the baseline models on finding researchers who have existing collaborations. The baseline models also confirmed the findings by Gollapalli et al. [4], that the simple models (TF and TF-IDF) work much better than more complex models (Topic Modelling). However, it is surprising to find that the simple TF-IDF shows on par performance with our method, because the TF-IDF model contains much less information than the proposed hierarchy models.

One possible explanation to this problem is the narrow scope of data selection and the specialities of the objectives. In this trial, all texts are peer-reviewed publications and all authors are highly educated scholars from the same faculty - therefore it is expected that the language used by these well trained experts are standardised, limited and niche. One of the most important contribution of this work is introducing the word level similarities as the basis of the whole system, therefore, for academic publications from the same discipline, it can be expected that people use the same terminologies for all scholarly arguments. Because of the lack of vocabulary diversity, the benefit from the Word2vec model is minimised, particularly given that the Word2vec model was trained over general non-academic corpora. In the scenario, where a concept may be described using different terms, such as in a multi-disciplinary environment, the authors believe the proposed hierarchical method should show significant advantage in cooperation with a specially trained Word2vec model over the academic texts.

5 Conclusion

The prototyping system of the proposed researcher matching system has shown promising results through successfully producing a researcher distance matrix based on the selected researchers' publication abstracts. However, there are also a number of problems to be solved in order to use it as a real-world application, such as:

1. The full text publications should be used instead of abstracts in the scale-up system, with more researchers from diverse disciplines.
2. A specially trained "scholarly" Word2vec model should be built and used instead of a generic language model.
3. An evaluation should be conducted to verify the predictive capacity, where the ground truth is not only based upon past co-authorships.
4. Accelerate the current algorithm by utilising multi-thread computing and HPC computing resources for much larger scale modelling process.

The researcher recommendation system demonstrated in this paper is essentially an NLP based querying method (that uses text contents as the input). By way of extension, this method could be used in various business cases other than collaboration suggestion - for example, a potential industrial partner could upload a text project proposal, which searches for and recommends academics with matching expertise based upon publication history. The known issues and potential usages will be addressed in the future work.

Acknowledgements. Dr Joel Nothman from the Sydney Informatics Hub has provided valuable suggestions and feedbacks to this work.

Prof. Nick Enfield, director of SSSHARC, Faculty of Arts and Social Sciences, the University of Sydney, initiated the question and supported this work.

The major development was conducted under the Capstone student project program initiated by the School of IT, the University of Sydney.

References

1. Ahlgren, P., Grönqvist, L.: Evaluation of retrieval effectiveness with incomplete relevance data: theoretical and experimental comparison of three measures. Inf. Process. Manag. **44**(1), 212–225 (2008)
2. Arora, S., Ge, R., Moitra, A.: Learning topic models - going beyond SVD. CoRR abs/1204.1956 (2012)
3. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
4. Gollapalli, S.D., Mitra, P., Giles, C.L.: Similar researcher search in academic environments. In: Proceedings of the 12th ACM/IEEE-CS Joint Conference on Digital Libraries, JCDL 2012, pp. 167–170. ACM, New York (2012)
5. Hitchcock, F.L.: The distribution of a product from several sources to numerous localities. J. Math. Phys. **20**(1–4), 224–230 (1941)
6. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1999, pp. 50–57. ACM, New York (1999)
7. Jones, K.S., Walker, S., Robertson, S.E.: A probabilistic model of information retrieval: development and comparative experiments. In: Information Processing and Management, pp. 779–840 (2000)
8. Kullback, S., Leibler, R.A.: On information and sufficiency. Ann. Math. Statist. **22**(1), 79–86 (1951). https://doi.org/10.1214/aoms/1177729694
9. Kusner, M.J., Sun, Y., Kolkin, N.I., Weinberger, K.Q.: From word embeddings to document distances. In: Proceedings of the 32nd International Conference on Machine Learning, vol. 37 (2015)
10. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. CoRR abs/1301.3781 (2013)
11. Newman, M.E.J.: Coauthorship networks and patterns of scientific collaboration. Proc. Natl. Acad. Sci. United States Am. **101**(1), 5200–5205 (2004)
12. Pele, O., Werman, M.: A linear time histogram metric for improved SIFT matching. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 495–508. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88690-7_37

13. Pele, O., Werman, M.: Fast and robust earth mover's distances. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 460–467. IEEE, September 2009
14. Řehůřek, R., Sojka, P.: Software framework for topic modelling with large corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, Valletta, Malta, pp. 45–50, May 2010
15. Wagner, W.: Steven bird, ewan klein and edward loper: natural language processing with python, analyzing text with the natural language toolkit. Lang. Resour. Eval. **44**(4), 421–424 (2010)
16. Wallach, H.M., Murray, I., Salakhutdinov, R., Mimno, D.: Evaluation methods for topic models. In: Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, pp. 1105–1112. ACM, New York (2009)
17. Xu, Y., Guo, X., Hao, J., Ma, J., Lau, R.Y.K., Xu, W.: Combining social network and semantic concept analysis for personalized academic researcher recommendation. Decis. Support Syst. **54**(1), 564–573 (2012)

Health, Software and Smart Phone

An Alternating Least Square Based Algorithm for Predicting Patient Survivability

Qiming Hu, Jie Yang(✉), Khin Than Win(✉), and Xufeng Huang

University of Wollongong, Wollongong, NSW 2500, Australia
qh042@uowmail.edu.au, {jiey,win,xhuang}@uow.edu.au

Abstract. Breast cancer is the most common cancer to females worldwide. Using machine learning technology to predict breast-cancer patients' survivability has drawn a lot of research interest. However, it still faces many issues, such as missing-value imputation. As such, the main objective of this paper is to develop a novel imputation algorithm, inspired by the recommendation system. More precisely, features with missing values are regarded as items to be evaluated for recommendation.

Consequently, a matrix factorisation algorithm (Alternating Least Square, ALS) is employed to replace missing values; accordingly, four different prediction strategies based on the ALS result are further discussed. The proposed ALS-based imputation algorithm is evaluated by using a large patient dataset from the Surveillance, Epidemiology, and End Results (SEER) program. Experimental results demonstrates a significant improvement on the survivability prediction, compared to existing methods.

Keywords: SEER dataset · Survivability prediction · Missing-value imputation · Alternating Least Square

1 Introduction

Machine learning has been successfully applied in many medical domains, such as disease diagnosis, treatment suggestion, drug manufacturing, *etc*. Survivability of breast cancer has been of interest to clinicians, patients and the healthcare industry. Various studies on the prediction of survivability of breast cancer have been published, for example by Delen et al. [1], Liu et al. [2] and Solti et al. [3]. However, with the increasing data samples, the performance varies from different machine learning methods, and the resultant models are unstable consequently.

In addition to the unstable performance for the large amount of input records, machine learning methods also suffer from other major drawbacks such as data integrity, algorithm transparency and so on. Among those issues, the problem associated with missing value is the most challenging one. Even the most sophisticated machine learning algorithms will fail due to the information loss [4].

R. Islam et al. (Eds.): AusDM 2018, CCIS 996, pp. 305–317, 2019.
https://doi.org/10.1007/978-981-13-6661-1_24

Therefore, there is an increasing need to develop alternative frameworks for conducting missing-value imputation before applying any machine learning algorithms.

The research aim of this study then is to present an novel imputation method, particularly when a large amount of missing values are present, to assist with a robust prediction model for the breast-cancer survivability. Towards this end, an ALS (Alternating Least Square) based algorithm is proposed. Traditionally, ALS is an efficient tool for matrix factorisation, which is commonly employed in recommendation systems to suggest items for users. In this study, features with missing values are considered as items to be evaluated in a recommendation system. As such, the ALS-related recommendation technique can be employed to replace the missing values. Furthermore, four different classification strategies have been implemented to cope with the outcome from the ALS factorisation, for the prediction purpose.

A large dataset available from the Surveillance, Epidemiology, and End Results (hereafter SEER) program is used in this study, while several existing missing-value imputation techniques have also been employed for comparison. Experimental results demonstrate a huge improvement in terms of the survivability prediction using the proposed ALS-based algorithm when replacing the missing values. The outcome of this study can further be applied to other datasets, or for other purposes such as diagnostic decision making, treatment selection, *etc*.

In the remainder of this report, Sect. 2 briefly reviews relevant literature on breast cancer study and related missing-value imputation techniques. Section 3 describes the target dataset and relevant feature selection. Section 4 formulates the missing-value imputation as a matrix factorisation problem; accordingly, four classification strategies are implemented. Section 5 details experimental analysis and the comparison of the proposed methods with other state-of-the-art imputation methods. And finally, Sect. 6 presents a summary and conclusion.

2 Literature Review

One particular interest of this study is to understand breast cancer survivability using different features from the SEER dataset [5]. Breast cancer is one of the most common cancers worldwide, and it is making up to 25% of all cancer-related cause of death cases, second only to lung cancer. Thanks to the imaging and other early diagnostic screening technology, the survival rates of breast cancer have been optimistically heading toward a positive trend, with about 80% to 90% in England and the United States alive for more than 5 years after being diagnosed [6]. Studies show that cancer patient survival rates are related to multiple factors such as tumor characteristics, surgical treatment, stage, year of diagnosis and age, *etc*. [7,8]. Towards this end, different strategies of breast cancer screening and diagnosing have been evolved over years. In addition, many machine learning technologies are being developed to detect distant metastases and recurrent disease as well as to assess response to breast cancer management [1–3].

Despite the general interest of applying machine learning algorithm for the survivability prediction, there are still some open questions, among which, the problem with missing values has always been a critical issue. Although it can be ignored to some extend, large amounts and complexity of the missing values need to be handled properly to provide statistically valid inferences. As such, some strategies, inspired by statistics and/or machine learning algorithms, have been developed to address the issue of missing values, including most-frequent value replacement [9], K-modes [10] and SOM-DBSCAN [11].

Most Frequent Value. One simple strategy to handle the missing data is to use the most-frequent value for the replacement. The assumption is that the observed data is a set of samples with a multivariate normal distribution. When there are some missing data, the correlation statistic among the variables can be replaced by the most frequent value. In other words, the missing data can be estimated using the conditional distribution of the target variable. As such, most-frequent value replacement provides an effective method for missing-data estimation, and it also turns out to be a useful statistic analysis using the likelihood ratio [12].

K-modes. The K-modes algorithm utilises a simple matching-dissimilarity measure to deal with missing values. More precisely, it replaces the means of clusters with modes, and then employs a frequency-based method to update modes during the clustering process [10]. Due to these improvements, the K-modes algorithm enables the clustering of categorical/discrete data with missing values, compared to the standard K-means clustering.

SOM-DBSCAN. The self-organizing map (SOM) is a single layer feed forward network where the output compositions are arranged in a lower dimension (usually in a form of a 2-D map) [13]. Each input is fully-connected with output neurons, and a weight vector is further attached to the output neuron with the same dimensionality as the input vectors. The goal of SOM training is then to project inputs to output neurons according to a pre-defined similarly function. One advantage of the SOM mapping is its capability of handling missing values as a null vector. With the presence of missing values, SOM still carries on the mapping process using other available data while ignoring missing ones automatically. Later, other clustering method (such as DBSCAN) can be applied to extract clusters from this SOM map, and the missing values can be replaced by the most-frequent value within each individual clusters [14].

Summary of Existing Imputation Methods. Each imputation method has its own advantages. However, it is unclear that which method is preferred given a specific dataset [15]. For instance, the most-frequent value based replacement has the advantage of easy implementation and less computational cost. However, its performance heavily depends on the prior probability of the given data. On the other hand, the K-modes method might introduce some data bias during the imputation, due to the pre-defined similarity function. Last, the SOM-based imputation has some limitation on the number of selected features and clusters. Overall, it is necessary to assess and evaluate the performance of different imputation methods, and further propose a more appropriate algorithm for missing-value imputation.

3 Dataset Description

In our study, we focus on the SEER dataset that consists of patient records from several types of cancer, such as lung cancer, leukemia, stomach cancer, brain and other nervous system cancers, *etc*. For this case study, a collection of the breast cancer patient records is particularly selected. The earliest entry was recorded in 1973 when several surgery-related features were introduced. Since then, the patient samples have been accumulated continuously. By 2012, this particular dataset consisted of 85,189 records with 203 features in total. Generally speaking, these features can be divided into two categories: demographical and clinical. Demographical information includes gender, age, ethnicity, ethnicity origin, year of birth, year of diagnosis and marital status. Clinical information is disease-related, such as primary site, tumor marker, the types of treatment (radiation, surgery), behaviour codes, laterality and histology.

Recall that the main purpose is to establish a robust model for predicting a patient's survivability. In this context, the target feature is chosen as survival months as a clear indicator. Independent features that may have impact on the patient survivability were selected, including: registry ID, marital status as diagnosis, ethnicity, Hispanic origin, gender, age at diagnosis, year of birth, sequence of all reportable malignant, year of diagnosis, primamy site, laterality, surgery of primary site, reason no cancer-directed surgery, survival months flag, radiation sequence with surgery, number of primaries, first malignant primary indicator, hostology, tumor marker 1, tumor marker 2, method of radiation therapy, behavior recode, survival months, Breast Adjusted AJCC 6th T, Breast Adjusted AJCC 6th N, Breast Adjusted AJCC 6th M and Breast Adjusted AJCC 6th Stage. Detailed description for those chosen features can be found at [5].

Among selected features, five of them contain massive amounts of missing values (53% on average), including: surgery of primary site, Breast Adjusted AJCC 6th T, Breast Adjusted AJCC 6th N, Breast Adjusted AJCC 6th M and Breast Adjusted AJCC 6th Stage. One reason for this could be that the patients did not provide enough information. On the other hand, the SEER dataset has evolved over time, and some clinical features only became available in recent years. As a result, it is impractical to backtrack the values of newly-added features onto old patients. This could be the main reason causing the problem of missing values.

4 Proposed Approach

In this section, an Alternating Least Squares (ALS)-based algorithm is proposed for missing-value imputation. Toward this end, the overview of the proposed algorithm is firstly introduced, in which the missing-value imputation is formulated as a matrix factorisation problem. Second, four classification strategies are proposed individually to cope with the outcome from ALS before establishing the prediction model.

4.1 Overview of the Proposed Algorithm

In general, the process of survivability prediction is to utilize patients' historical records in order to extract subjective information, and then to categorise or classify them based on their survivability level. To express the matter in mathematical terms: let a patient record be represented as a vector with n-independent features $\mathbf{x} = \{x_1, x_2, ..., x_n\}$ and the class label be $y = \{0, 1\}$, where 0 represents the patient with survival month less than a cut-off period and 1 represents a patient that has survived longer than that. As such, survivability prediction modelling aims to train a classifier that extracts the decision rule subject to the following constraint:

$$y = f(\mathbf{x}) + e, \tag{1}$$

where $f(*)$ is an unknown decision function to be estimated by the classifier, and e is the corresponding error. However, with the presence of missing values on $\mathbf{x}$, it became difficult to directly apply machine learning algorithms due to the information loss. To this end, missing-value imputation method will be investigated first by looking for suitable values to replace the null, without compromising the prediction performance.

In this study, the proposed approach is inspired by the recommendation system, which aims to recommend items that may be of interest to users [16,17]. By analysing the user's personal information and historical behaviour, the recommendation system predicts user's future behaviour that is missing from the current observations. In this study, features with missing values are considered as items to be evaluated in a recommendation system. Accordingly, the recommendation-related technique can be employed to replace the missing values.p

4.2 ALS Algorithm for Missing Value Imputation

The Alternating Least Squares (ALS) algorithm is a popular tool adopted in the recommendation system [18]. Mathematically, the ALS model can be expressed as below:

$$X = PF, \tag{2}$$

where X is the matrix with each row represent a training input sample, P and F are the unknown data to be estimated. In this study, let m, n, and r represent the number of patient, total number of features, and number of dimensionality of latent feature space, respectively. As such, $X \in R^{m \times n}$ becomes the raw dataset with missing values, $P \in R^{m \times r}$ denotes the patient matrix after the factorisation, and $F \in R^{r \times n}$ is the feature matrix. In ALS, the following is employed for the least-squares based estimation of P and F:

$$\begin{cases} P = (XF^T)(FF^T)^{-1} \\ F = (P^T P)^{-1}(P^T X) \end{cases} \tag{3}$$

In this paper, we adopt the implementation from Zhou at el. [18] to conduct the ALS factorisation. Accordingly, the pseudocode of the ALS-based algorithm for the patient dataset is summarized in Algorithm 1. As observed, in the objective function (Eq. (4)), $f(P, F)$, $x_{i,j}$ stands for the value of the i-th row and the j-th column in $X \in R^{m \times n}$, while $\mathbf{p}_i$ and $\mathbf{f}_j$ stand for the i-th row in P and the j-th column in F, respectively. On the other hand, note that required parameters for ALS include: maximal iteration k, the weight λ, and the dimension size r. We will investigate the impact of the parameter r with regards to the imputation performance. Other parameters are fixed ahead as suggested in [18] with $\lambda = 0.065$ and $k = 50$.

```
Input  : Breast-cancer dataset $X \in R^{m \times n}$ with the missing value, weight $\lambda$,
         maximal iteration $k$, the dimension size $r$.
Output: $P$ (patient matrix) and $F$ (feature matrix);

Data pre-processing:
    feature selection; string-value replacement; normalization for re-scaling;
Initialization:
    Initialize the patient and feature matrix ($P \in R^{m \times r}$, $F \in R^{r \times n}$) randomly;
for $t = 0$ to $(k - 1)$ do
    Compute $P$ while fixing $F$ by minimizing the objective function;

        $f(P, F) = \sum_{i,j \in I} (x_{i,j} - \mathbf{p}_i^T \mathbf{f}_j)^2 + \lambda(\sum_i n_{\mathbf{p}_i} |\mathbf{p}_i|^2 + \sum_j n_{\mathbf{f}_j} |\mathbf{f}_j|^2)$    (4)

      Compute $F$ while fixing $P$ by minimizing the same objective function;
    Repeat above until the termination condition is satisfied.
end
```

Algorithm 1. ALS algorithm

4.3 Four Classification Strategies

The output (of P and F) from the ALS algorithm can be used to calculate the missing values for the original input by simply using:

$$X^* = P^k F^k, \tag{5}$$

where k is the maximal iteration defined in ALS. However, it is also worthwhile noting that the matrix P^k now is of much greater significance, as the reason is two-fold: (1) P^k has no missing values; (2) P^k is a low-rank projection of X in a less-dimension space.

As such, the matrix P^k can be directly combined with classification methods for prediction, without any missing-value imputation process. To verify this idea, four different prediction approaches are proposed by manipulating the patient matrix P^k. Detailed steps of these four approaches are listed as below.

1. Fist algorithm (hereafter XNC) aims to replace the missing values for the raw dataset, as shown in Eq. (5). In other words, this algorithm combines the output from ALS to reform the original dataset without any missing value. Next, it becomes the traditional machine-learning problem, and any standard classifiers can be applied to train on this new dataset;
2. Second algorithm (hereafter PNC) employs the patient matrix P^k from ALS as the only input, rather than X^*, for the classifiers. Again, the P^k matrix is a low-rank approximation of the matrix X^*, and the patient information is kept the same as the original data. More importantly, as there is no more missing values existing in P^*, the classifies can be employed in a more direct way;
3. Third algorithm (hereafter XCC) follows the same way as XNC to compute X^*, and then the clustering process is introduced to group records from X^*; last, the classifier is employed to each individual cluster. As verified in [14], splitting a large amount of samples into small groups helps in enhancing the performance of the classification. Consequently, in this approach, we establish the prediction model within each cluster;
4. Forth algorithm (hereafter PCC), similar to XCC, employs the classification process on all split clusters. The major difference, however, is to group samples from P^k instead of X^*.

Overall, Table 1 summarises the differences among the four proposed classification strategies. As observed, in XNC and PNC, the matrix X^* and P^k are used directly as an input of the classification process; by contrast, in XCC and PCC, the clustering process is introduced to group patients before the prediction process.

Table 1. Four different prediction algorithms utilizing the ALS output.

Key steps	XNC	PNC	XCC	PCC
Computing the matrix X^*	✓	×	✓	×
Clustering	×	×	✓	✓
Classification on clusters	×	×	✓	✓
Classification on P^k or X^*	✓	✓	×	×

5 Experiment Results and Analysis

In this section, the efficiency of the proposed algorithm (ALS and related four prediction strategies) is investigated based on the SEER dataset. First, the experiment design and the evaluation criterion are presented. Then experiments are conducted for the following purposes:

1. to understand the impact of different parameters (such as matrix ranks and number of clusters) on the performance of the proposed algorithms.
2. to compare the proposed algorithms with existing work for prediction modelling.

5.1 Experiment Design

The analysis goal of this study is to predict the patient survivability (in terms of months of survival time). Based on previous studies on the SEER dataset [14], the 60 months is a median survival time that has been established as a cut-off. Accordingly, the entire dataset is split into two groups: if a patient passed away within 60 months, then she will be assigned to one group and label as 0. By contrast, if the patient survived 60 months after diagnosis, this instance will be assigned to another group and label as 1. Furthermore, the entire dataset is also randomly partitioned into two independent sets: a training set, and a testing set. The training set is used as the input for different missing-value imputation methods and classification, and the testing set is used for evaluating the generalization ability. The size of the training and testing set is set as 60%, and 40%, respectively.

Meanwhile, feed-forward neural network (FNN) has drawn great interest in many applications due to its universal approximation capability [19]. Consequently, in this paper, a feed-forward neural network was employed as the classifier, with an input layer, a hidden layer, and an output layer. The activation function of the hidden layer was set to tangent sigmoid function, and the output neuron used the linear function as activation function. The bias vector was implemented as an incoming connection from the particular bias node with the input value of 1. The network was initialized with random weights in the range $[-0.1, 0.1]$, and then trained with the RPROP algorithm. The training parameters are: maximum number of training iterations 500, minimum performance gradient 10^{-6}, learning rate 0.1. The network training terminates when either the maximum number of iterations is reached, or the performance gradient falls below minimum performance gradient.

An agglomerative method was employed in the clustering process, using the average distance between any two sets as the linkage criteria [20]. The agglomerative-based approach is a typical hierarchical clustering technique, that initializes clusters with pairwise comparisons and scores. Those scores are then used to align hierarchical sequences within groups until a certain termination criterion is met. The agglomerative algorithm is implemented in this paper since it is more suitable for large samples that need to be divided into small sub-clusters.

In addition, some pre-processing work is also conducted on selected features before computing ALS. For example, the feature "SITEO2V" with string values is replaced with unique real values. In addition, normalization is applied to some features (such as ORIGIN, SEQ_NUM, NO_SURG, RADIATN and RAD_SURG) to scale the data.

5.2 Performance Evaluation

In this section, the impact of input parameters is assessed, including the matrix rank r and cluster size $n_{cluster}$. Recall that r is the number of columns from the patient matrix P, and $n_{cluster}$ is the number of clusters. They are key parameters as a bigger value r might lead to an intensive computation for ALS, while smaller r might suffer from the insufficient information. Similarly, more computational cost is required for bigger $n_{cluster}$, where samples will be mismatched with smaller value of $n_{cluster}$.

Toward this end, experiments were conducted by considering different combinations for r ($r \in [10, 20, 30]$ and $n_{cluster}$ ($n_{cluster} \in [3, 5, 7, 9]$) to get an overall view. The experiments were repeated 20 times, and the comparison results are shown in Table 2.

Table 2. Average results from four prediction strategies on the test set, taking different combinations of r and $n_{cluster}$ into account.

Prediction	$r = 10$	$r = 20$	$r = 30$
XNC	57.98%	70.00%	70.01%
PNC	57.50%	70.00%	70.18%
XCC($n_{cluster}$=3)	67.63%	69.16%	69.75%
XCC($n_{cluster}$=5)	68.69%	69.98%	70.58%
XCC($n_{cluster}$=7)	69.15%	71.96%	71.25%
XCC($n_{cluster}$=9)	69.82%	71.67%	71.29%
PCC($n_{cluster}$=3)	69.35%	70.38%	71.79%
PCC($n_{cluster}$=5)	72.36%	73.66%	72.51%
PCC($n_{cluster}$=7)	70.87%	74.82%	74.18%
PCC($n_{cluster}$=9)	71.88%	74.04%	74.68%

From the average prediction results, a few observations can be made. In general, the performance of all algorithms increased when r increases from 10 to 30. With a lower dimension, the information loss during the dimension alternation results in lower performance results. By contrast, the classification accuracy is improved with larger r. On the other hand, the performance reached its peak around $[20, 30]$, which indicates a suitable value for r is found. Similarly, the increasing value of $n_{cluster}$ likely contributes to a better performance. The reason could be a smaller number of clusters resulting in the mismatching groups, thereby making it difficult for the classifier to identify related patterns.

Furthermore, XNC and PNC approaches demonstrate similar outcome as the average results achieve approximately around 65%. That shows the information loss is negligible after ALS, and the patient matrix P keeps the data characteristic as the original dataset. Consequently, P can be used to replace the original X matrix, given that a similar performance is observed. In addition, both XCC and PCC approaches show higher prediction result (above 70% accuracy on average) compared to XNC and PNC, which indicates clustering before classification indeed improves the prediction performance. This finding is consistent with the existing work of Shukla et al. [14].

We also note that the best classification result was observed using PCC. Recall that the main difference between PCC and XCC is the patient matrix P is used for clustering and classification (rather than X^* with replaced values). Again, recall that P is a low-rank projection of the original data in a less-dimension space; and we interpret this as the main reason for the improved result. As such, PCC is adopted in the following experiment.

5.3 Comparison with State-of-Art Methods

In this section, three existing approaches are selected to compare with the proposed algorithm, including: most-frequent value replacement, SOM-DBSCAN and K-modes. Same data inclusion and feature selection processes were implemented to make a fair comparison. Table 3 summarises input parameters for three selected algorithms.

Table 3. Summary of input parameters for selected imputation methods

Approach	Most frequent	SOM-DBSCAN	K-modes
Initial rate	N/A	0.32	0.45
Iteration	N/A	10,000	25
$n_{cluster}$	N/A	$[3, 5, 7, 9]$	$[3, 5, 7, 9]$

The comparison result is shown in Fig. 1. As observed, the proposed PCC approach outperforms other methods. In all cases, PCC achieved the best prediction performance. By contrast, most-frequent based approach reached approximately 70% accuracy, that was caused by the misleading information introduced by the most-frequent imputation. As for the K-modes based method, clusters are established while ignoring the impact of missing values. Therefore, there is also a lack of some data information, that affects the sequential prediction. By contrast, the proposed algorithm manages to keep the data characteristics, and reduce the data dimension. Overall, it can be empirically confirmed with the performance of the ALS-based algorithm.

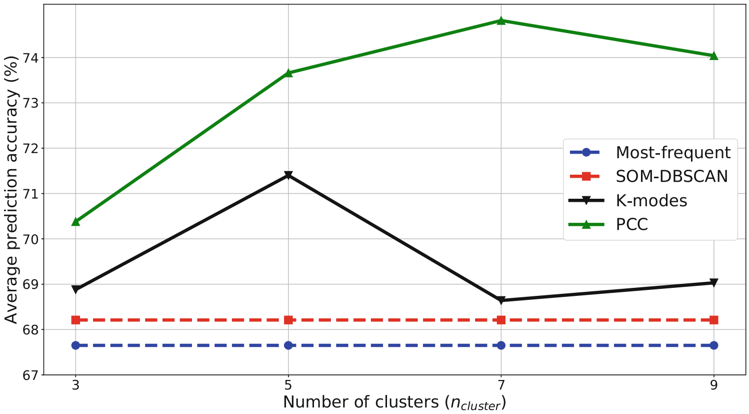


Fig. 1. The comparison in terms of prediction accuracy (%) among different imputation algorithms.

6 Conclusion

Breast cancer survivability has been comprehensively studied in the past in order to generate better prediction models that may be of help in the real-life cancer treatment. As such, a variety of many machine learning models have been employed. However, when a large amount of missing data presents, machine learning methods became ineffective and unstable due to the large information loss.

The Alternating Least Squares (ALS) algorithm has been implemented as a popular tool for recommending new items to users. In this study, missing-value imputation was formulated as a recommendation problem, in which features with missing values are regarded as items to be recommended. As such, the ALS algorithm was employed to replace missing values, and four different classification strategies were introduced accordingly.

The performance of the proposed algorithm was verified with a neural network classifier. Besides, three existing imputation techniques (most frequent value, SOM-DBSCAN, k-modes) were also applied for comparison. Experimental results confirm that the ALS-based algorithm is capable of achieving accurate and robust survivability models with the presence of missing values.

The prediction model investigated in this study is basically based on a binary classification problem. It would be of great interest to extend into a multiple classification case. In addition, the performance enhancement of ALS requires a deeper investigation about input parameters, such as optimizing the weight λ with some statistical techniques (*i.e.* cross-validation). Lastly, we also plan to utilize the feature matrix in the prediction process.

References

1. Delen, D., Walker, G., Kadam, A.: Predicting breast cancer survivability: a comparison of three data mining methods. Artif. Intell. Med. **34**, 113–127 (2005)
2. Liu, Y.Q., Wang, C., Zhang, L.: Neural network based models for predicting breast cancer survivability. Chin. J. Biomed. Eng. **28**, 221–225 (2009)
3. Solti, D., Zhai, H.: Predicting breast cancer patient survival using machine learning. In: ACM Conference on Bioinformatics, Computational Biology and Biomedical Informatics, BCB 2013, pp. 704–705. ACM (2013)
4. Lang, K.M., Little, T.D.: Principled missing data treatments. Prev. Sci. **19**, 284–294 (2018). https://doi.org/10.1007/s11121-016-0644-5
5. Surveillance, Epidemiology, and End Results. http://www.seer.cancer.gov
6. McGale, P., et al.: Effect of radiotherapy after mastectomy and axillary surgery on 10-year recurrence and 20-year breast cancer mortality: meta-analysis of individual patient data for 8135 women in 22 randomised trials. Lancet (London) **383**, 2127–2135 (2014). https://doi.org/10.1016/S0140-6736(14)60488-8
7. Jia, Y., Sun, C., Liu, Z., Wang, W., Zhou, X.: Primary breast diffuse large B-cell lymphoma: a population-based study from 1975 to 2014. Oncotarget **9**, 3956–3967 (2018)
8. Agarwal, S., Pappas, L., Agarwal, J.: Association between unilateral or bilateral mastectomy and breast cancer death in patients with unilateral ductal carcinoma. Cancer Manag. Res. **9**, 649–656 (2017)
9. Webb-Robertson, B.J.M., et al.: Review, evaluation, and discussion of the challenges of missing value imputation for mass spectrometry-based label-free global proteomics. J. Proteome Res. **14**, 1993–2001 (2015). https://doi.org/10.1021/pr501138h
10. Jiang, F., Liu, G., Du, J., Sui, Y.: Initialization of K-modes clustering using outlier detection techniques. Inf. Sci. **332**, 167–183 (2016). https://doi.org/10.1016/j.ins.2015.11.005
11. Brock, G.N., Shaffer, J.R., Blakesley, R.E., Lotz, M.J., Tseng, G.C.: Which missing value imputation method to use in expression profiles: a comparative study and two selection schemes. BMC Bioinf. **9**, 1–12 (2008). https://doi.org/10.1186/1471-2105-9-12
12. Nguyen, L.T., Schmidt, H.A., von Haeseler, A., Minh, B.Q.: IQ-TREE: a fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. Mol. Biol. Evol. **32**, 268–274 (2015). https://doi.org/10.1093/molbev/msu300
13. Abaei, G., Selamat, A., Fujita, H.: An empirical study based on semi-supervised hybrid self-organizing map for software fault prediction. Knowl.-Based Syst. **74**, 28–39 (2015). https://doi.org/10.1016/j.knosys.2014.10.017
14. Shukla, N., Hagenbuchner, M., Win, K.T., Yang, J.: Breast cancer data analysis for survivability studies and prediction. Comput. Methods Programs Biomed. **155**, 199–208 (2018). https://doi.org/10.1016/j.cmpb.2017.12.011
15. Yamaguchi, Y., Misumi, T., Maruo, K.: A comparison of multiple imputation methods for incomplete longitudinal binary data. J. Biopharm. Stat. **28**, 645–667 (2018). https://doi.org/10.1080/10543406.2017.1372772
16. Bian, Y., Li, H.: Recommendation system based on trusted relation transmission. In: 12th International Conference Intelligent Systems and Knowledge Engineering (ISKE), pp. 1–8. IEEE, November 2017. https://doi.org/10.1109/ISKE.2017.8258843

17. Nguyen, J., Zhu, M.: Content boosted matrix factorization techniques for recommender systems. Stat. Anal. Data Min.: ASA Data Sci. J. **6**, 286–301 (2013). https://doi.org/10.1002/sam.11184
18. Zhou, Y., Wilkinson, D., Schreiber, R., Pan, R.: Large-scale parallel collaborative filtering for the netflix prize. In: Fleischer, R., Xu, J. (eds.) AAIM 2008. LNCS, vol. 5034, pp. 337–348. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-68880-8_32
19. Yang, J., Ma, J.: A structure optimization framework for feed-forward neural networks using sparse representation. Knowl.-Based Syst. **109**, 61–70 (2016)
20. Rokach, L., Maimon, O.: Clustering methods. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, pp. 321–352. Springer, Boston (2005). https://doi.org/10.1007/0-387-25465-X_15

Interpreting Intermittent Bugs in Mozilla Applications Using Change Angle

David Tse Jung Huang(✉), Yun Sing Koh, and Gillian Dobbie

Department of Computer Science, University of Auckland, Auckland, New Zealand
{dtjh,ykoh,gill}@cs.auckland.ac.nz

Abstract. Learning in evolving environments involves learning from data where the statistical characteristics can change over time. Current change detection algorithms that are used online for data streams detect whether a change has occurred in the data but there is always a detection delay. None of the existing online techniques can accurately pin-point the exact location of when the change starts to occur, which can be critical. We present a novel method Change Angle and we show, for the first time, how to pin-point online the location at which change starts to occur. We apply our Change Angle method in the application area of software revision control using Mozilla data, where it is important to detect not only the presence of change but also to pin-point accurately the location of when change starts to occur.

Keywords: Data streams · Change detection · Software repository

1 Introduction

Concept drift detection or change detection, is concerned with learning from data where the statistical characteristics can change over time. Change detectors used in data streams are used to identify online, as early as possible, whether a change has occurred. Change detectors generally operate by comparing two sets of data: reference data and current data. A change is signaled when there is a statistical difference between the two sets of data. In practice, for the detectors to signal change, a sufficient number of new data with the new characteristics needs to be observed. This requirement causes a delay between when a change occurs and when a change is detected by a change detector. For example, a change can start at 1 o'clock but the detector might detect it at 2 o'clock, yielding a detection delay of an hour. In real world applications, the detection delay is generally unknown and is not easily approximated, thus making it difficult to use the detection delay as a strategy to estimate the exact location at which change starts to occur. Consider a simple scenario where a machine's error rate is tracked using change detection to monitor the performance of the machine and whether it deteriorates over time. If the error rate starts to increase and it increases sufficiently, the change detector will signal a change. The point at which a change is signaled is known as the detected change point. There will also be a point earlier than the detected change point where the error rate just

R. Islam et al. (Eds.): AusDM 2018, CCIS 996, pp. 318–330, 2019.
https://doi.org/10.1007/978-981-13-6661-1_25

begins to increase due to an underlying cause. This point is called the true change point. The true change point is the actual starting point of a change in the error rates of the machine. The time difference between the detected change point and the true change point is the detection delay, which is inevitable in all change detectors. In Fig. 1 we present an example depicting the true change point, detected change point, and detection delay.

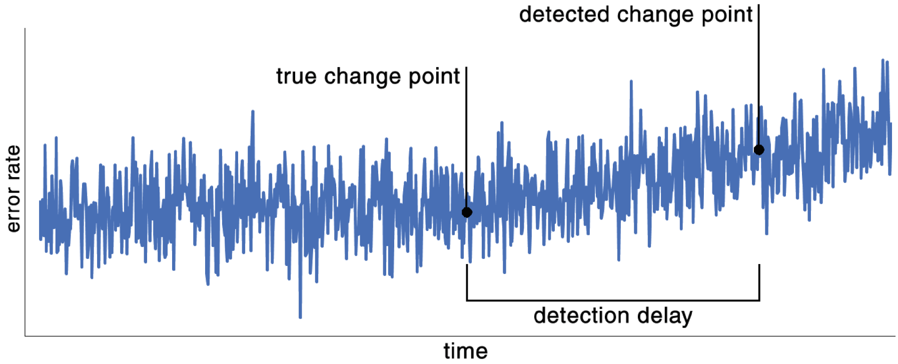


Fig. 1. Example of change detection

In application areas where it is beneficial to explain the change, such as software revision control and more specifically in the Mozilla software revision control, we must first pin-point the location of when the change started. Currently, none of the online change detectors can accurately identify the true change point. The reasons the true change point is difficult to detect in a data stream are (1) data streams require online processing and real-time responses, (2) methods that process data streams cannot assume a particular data distribution, (3) pin-pointing the start of a change is difficult due to variance in the data. As a result, in practice, often the detected change point is used in place of the true change point. While there are offline methods such as using gradient descent and hill climbing to more accurately approximate the true change point, these methods are computationally expensive when used for online processing of data streams. Some methods also assume particular data distributions, which can lead to inaccurate results. Currently none of the techniques can accurately identify the true change point online. Some change detectors like DDM [5] have a warning level mechanism which uses a lower confidence threshold to identify an earlier point closer to the true change point than the actual detected change point. However, it is still very inaccurate.

The contribution of this paper is a novel method Change Angle that can be used to pin-point true change points in a data stream. We apply our method in Mozilla software revision control and demonstrate its accuracy and effectiveness in helping Mozilla engineers better identify software bugs and strategies for resolving the bugs.

The structure of this paper is as follows. In Sect. 2, we present previous relevant research. We elaborate on the details of our algorithms and implementations in Sect. 3. Section 4 presents evaluations of our work and Sect. 5 concludes the paper and describes future directions of this research.

2 Related Work

We discuss the related work in two different sections. The first section focuses on the change detection problem in data streams and in the second section we discuss the offline search methods to determine the true change point.

2.1 Change Detection Algorithms

Several change detection algorithms for data streams have been proposed and they can be divided roughly into the following categories: Sequential analysis techniques [11], statistical process control [1,5,13], and monitoring two different time-windows [2,12]. These methods often utilize different forms of statistical significance tests or upper bounds [7,10] to signal change. Change detectors are often used together with classifiers such as [4] in the process of Adaptive Learning [3]. Gama et al. [6] present an in-depth survey on change detection algorithms.

The SEED algorithm [9] that is predominantly used in this paper falls under the monitoring two different time-windows category. Techniques in this category detect change based on comparing two monitored time-windows. A reference window which represents past data from the stream and a recent current window which represents current data from the stream are kept. A change in the data is signaled when the difference between the data in the two windows is significantly different. Another notable technique under this category is ADWIN [2].

Change detection algorithms vary across five important performance measures: true positive rate, false positive rate, detection delay, run time, and memory. While some algorithms perform better in run-time and others better in memory or true positive rate and false positive rate, each algorithm has an inevitable detection delay in their detected change point. Currently, none of the existing research in data streams looks at more accurately pin-pointing the location of the true change point online.

2.2 Search Algorithms

Search algorithms are methods that perform searching in a given problem space. Many search algorithms are used for various optimization problems where the goal is to find the best solution from all feasible solutions. The problem space for the search algorithm then becomes the set of all feasible solutions and the criteria is to discover the best solution through searching the problem space. Hill climbing search [14] can be used to address our problem.

Hill Climbing Search. Hill climbing is a metaheuristic iterative local search algorithm that starts with an arbitrary solution to a problem, and then iteratively finds a better solution by finding nearby better solutions until it reaches a state where no better nearby solutions can be found. The iterative nature of finding a better nearby solution forms a hill climbing action which is the basis of the algorithm. Hill climbing search has several variants in addition to the standard simple hill climbing search: Stochastic hill climbing, first-choice hill climbing, and random restart hill climbing. Simple hill climbing is prone to fall into local maxima and thus hill climbing with random restarts is generally accepted as a better solution to avoiding local maxima. Hill climbing with random restart takes hill climbing search but repeats the search with random initial starting points, then reports the overall best solution to the problem. While hill climbing with random restarts theoretically provides a better solution than other hill climbing variants, it also consumes more resources and is less efficient. In our experimental evaluations, we compare our online method against hill climbing with random restarts.

3 Resolving Intermittent Bugs Using Change Angle

We propose a novel method Change Angle that can be used online to pin-point the true change point. Change Angle is an online method that can be used alongside the change detection algorithm SEED [9] to pin-point true change points. The SEED algorithm accumulates data instances x into blocks B of size b and stores all the blocks in a sliding window W. SEED detects change within the window W by first dividing W into two sub-windows W_L and W_R. Then, performs hypothesis testings by comparing the sample means of the two sub-windows μ_{W_L} and μ_{W_R}. Whenever the two sub-windows of W exhibit sufficiently different means, SEED confirms the alternate hypothesis H_1 that the two sub-windows have different distributions: $\mu_{W_L} \neq \mu_{W_R}$. SEED will signal a change has been detected and drop the older portion of the window, W_L. With the arrival of each block B, SEED performs the hypothesis test at the boundaries between $W_L.W_R$ of W. The change angle can be calculated as follows. The non-reflex angle of a change in a data stream is a function of two 2-dimensional vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ given three points i, j, and k. The 2-dimensional vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ can be defined as follows where $i_x < j_x < k_x$: $\boldsymbol{u} = (j_x - i_x, j_y - i_y) \qquad \boldsymbol{v} = (j_x - k_x, j_y - k_y)$. The magnitude of the two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ are $\|\boldsymbol{u}\| = \sqrt{u_x^2 + u_y^2} \qquad \|\boldsymbol{v}\| = \sqrt{v_x^2 + v_y^2}$. We denote the cosine angle between the two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ as $(\cos\theta)$. The angle θ in radians is then given by: $\theta = \arccos\left(\frac{\boldsymbol{u}\cdot\boldsymbol{v}}{\|\boldsymbol{u}\|\cdot\|\boldsymbol{v}\|}\right)$ and in degrees $\theta^\circ = \theta \cdot \frac{180}{\pi}$. To calculate the change angle in order to pin-point the true change point, the selection of points i and j is important. We select point k as the detected change point. Our method performs a search using different values for point j to find the best value that matches a true change point. To illustrate how the points and calculation of change angle fit together, we use a

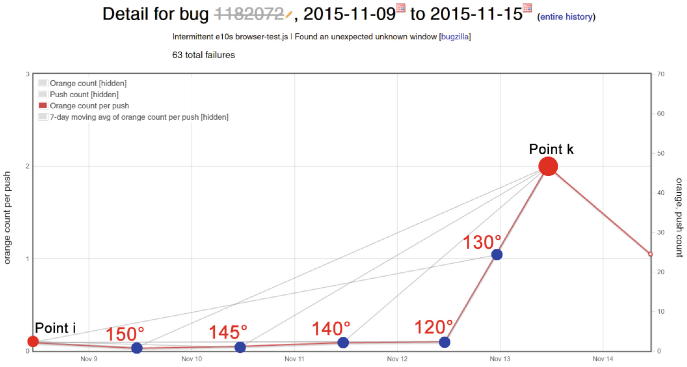


Fig. 2. Change angle calculation demonstration

real example of an intermittent bug pulled from the Mozilla intermittent bug tracking system shown in Fig. 2. Here, point i is the first block of data, which is at the beginning of the stream. Point k is the last block of data, which is at the end of the stream where change is detected. Points i and k are anchored and do not change as we calculate the various angles formed by this particular change. For each of the point j we calculate a change angle.

3.1 Mozilla's Intermittent Bug Issue

In Mozilla's software revision control and code integration process, errors or software bugs arise. Engineers create code revisions, called *changesets*, and integrate them into a central repository by *pushing* their changeset. Resolving these errors requires two steps: (1) identify a bug that has been introduced into the software, and (2) find the changeset that introduced the bug into the software. For every changeset that is pushed into the central repository, a large set of software tests is run. The result of these tests (pass or fail) is one of the primary ways of identifying possible bugs. A team of engineers called the Sheriffs manually go through these test results and identify bugs associated with the test failures. We particularly focus on intermittent bugs where the tests are not consistently reproducible. In other words, given the same test conditions, they would sometimes pass and sometimes fail. Due to the data volume, it is impractical to assign enough engineers to look into all existing intermittent bugs and as a result the Sheriffs need an automatic approach that prioritizes intermittent bugs that require immediate attention. They also need to know when the bug started to occur in order to find the changeset that introduced the bug. For this application, we use change detection to identify the presence of severe intermittent bugs that are causing instability in the software and we use Change Angle to identify the starting point of when bugs are introduced.

Identifying and Prioritizing Key Intermittent Bugs. We use change detector SEED to build an automatic tool for the Sheriffs to identify and prioritize intermittent bugs. Each intermittent bug is linked to a software test. We monitor the failure rates of these tests associated with the intermittent bugs. When a bug is introduced, the failure rate of these tests will often sharply spike and can be picked up by change detectors. The failure rate is derived by pulling data from two Mozilla data sources: the *repo* and *orange factor*. The set of sequential failure rate data is then passed through change detection. The system runs one instance of the change detector on each occurring intermittent bug in the time frame we set for our inputs. The system then simultaneously produces a list of bugs that have an associated increase in frequency of occurrence change alerted by their instance of change detector along with the time that the change signal was alerted.

Finding the Starting Point of the Intermittent Bug. Our objective is to point the engineer to the starting point of the bug so that the changeset that introduced the bug can be identified and corrected. Using Change Angle, however, we can better pin-point the starting point of the bug and help the engineer resolve the intermittent bug.

We added this additional information onto the prioritized list of bugs produced in the first part. The overall result is a list of prioritized intermittent bugs that contains bugs that have significantly increasing failure rates and also the estimated time when the bug started to occur.

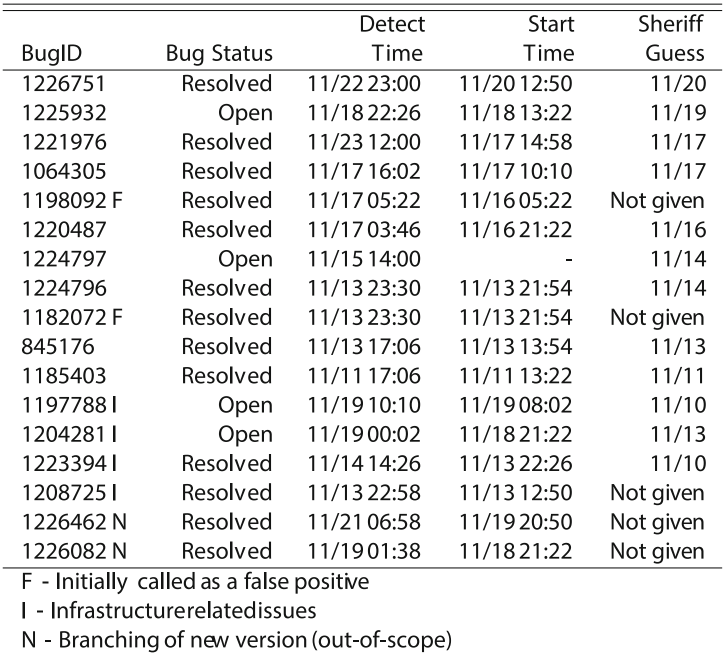
4 Evaluations

As part of our evaluations we performed a real-world evaluation on Mozilla data in Sect. 4.1. The goal is to demonstrate that Change Angle works as expected in the Mozilla application. We then further tested our algorithm on a series of synthetic test environments in Sect. 4.2. The purpose is to test the performance and behavior of our algorithm with varying data environments.

4.1 Real-World Evaluation on Mozilla Application

We ran the algorithms on the Mozilla data and produced 17 alerts pointing to 17 intermittent bugs that the algorithm believed were undergoing a change in behavior and required some possible attention. We then asked the Sheriffs at Mozilla to evaluate and validate the list.

Figure 3 shows that out of the 17 bugs that were alerted, two were false alarms (code F) and two others were out-of-scope (code N). Most of the remaining 13 bugs have been resolved or stopped occurring due to disabled tests. For those that the Sheriff provided a guess of the starting location, our prediction using the Change Angle is generally accurate. It is important to note that infrastructure related bugs (code I) are generally long running and they keep coming back. They are usually left open until an action needs to be taken, such as increasing the max heap size. In the case of infrastructure related bugs, the Sheriffs are not

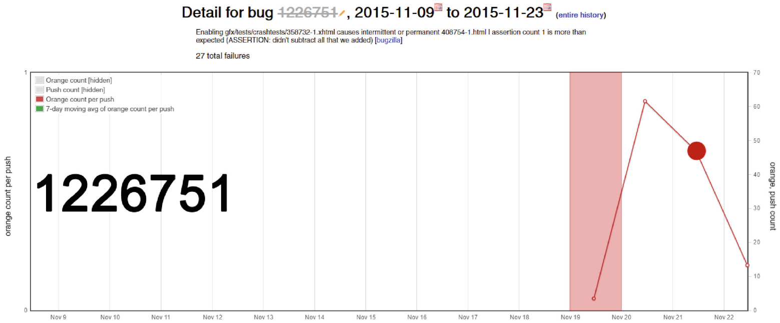


		Detect	Start	Sheriff
BugID	Bug Status	Time	Time	Guess
1226751	Resolved	11/22 23:00	11/20 12:50	11/20
1225932	Open	11/18 22:26	11/18 13:22	11/19
1221976	Resolved	11/23 12:00	11/17 14:58	11/17
1064305	Resolved	11/17 16:02	11/17 10:10	11/17
1198092 F	Resolved	11/17 05:22	11/16 05:22	Not given
1220487	Resolved	11/17 03:46	11/16 21:22	11/16
1224797	Open	11/15 14:00	-	11/14
1224796	Resolved	11/13 23:30	11/13 21:54	11/14
1182072 F	Resolved	11/13 23:30	11/13 21:54	Not given
845176	Resolved	11/13 17:06	11/13 13:54	11/13
1185403	Resolved	11/11 17:06	11/11 13:22	11/11
1197788 I	Open	11/19 10:10	11/19 08:02	11/10
1204281 I	Open	11/19 00:02	11/18 21:22	11/13
1223394 I	Resolved	11/14 14:26	11/13 22:26	11/10
1208725 I	Resolved	11/13 22:58	11/13 12:50	Not given
1226462 N	Resolved	11/21 06:58	11/19 20:50	Not given
1226082 N	Resolved	11/19 01:38	11/18 21:22	Not given

F - Initially called as a false positive
I - Infrastructurerelatedissues
N - Branching of new version (out-of-scope)

Fig. 3. Summary list of bugs (Nov 2015)

particularly concerned about the accuracy of the starting point because they are not one-off occurrences. It can also be noted that the starting location prediction of these infrastructure related bugs are less accurate compared to the other ones that are the main focus of the list.



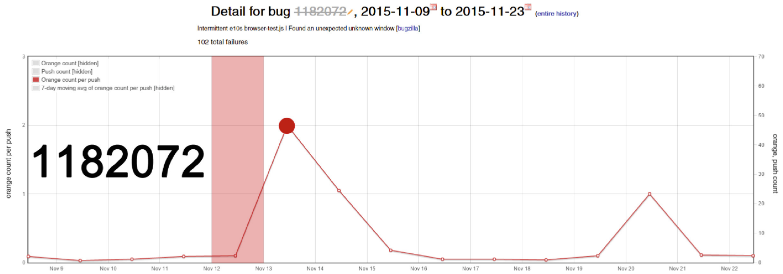
Detail for bug ~~1226751~~, 2015-11-09 to 2015-11-23 (entire history)

Enabling gfx/tests/crashtests/358732-1.xhtml causes intermittent or permanent 408754-1.html | assertion count 1 is more than expected (ASSERTION: didn't subtract all that we added) [bugzilla]

27 total failures

Fig. 4. Bug 1226751

In Fig. 4 we first present one of the cases: Bug 1226751. We show the time of detection with a large dot and shade the time frame that the sheriff believes is the first point to look at. This bug was alerted on the 22nd of November 2015, just after the peak of the spike. The predicted starting time of the bug is on the 20th of November. The Sheriff also validated that the starting point to be the 20th of November. This bug was later fixed and is now a resolved bug.



Detail for bug ~~1182072~~, 2015-11-09 to 2015-11-23 (entire history)

Intermittent e10s browser-test.js | Found an unexpected unknown window [bugzilla]

102 total failures

Fig. 5. Bug 1182072

In Fig. 5 we show Bug 1182072, an intermittent browser test failure. This bug was identified as one of the two false alarms in our validated list. This bug was alerted on the 14th of November with the Sheriff comment: “Long running bug, I don’t think Nov 14th has anything significant”. We brought this case up here because interestingly, the bug increased in frequency in the month of December 2015 and action was quickly taken late December and the error was quickly resolved. This counter example gives confidence that our system does pick up increases in the frequency of the bugs, sometimes even the Sheriffs cannot point out immediately that it is a bug worth investing.

As an overall summary, using our system that produces a prioritized list, a majority of the bugs that are on the list eventually do become issues that require engineers to look into and are not random errors. It is also important to note that the detection times are much faster compared to a human manually recognizing that there is an issue, most of the time. There is on average about a 1–2 day detection delay before an actual alert is signalled on the list but the prediction of the starting point is accurate at telling when the bug began spiking.

4.2 Theoretical Evaluation on Synthetic Data Environments

In this set of experiments we use the standard synthetic testing condition used in other change detection papers [2,8,9,12] to evaluate how well Change Angle locates the true change point. We reference the results of our Change Angle performance against the values of detected change points (in the form of detection delay) from SEED and the best offline optimized Hill Climbing performance values for true change points. The testing condition consists of data streams each with one change at $\mathcal{L} - 1000$ with varying magnitudes of change. Change is induced with different slope values over a period of 1,000 steps, where $\mathcal{L}$ is the total length of the stream. We used $\mathcal{L} = 100,000$ and ran all experiments over 100 iterations. Change Angle has a runtime performance in $< 2ms$ for all test cases. For all of the experiments, true positive rate was 100% and for each set of parameter values. We ran Hill Climbing on 30 random restarts with start point from the detected change point using SEED. We ran Change Angle using SEED as the base change detector with $\delta = 0.05$.

We report two values for comparison: the mean and standard deviation of the predictions and the proportion of predictions within 100 steps margin of error from the true change point (we abbreviate this as PWM). PWM is an indicator of how close the predictions are to the true change point made by each of the techniques. Note that in our synthetic data tests we are able to show the detection delay because we know where the true change point is. However, in real world data, we cannot reliably know where the true change point is and thus unable to reliably know the detection delay.

Table 1 shows the results when we vary the stream variance from $\sigma = 0.0050$ to 0.0200. The SEED Delay values shows where the detected change point is. In most cases the detected change point is at least more than 150 steps away from the true change point. This indicates that if we used the detected change point as a rough estimate of the true change point, the results are relatively inaccurate.

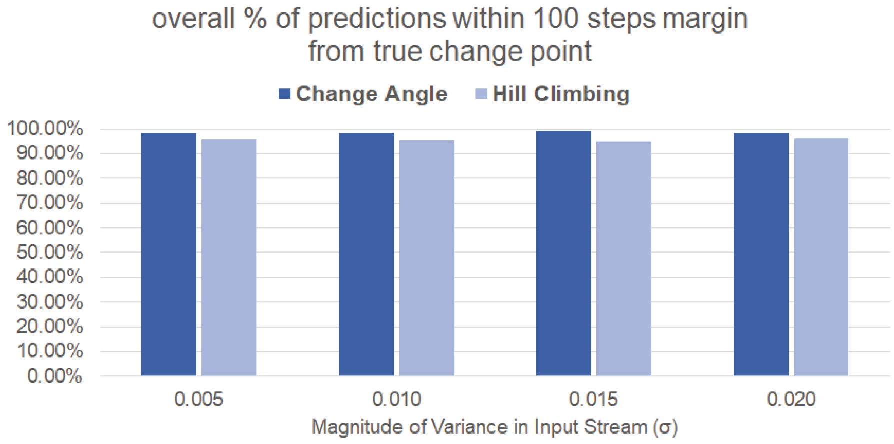


Fig. 6. Change angle vs. Hill climbing

We can also compare the performance of the best Hill Climbing results versus Change Angle in Table 1. A mean value at 0 represents an exact on-point estimation, while positive values represent an estimation that is later than the true change point, and negative values represent an estimation that is earlier

Table 1. Detection delay with varying fluctuation in data input

	SEED Delay	**Hill Climbing**		**Change Angle**	
	$\sigma = 0.0050$				
Slope	mean±sd	mean±sd	PWM	mean±sd	PWM
0.0001	309.7±16.0	-15.1±46.0	0.94	4.4±64.4	**0.98**
0.0002	231.0±0.0	-26.1±33.1	0.99	29.1±14.8	**1.00**
0.0003	192.6±12.8	-24.6±35.7	0.95	4.4±17.7	**0.96**
0.0004	167.0±0.0	-22.9±43.3	0.95	7.0±0.0	**1.00**
	$\sigma = 0.0100$				
Slope	mean±sd	mean±sd	PWM	mean±sd	PWM
0.0001	324.1±9.2	4.8±52.6	0.90	15.0±148.0	**0.95**
0.0002	231.0±0.0	-16.5±43.7	0.99	9.2±8.2	**1.00**
0.0003	198.4±4.5	-13.5±44.1	0.93	16.9±23.4	**0.99**
0.0004	167.0±0.0	-6.4±44.6	**1.00**	7.0±0.0	**1.00**
	$\sigma = 0.0150$				
Slope	mean±sd	mean±sd	PWM	mean±sd	PWM
0.0001	328.0±8.4	1.9±50.3	0.89	23.3±34.0	**0.98**
0.0002	231.0±0.0	-3.0±43.3	0.95	0.9±12.6	**1.00**
0.0003	198.7±3.2	1.5±47.0	0.96	7.0±20.7	**0.99**
0.0004	167.0±0.0	-2.0±40.0	0.99	-3.6±15.1	**1.00**
	$\sigma = 0.0200$				
Slope	mean±sd	mean±sd	PWM	mean±sd	PWM
0.0001	334.7±14.4	12.0±74.7	0.89	1.9±130.1	**0.95**
0.0002	234.2±9.6	9.2±45.4	0.97	8.7±33.7	**0.99**
0.0003	198.7±3.2	2.6±46.8	**0.99**	4.4±20.6	**0.99**
0.0004	167.0±0.0	2.2±38.1	0.99	16.7±14.0	**1.00**

*PWM is the proportion of values within ±100 steps from true change point

than the true change point. We observed that Change Angle in all test cases had at least more than 95% of the true change point estimations fall within ±100 steps of the true change point and generally performs better or on par with offline Hill Climbing method. A comparison of the overall performance is shown in Fig. 6. It shows that Change Angle outperforms hill climbing in all cases. Note that we observed the distribution of the estimations tends to normal distribution, but sometimes one or two outliers does skew the mean and standard deviations reported. Hence, the area under the curve from a normal distribution constructed from the raw mean and standard deviation values might not always equal the PWM measure. We calculated the PWM from the actual results of the estimations in each of the iterations.

In Fig. 7 we present the summary performance of Hill Climbing while varying the speed parameter. The speed parameter can be tricky to set. A speed parameter too high might mean we miss the actual true change point, but a speed parameter too low might mean we fall into a local minima before finding

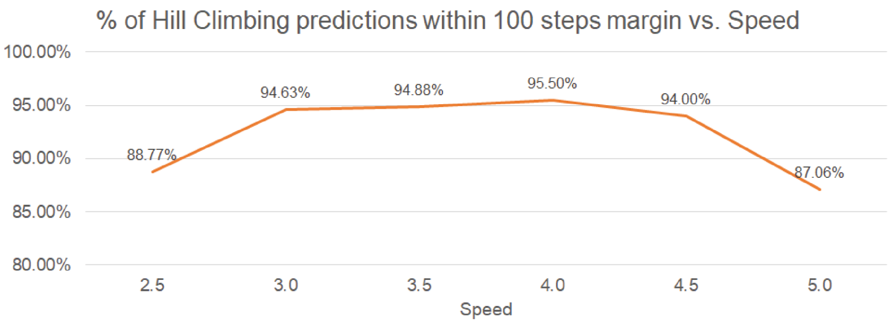


Fig. 7. Hill climbing with varying speed

Table 2. Varying noise in data input on change angle PWM* value using $\alpha = 2$

	Gaussian Distribution				
Slope	$\sigma = 0.0050$	$\sigma = 0.0100$	$\alpha = 0.0150$	$\sigma = 0.0200$	mean
0.0001	0.98	0.95	0.98	0.95	0.97
0.0002	1.00	1.00	1.00	0.99	1.00
0.0003	0.96	0.99	0.99	0.99	0.98
0.0004	1.00	1.00	1.00	1.00	1.00
mean	0.99	0.99	0.99	0.98	**0.99**
	Exponential Distribution				
Slope	$\sigma = 0.0150$	$\sigma = 0.0300$	$\alpha = 0.0450$	$\sigma = 0.0600$	mean
0.0001	0.98	0.94	0.80	0.65	0.84
0.0002	1.00	0.99	0.95	0.96	0.98
0.0003	1.00	0.99	1.00	0.98	0.99
0.0004	1.00	1.00	1.00	0.95	0.99
mean	1.00	0.98	0.94	0.89	**0.95**
	Uniform Distribution				
Slope	$\sigma = 0.0150$	$\sigma = 0.0300$	$\alpha = 0.0450$	$\sigma = 0.0600$	mean
0.0001	0.99	0.92	0.81	0.82	0.89
0.0002	1.00	1.00	0.98	0.93	0.98
0.0003	1.00	1.00	1.00	1.00	1.00
0.0004	1.00	1.00	1.00	0.99	1.00
mean	1.00	0.98	0.95	0.94	**0.97**

*PWM is the proportion of values within ±100 steps from true change point the true change point. We empirically look at the most suitable speed parameter. We can see the effect of falling into a local minima with a slower speed (*speed* = 2.0) with a falling PWM. At speed around 3.0 to 4.5 the performance peaks for this experimental setting with *speed* = 4.0 yielding the overall highest PWM, hence we used *speed* = 4.0 for Hill Climbing to compare against Change Angle.

Table 2 shows the results of different noise distribution and magnitude using constant $\alpha = 2$ for Change Angle. We included data with noise generated from a Gaussian distribution, Uniform distribution, and Exponential distribution (natural logrithmic). We can see from the table that the performance of Change Angle is in general consistent across streams of increasing noise, except when the slope is 0.0001, where increasing the noise drops the mean PWM below 0.90. These results first provide some empirical evidence that setting $\alpha = 2$ is sensible for these stream conditions and second show that Change Angle will still work with different types and magnitudes of noise.

5 Conclusion and Future Work

The development process of large software systems, such as those at Mozilla, involves complex revision control and code integration. During this process, software bugs inevitably occur. In this paper we have tackled the quality assurance task of identifying errors and identifying the cause of the error. We focused on resolving intermittent bugs and we provided automated tool solutions that used machine learning and change mining approaches and theories. In particular, we implemented a prototype that provides the Sheriffs at Mozilla with a prioritized list of key intermittent bugs. We then proposed the novel idea of Change Angle which discovers the starting location of a change in the data that leads the engineers to the location of the cause of the intermittent bugs. Our evaluations on the Mozilla data have shown that the prototype and approaches we have developed are efficient and help engineers at Mozilla tackle and resolve intermittent bugs more effectively. Our evaluations on the synthetic data using standard testing conditions showed, for the first time, that we can more accurately pin-point the location of when a change starts to occur in a data stream when compared against existing approaches.

In the future we want to pursue two directions to build on top of our current work. First, we want to investigate ways to further improve the accuracy of Change Angle in pin-pointing true change points under stream conditions where gradual changes are frequently seen. Currently, it is capable of estimating the true change point accurately in both gradual and abrupt changes, but can very occasionally produce inaccurate off-estimation for gradual changes. We would like to refine the way Change Angle is estimating the results by mapping the trajectory of the change based on the running accuracy of change angle. This will in theory improve the accuracy of Change Angle estimations in streams experiencing frequent gradual changes. Second, we want to use the information gained from Change Angle to mine additional knowledge from the data. The first thing we want to do is to characterize the magnitude of change in data streams. Change Angle can provide users with an angle value that can be interpreted as the magnitude of a change in the data. An example is an angle of 180° (*i.e.* no change) while an angle of 90° means a very abrupt change. The additional discovery of the magnitude of a change provides additional information about the nature of the change to the users and the systems involved.

Acknowledgment. We thank all Mozilla engineers that were involved in the development of this research project especially Kyle Lahnakoski, Joel Maher, and Jonathan Griffin for helping us acquire access to the relevant data and validate the work.

References

1. Baena-Garcıa, M., del Campo-Ávila, J., Fidalgo, R., Bifet, A., Gavaldà, R., Morales-Bueno, R.: Early drift detection method. In: Proceedings of the 4th International Workshop on Knowledge Discovery from Data Streams, pp. 77–86 (2006)
2. Bifet, A., Gavaldà, R.: Learning from time-changing data with adaptive windowing. In: Proceedings of the 7th SIAM International Conference on Data Mining (SDM), pp. 443–448 (2007)
3. Bifet, A., Gavaldà, R.: Adaptive learning from evolving data streams. In: Adams, N.M., Robardet, C., Siebes, A., Boulicaut, J.-F. (eds.) IDA 2009. LNCS, vol. 5772, pp. 249–260. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03915-7_22
4. Domingos, P., Hulten, G.: Mining high-speed data streams. In: Proceedings of the 6th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), pp. 71–80 (2000)
5. Gama, J., Medas, P., Castillo, G., Rodrigues, P.: Learning with drift detection. In: Bazzan, A.L.C., Labidi, S. (eds.) SBIA 2004. LNCS (LNAI), vol. 3171, pp. 286–295. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28645-5_29
6. Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., Bouchachia, A.: A survey on concept drift adaptation. ACM Comput. Surv. **46**(4), 1–35 (2014)
7. Hoeffding, W.: Probability inequalities for sums of bounded random variables. J. Am. Stat. Assoc. **58**(301), 13–30 (1963)
8. Huang, D.T.J., Koh, Y.S., Dobbie, G., Bifet, A.: Drift detection using stream volatility. In: Appice, A., Rodrigues, P.P., Santos Costa, V., Soares, C., Gama, J., Jorge, A. (eds.) ECML PKDD 2015. LNCS (LNAI), vol. 9284, pp. 417–432. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23528-8_26
9. Huang, D.T.J., Koh, Y.S., Dobbie, G., Pears, R.: Detecting volatility shift in data streams. In: Proceedings of the 10th IEEE ICDM, pp. 863–868 (2014)
10. Kanji, G.K.: 100 Statistical Tests. Sage Publications, London (2006)
11. Page, E.S.: Continuous inspection schemes. Biometrika **41**, 100–115 (1954)
12. Pears, R., Sakthithasan, S., Koh, Y.S.: Detecting concept change in dynamic data streams. Mach. Learn. **97**(3), 259–293 (2014)
13. Ross, G.J., Adams, N.M., Tasoulis, D.K., Hand, D.J.: Exponentially weighted moving average charts for detecting concept drift. Pattern Recogn. Lett. **33**(2), 191–198 (2012)
14. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach, vol. 25 (1995)

Implementation and Performance Analysis of Data-Mining Classification Algorithms on Smartphones

Darren Yates[1](✉), Md. Zahidul Islam[1], and Junbin Gao[2]

[1] School of Computing and Mathematics, Charles Sturt University,
Panorama Ave, Bathurst, NSW 2795, Australia
{dyates,zislam}@csu.edu.au

[2] University of Sydney Business School, The University of Sydney,
Sydney, NSW 2006, Australia
junbin.gao@sydney.edu.au

Abstract. Smartphones are increasingly being used to capture data and perform complex tasks, however, this rarely extends to the local training of data models. This study investigates the implementation of data mining classification algorithms on smartphones, using 20 classifiers and nine mixed-design datasets on three devices. Their accuracy and processing speed are further compared against a laptop computer using our cross-platform 'DataBench' software. Results show that smartphones not only deliver classification accuracy equal to that of more powerful computers when using the same algorithms, as expected, but also, as many as 75% of the 180 algorithm/dataset learning tasks tested were completed on smartphone hardware within three seconds. However, tests further show that the increased complexity of newer algorithms searching for ever greater classification accuracy is resulting in model-build times growing at an exponential rate. Additional testing identified that while a single algorithm execution can have negligible effect on battery life, power efficiency is affected by algorithm complexity, data size and attribute type. The increased processing demand of local model-learning on smartphones also results in increased power dissipation. Yet, even on a continuous-loop execution basis, mobile temperature gains over a 15-min period did not exceed 7 °C. Our conclusion is that smartphones are ready to form self-reliant mobile data-mining solutions able to efficiently execute a wide range of classification algorithms. This offers numerous advantages, including data security and privacy improvements, removal of reliance on network connectivity and delivery of personalised learning.

Keywords: Battery life · Classification · Data mining · Smartphones

1 Introduction

The arrival of smartphones has changed not only the way we communicate, but also how we compute. As the technology continues to improve, mobile devices have matured from phones executing basic applications to become mobile computers with multi-core processors, on-board sensors and fast wireless communications. This has led

R. Islam et al. (Eds.): AusDM 2018, CCIS 996, pp. 331–343, 2019.
https://doi.org/10.1007/978-981-13-6661-1_26

to considerable research into the use of these devices in a wide array of fields from personal health [1] to agriculture [2]. Yet, for many of these applications, smartphones are only used to gather data, either from the user directly or the device's built-in sensors. The data is then transferred to server storage for off-site processing.

Nevertheless, since the smartphone's arrival, attempts have been made to incorporate local machine-learning and data-mining into the device. The 'acquisitional context engine' (ACE) [3] and 'MobileMiner' [4] apps provided early but limited functionality. However, with ACE designed for Windows Mobile 7.5 devices and MobileMiner running on Samsung's Tizen operating system, our review of existing literature has found, to our knowledge, no substantial body of research into the performance of classification algorithms on current Android smartphone-grade hardware.

This paper presents test results from a prototype design incorporating 20 well-known classification algorithms into an application for Android devices. Moreover, it provides a detailed performance comparison, with model build times over nine publicly-available datasets recorded on three different Android devices. Further tests are conducted on device battery life and temperature while executing these algorithms.

There are numerous benefits of operating localised machine learning on mobile devices. First, no wireless connectivity is required, since all collection, processing and machine-learning tasks can be performed on the device itself. This enables remote-location applications, where wireless connectivity may not be guaranteed. Second, data security and privacy are significantly improved, since no data is required to leave the mobile device. Third, models learned on the device can be reapplied to that device quickly without need for external processing, speeding up implementation. Fourth, locally-learned models provide greater scope for personalised feedback to the user, particularly in the broad and growing research field of e-health care.

This paper continues with Sect. 2 backgrounding previous research efforts in applying data mining to smartphones. Section 3 describes the experiments and test procedures, while Sect. 4 presents the test results. Section 5 offers analysis and discussion on future directions and finally, Sect. 6 concludes this paper.

2 Related Work

Smartphones are a rich source of personal data, thanks to their growing array of built-in sensors, from accelerometers to environmental sensors recording temperature and humidity. Yet, despite the availability of high-level on-board processing, the use of smartphones in data applications is predominantly passive from an existing-literature viewpoint and falls within three broad categories.

The most common category is data-collection only. In this setting, user data is captured by the smartphone and wirelessly forwarded onto remote services for analysis, with the smartphone itself taking no further part. For example, smartphones are increasingly being used to monitor patients' depression levels. One such approach uses the Patient Health Questionaire-9 (PHQ-9) to gather responses from smartphone users. These responses are anonymised and transferred to a remote server, however, no local machine-learning is conducted by the phone itself [5].

A second, but less-common category consists of data-collection and execution. Here, the smartphone gathers data and forwards it onto remote servers for processing as before, but then receives back a set of rules or 'model' that can be executed locally on the mobile device. An example of this is PredictRisk [6], an Android application designed to predict the risk of heart attack amongst users. The application employs an existing model learned from previous data, activated via user answers to an in-app questionnaire. While the user receives back an automated predicted response from the current built-in model, the new data is transferred to remote servers and added to existing data for future learning.

However, while this category utilises the phone's on-board processing for end-case decision-making from these rules, the initial, more complex data-mining process is still completed remotely via a separate computer. As a result, from the perspective of requiring network connectivity and cloud computing, this category could be said to offer little improvement. Moreover, from a data security and privacy viewpoint, the process of transmitting the original data to remote servers and receiving back a model of that data potentially may offer a second attack vector for obtaining personal information. Thus, at minimum, the data should be pre-processed to remove user-identifying content before initial transmission.

A third and far-less-common category is the closed-loop, whereby the smartphone collects, mines and uses the data captured from the user or the device's on-board sensors. It is also the most complex option to implement and to our knowledge, has received only limited research. The 'Acquisitional Context Engine' (ACE) [3] learns user contexts from mining phone sensor data, but was set for Boolean (true/false) attributes only and implemented on Microsoft Windows Phone 7.5 operating system. MobileMiner [4] was a 'general purpose service' to mine frequently occurring patterns from phone sensor and log data through a custom algorithm called WeMiT (Weighted Mining of Temporal Patterns). However, MobileMiner was designed for the Samsung Tizen operating system and does not support classification mining. A rare example of this category aimed at Android devices focused on human activity recognition by incorporating Naïve Bayes and a custom 'K-Nearest Neighbours' algorithm into an Android app for local model-learning [7]. The app predicted running, walking, standing and sitting states by modelling locally-sourced data.

Overall, however, the review of existing research highlights the lack of detailed literature involving model learning of data on mobile devices. Moreover, where reasons have been given for not using smartphones in model-learning applications, these have included insufficient storage, processing power or battery life. Yet, to our knowledge, there is no published research that quantifies the accuracy and speed of a broad range of classification algorithms when executed on smartphone technology.

3 Our Experimental Design and Contribution

This paper implements a number of original contributions to expand understanding in classification algorithm implementation and execution on mobile Android devices. These include development of a testing system combining a broad collection of classification algorithms and training datasets for execution on multiple smartphones, covering four key aspects of smartphone capability – accuracy, speed, battery life and device temperature. Moreover, to allow cross-comparison of results with traditional Windows computers, this testing system features a cross-platform (Windows, Android) application called ‘DataBench’. It incorporates elements of the open-source Weka data-mining engine, developed by the University of Waikato [8] and provides opportunity to study the execution of algorithms on different dataset types, along with their performance on varying grades of mobile device.

3.1 Algorithms Tested

In order to provide a broad base of understanding, 20 classification algorithms were selected for inclusion into the DataBench applications: Bagging (Bag), BayesNet (BNet), Classification via Clustering (CvC), Classification via Regression (CvR), ConjunctiveRule (CR), DecisionTable (DT), IBk (K-nearest neighbour), J48 (C4.5), LogitBoost (LB), MultiBoostAB (MAB), NaiveBayes (NB), OneR, PART, RandomCommittee (RC), RandomForest (RanF), RandomTree (RT), REPTree, RotationForest (RotF), SimpleCart (SC) and ZeroR. These algorithms are contained within the Weka data-mining suite [8], and deployed in the ‘weka.jar’ Java archive file. While Weka is noted to support Windows, macOS and Linux operating systems, this support does not officially extend to the Android operating system, despite the Weka source code being developed in Java, Android’s native programming language. Nevertheless, the Weka v3.6.15 engine was successfully imported into the Android version of our DataBench application.

3.2 Datasets Tested

In addition to the 20 classification algorithms, nine datasets were chosen to train with these algorithms and selected to create diversity in the number of instances and attributes, as well as attribute type. These datasets included ‘Ecoli’, ‘Contraceptive Method Choice’ (CMC), ‘Car Evaluation’, ‘Wine Quality’ (WQ), ‘Chronic Kidney Disease’ (CKD), ‘Hypothyroid Disease’, ‘Soybean’, ‘Mushroom’ and ‘Nursery’. Details of these datasets are shown in Table 1. All datasets are publicly available at the University of California, Irvine (UCI) Machine Learning Repository [9], except for ‘Soybean’, which is bundled into the Weka data mining suite download.

3.3 Hardware Tested

Three smartphones were selected for these tests - a Samsung Galaxy SIII GT-I9300 (released in 2012), an LG F70 D315 (2014) and a Motorola Moto G5 (2017). An HP Pavilion dm4-3114tx laptop PC was chosen as the Windows personal computer to act as the point-of-reference. The hardware specifications are shown in Table 2.

Table 1. Publicly available datasets used in testing.

Dataset	Instances	Attributes	Attribute type	Missing values
Ecoli	336	8	Numeric	No
CMC	1473	9	Mixed	No
Car Eval	1728	6	Categorical	No
Wine Qual.	4898	12	Numeric	No
Chr. Kidney. Dis.	400	25	Mixed	Yes
Hypothyroid	7200	21	Mixed	No
Soybean	683	36	Categorical	No
Mushroom	8124	22	Categorical	Yes
Nursery	12960	8	Categorical	No

Table 2. Hardware specifications for devices used in experimentation.

Device model	Samsung Galaxy SIII (I9300)	LG F70 D315	Motorola Moto G5	HP Pavilion dm4-3114tx
Device type	Smartphone	Smartphone	Smartphone	Notebook PC
Processor/SoC	Samsung Exynos 4412	Qualcomm SD400	Qualcomm SD430	Intel Core i5-3210M
CPU cores	4	4	8	2
CPU architecture	ARM Cortex-A9	ARM Cortex-A7	ARM Cortex-A53	Intel x86_64
Memory (RAM)	1 GB	1 GB	2 GB	4 GB
Internal storage	16 GB (Flash)	8 GB (Flash)	16 GB (Flash)	640 GB (HDD)
Battery capacity	2100 mAh	2440 mAh	2800 mAh	N/A
Operating system	Android 4.3	Android 4.4	Android 7.0	Windows 10

3.4 Tests Conducted

All 180 combinations of 20 classification algorithms and nine datasets were trained on all three smartphones as well as the Windows notebook PC acting as reference. Classification accuracy and model training or ‘build’ times were recorded for each device. Moreover, additional continuous-loop tests were conducted to determine the effect of local machine-learning on smartphone battery life and device temperature.

Classification Accuracy

Classification accuracy was tested by training a model using each algorithm/training dataset combination, then evaluating the model using 10-fold cross-validation. This process involves dividing the training dataset into ten roughly-equal partitions or ‘folds’. Nine of the ten folds are used to train the model while the tenth is used for testing. Each of the folds takes turn to act as the testing fold, such that by the process end, every dataset instance has been used to test the learned model exactly once [10]. Classification accuracy is recorded as the number of instances correctly classified by the model as a percentage of the total number of instances in the training dataset.

Classification Model Build/Training Time

The time taken for each device to build a model using the selected algorithm and training dataset is recorded, with the test repeated twice and the three scores averaged to minimise spurious results. All times are recorded in seconds and measured programmatically via the device's internal clock systems. Timing begins immediately prior to the algorithm being launched and stops immediately after its completion.

Battery Life

To understand the effect of local machine-learning on smartphone battery life, tests were conducted using two of the experiment's classification algorithms on all nine datasets and all three smartphones. An increase in power consumption is a common side-effect arising from an increase in device processor utilisation and these tests were designed to shed light on how smartphones handle training a model. However, rather than look at battery life through the usual lens of Central Processing Unit (CPU) utilisation, whereby the percentage of CPU utilisation is recorded, it was decided that recording the number of models built within a predefined portion of the smartphone's battery life would be more informative. A more direct representation could then be made between battery life and classification algorithm model training.

The process for this involves continuously building models using the preselected classification algorithm and training dataset whilst running down the smartphone over a 10% range of battery life from 99% to 89% of capacity. Since the energy consumed during model training is a combined function of the battery capacity, processor efficiency and algorithm tested, the number of models trained over that capacity range gives a practical indication of the processing demands of each combination of classification algorithm, dataset and smartphone device. These tests involved all training datasets, all three smartphones and the 'J48' and 'REPTree' classification algorithms.

Device Temperature

Previous research indicates that increased temperature reduces long-term reliability of electronic circuits and computer processors [11]. Thus, any significant temperature gain as a result of locally-executed data mining could be noteworthy. Preliminary thermal testing was conducting on all three phones using the same methodology as described above, but with the RotationForest/Nursery algorithm/dataset combination. This selection represents the extremes of complexity and dataset size available. Thermal testing involved, first, powering up each phone into 'airplane' (no wireless connectivity) mode, screen set to minimum brightness and left to idle for 30 min to allow device temperature to stabilise. The RotationForest/Nursery combination was then executed in a continuous model-training loop. Temperature was recorded at equidistant locations around the device's rear cover over a 15-min period at one-minute intervals via a laser-guided infrared thermometer with 0.1 °C precision. The assumption is that continuous model-training will provide a worst-case thermal load and offer understanding of mobile machine learning from a thermal perspective.

4 Results

Results of the tests described in Sect. 3 are detailed here in four sections – classification accuracy (Sect. 4.1), processing speed (Sect. 4.2), smartphone battery life (Sect. 4.3) and device temperature (Sect. 4.4).

4.1 Classification Accuracy

It is a logical conclusion that if the same source code is executed successfully on two separate devices of differing architecture, the output of the two devices should be identical, despite any architectural differences. Nevertheless, classification accuracy tests using all 180 algorithm/dataset combinations were conducted on all three smartphones as well as the Windows PC and, as expected, the classification accuracies were identical in all cases. Thus, in terms of accuracy, smartphones as a computing platform are no impediment to data-mining or machine-learning. The classification accuracy results, shown as percentages, for models built using all 20 classification algorithms and nine datasets are shown in Table 3. For example, the classification accuracy of the J48 algorithm model built on the Ecoli dataset was 89.3%. The final two columns of Table 3 show the average classification accuracy percentage and standard deviation for each algorithm across all nine test datasets.

4.2 Classification Processing Speed

To accurately test the phones' processing speed for local data-mining, the DataBench software performed all 180 algorithm/dataset model build/learning combinations three times, with the average time of all three iterations recorded. The cumulative density function (CDF) graphs for each device shown in Fig. 1 display the rate of completion as a function of time and exhibit notable similarities. Models trained on the HP Pavilion dm4 laptop completed the test combinations with 50th, 75th and 90th percentiles of 0.03, 0.15 and 0.72 s, respectively. By way of comparison, the same percentiles for models trained on the Motorola Moto G5 phone (Fig. 1a) occurred at 0.24, 2.76 and 11.05 s, the Samsung Galaxy S3 (b) at 0.89, 10.41 and 44.42 s and the LG F70 D315 phone (c) 1.23, 14.08, 52.26 s. Despite not having the same processing power as the HP laptop, the completion of 75% of its tests within three seconds suggests the Moto G5 and similar phones have sufficient processing speed to handle many modelling tasks within a user-friendly timeframe.

However, comparing the Moto G5 phone's average classification accuracy versus average build time over the nine datasets in Fig. 2 highlights the minor accuracy gains from rising algorithm complexity have an exponential cost in execution time.

4.3 Smartphone Battery Life

As discussed in Sect. 3.4, J48 (C4.5) and REPTree were selected as algorithms to discover the effects of their execution on smartphone battery life. This was done by continuous execution of the algorithm to learn models from each of the nine datasets, in turn, on each smartphone. The numbers of models built during the 10% run-down in

Table 3. Classification accuracy (tested via 10-fold cross-validation) of the 20 classification algorithms over the nine datasets featured in the experiment (scores as percentages)

ID	D'set/Algor.	Ecoli	CMC	Car Eval	WQ	CKD	Thyroid	Soybean	Mushroom	Nursery	Avg. Acc.	Std Dev.
1	RotF	91.1	53.6	98.7	68.6	99.0	99.5	95.0	100	99.9	**89.5**	15.8
2	RanF	89.3	51.7	94.5	70.2	99.8	99.4	92.1	100	99.1	**88.4**	15.7
3	SC	90.8	55.2	97.1	59.3	97.5	99.5	91.1	99.9	99.6	**87.8**	16.7
4	RC	86.3	49.6	94.4	66.6	100	99.3	93.0	100	98.4	**87.5**	16.8
5	CvR	90.8	54.7	96.8	55.2	98.3	99.4	92.8	100	99.5	**87.5**	17.7
6	PART	89.3	49.2	95.8	60.1	98.5	99.4	91.9	100	99.1	**87.0**	17.8
7	Bag	90.8	52.4	92.0	63.3	98.5	99.5	86.2	100	97.3	**86.7**	16.2
8	J48	89.3	52.1	92.4	58.1	99.0	99.6	91.5	100	97.1	**86.6**	17.2
9	IBk	87.8	44.3	93.5	65.4	95.8	91.5	91.2	100	98.4	**85.3**	17.4
10	LB	89.9	56.2	86.7	53.3	99.8	99.6	93.0	98.2	90.3	**85.2**	16.9
11	REPT	90.2	52.8	87.7	55.7	96.8	99.6	84.8	100	96.0	**84.8**	17.1
12	DT	87.5	54.3	91.0	52.5	99.0	99.3	84.3	100	94.7	**84.7**	17.5
13	RT	87.5	46.6	83.2	62.6	95.5	97.1	84.0	100	94.6	**83.5**	16.8
14	BNet	85.7	51.1	85.7	48.5	98.8	98.6	93.3	96.2	90.3	**83.1**	18.4
15	NB	85.4	50.8	85.5	44.3	95.0	95.3	93.0	95.8	90.3	**81.7**	18.7
16	OneR	87.5	48.0	70.0	45.8	92.0	96.2	40.0	98.5	71.0	**72.1**	21.7
17	MAB	87.2	42.7	70.0	44.9	94.5	95.4	28.0	94.5	66.3	**69.3**	24.3
18	CR	86.9	42.7	70.0	48.9	94.8	97.1	26.2	88.7	66.3	**69.0**	23.8
19	CvC	72.0	39.4	54.9	45.1	71.8	76.0	22.5	75.9	40.8	**55.4**	18.4
20	ZeroR	77.1	42.7	70.0	44.9	62.8	92.3	13.5	51.8	33.3	**54.3**	22.6

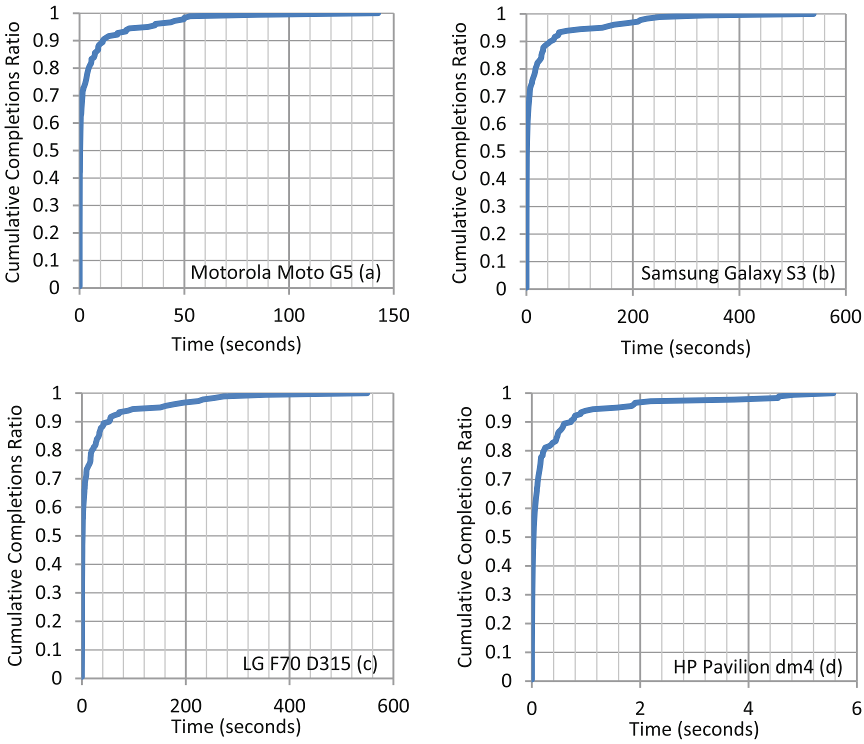


Fig. 1. Cumulative Distribution Function (CDF) graphs of model build times for the 180 classification algorithm/dataset tests executed on (a) Motorola Moto G5 phone, (b) Samsung Galaxy S3 phone, (c) LG F70 D315 phone and (d) HP Pavilion dm4 notebook.

battery capacity for each dataset are shown in Table 4. For example, the Moto G5 phone completed the J48 algorithm on the 'Car Eval' dataset 52,767 times in just 10% battery capacity. Moreover, the average run-times to drop 10% of battery capacity were 3,554secs (Moto G5), 3,275secs (LG F70) and 1,376secs (Galaxy SIII).

4.4 Smartphone Temperature Performance

The results of thermal testing are shown in Fig. 3. These detail the change in temperature whilst executing the RotationForest algorithm continuously on the Nursery dataset on all three phones. For comparison, the temperature change over the same period was recorded on the Galaxy S3 phone while playing 720 × 576-pixel MPEG-4 video. The Galaxy S3's relatively-short 10% battery run-time despite a battery capacity comparable to the other two phones suggests a higher comparative energy usage. This manifested itself here with the largest temperature gain of 6.6 °C after 15 min of continuous execution. This was followed by the LG F70 increasing its temperature by 6.0 °C. By comparison, the Motorola Moto G5 saw a temperature rise of just 2.6 °C after 15-min, 0.2 °C less than the Galaxy S3 playing an MPEG-4 video file. The ambient temperature during testing was nominally 23 °C (thermostat-controlled).

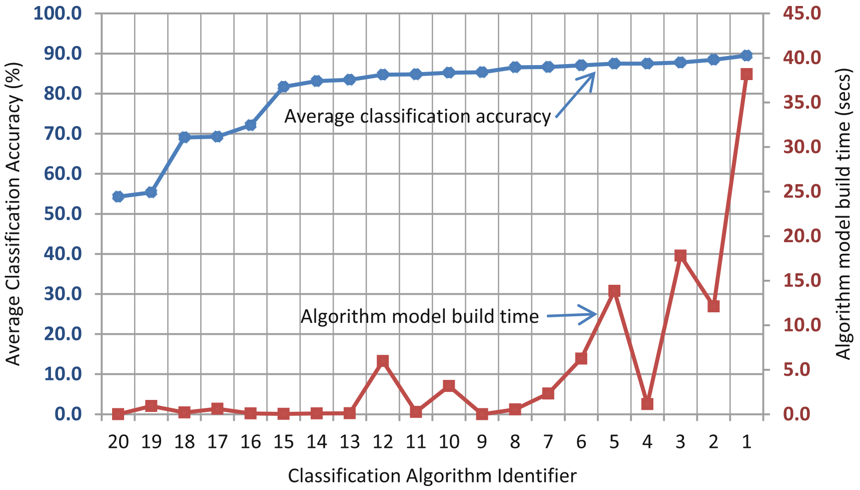


Fig. 2. Comparison of average classification accuracy (Table 3) versus average model learning time across all nine test datasets recorded from a Motorola Moto G5 smartphone. Algorithm identifiers match those in Table 3.

5 Discussion

The purpose of this research has been to identify how well popular classification algorithms perform on smartphone-grade hardware. The decline in processing speed exhibited in Fig. 2 by some of the more complex algorithms, such as SimpleCART and Rotation Forest, is significant compared with the smaller increase in classification accuracy they provided. More broadly, these results suggest that classification accuracy and processing speed are becoming mutually-exclusive, as the complexity of newer classification algorithms increases. Further, this may limit the value of some multi-tree algorithms such as Rotation Forest in mobile applications for all but smaller datasets. However, the ability to successfully weave the Weka data-mining engine into Android may encourage the development of more machine-learning applications for mobile devices to take advantage of the benefits these devices have to offer:

1. Our experiments have shown smartphones are capable of executing a wide range of data-mining classification algorithms.
2. As expected, smartphones achieve classification accuracy identical to laptops and personal computers when using the same algorithms.
3. Smartphones were able to complete as many as 75% of our algorithm-dataset test combinations within three seconds, indicating user-friendly performance levels.
4. The results suggest that continued improvement in smartphone CPU performance will translate into faster model build speeds.
5. Classification performance is dependent upon the device processor, algorithm and dataset size, yet, a single model build can have negligible effect on battery life.

6. Smartphone temperature rises during testing never exceeded 7 °C, even amidst a 15-min continuous data-mining operation. Moreover, this 15-min period exceeded the individual execution time of all 180 algorithm-dataset test combinations.

Table 4. Number of models built over a 10% range of battery life.

Dataset	Ecoli	CMC	Car Eval	Wine Quality	CKD	Thyroid	Soybean	Mushroom	Nursery
Smartphone	*J48 (C4.5) algorithm*								
Motorola Moto G5	143465	12715	52767	980	57489	7052	19504	8333	5859
LG F70 D315	23757	1984	8383	144	9907	1213	3545	1170	970
Samsung Galaxy S III	10861	1092	4095	73	4689	621	1852	334	572
	REPTree algorithm								
Motorola Moto G5	132523	27638	60543	3306	62892	7543	20198	5615	5361
LG F70 D315	25919	5537	10936	612	11998	1304	4111	896	966
Samsung Galaxy S III	10618	2658	4812	296	5591	610	1843	451	477

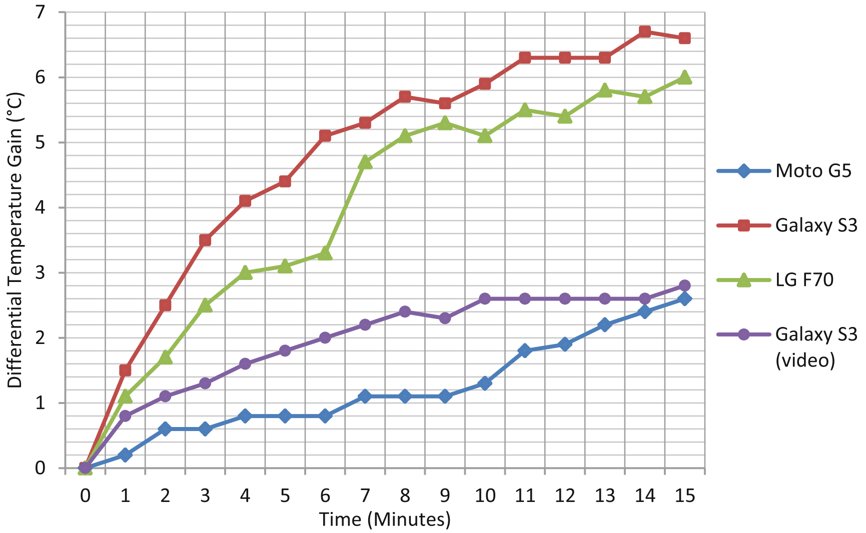


Fig. 3. Temperature rise over time of three devices executing the RotationForest algorithm on the Nursery dataset recorded over a 15-min period at one-minute intervals.

Figure 2 shows the algorithms that perform more slowly on smartphone hardware. However, it also indicates those algorithms that provide an optimal mix of classification accuracy and processing speed. For example, Random Committee (#4) and J48 (#8) both achieved classification accuracies within 3% of class-leader Rotation Forest. Yet, they do so in less than 3% of the time required by Rotation Forest on the same tests. Elsewhere, algorithms such as REPTree and IBk (K-Nearest Neighbours) also deliver similar levels of accuracy and performance, with only minor trade-offs.

6 Conclusion

Smartphones have inspired a wave of creative new applications, thanks to their combination of on-board resources. However, despite continual improvement in smartphone hardware, the amount of quantitative research available in machine learning execution on smartphones is limited. This paper has explored the direct implementation of data-mining algorithms on smartphones and found not only is it possible, but that in many applications, it can be applied efficiently with equal accuracy to more traditional computing devices. This paper also detailed our experimental design and testing through custom software involving multiple classification algorithms, training datasets and mobile devices. It found that smartphones are capable of performing local machine-learning through the Weka data-mining engine with user-friendly speed, but that the twin ideals of classification accuracy and model-learning speed are becoming mutually exclusive. Through experimentation, this research found that the choice of classification algorithm and size of dataset affects not only model-learning speed, but also device battery life. Yet in applications where only a single model is built, the overall effect on battery life can be negligible. Moreover, this paper observed that the processing demands of data-mining cause temperature rises within smartphones, but that these rises are moderate. However, despite achieving success with multiple algorithms executing successfully on smartphones, there is considerable room for further research into accurate, efficient on-device machine-learning methods that better enable smartphones to be utilised in a wide array of data-rich applications.

Acknowledgements. This research is supported by an Australian Government Research Training Program (RTP) scholarship.

References

1. Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M.: Internet of Things (IoT): a vision, architectural elements, and future directions. Futur. Gener. Comput. Syst. **29**(7), 1645–1660 (2013)
2. Pongnumkul, S., Chaovalit, P., Surasvadi, N.: Applications of smartphone-based sensors in agriculture: a systematic review of research. J. Sens. **2015** (2015)
3. Nath, S.: ACE: exploiting correlation for energy-efficient and continuous context sensing. In: Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services. ACM (2012)

4. Srinivasan, V., Moghaddam, S., Mukherji, A., Rachuri, K.K., Xu, C., Tapia, E.M.: Mobileminer: mining your frequent patterns on your phone. In: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing. ACM (2014)
5. BinDhim, N.F., Shaman, A.M., Trevena, L., Basyouni, M.H., Pont, L.G., Alhawassi, T.M.: Depression screening via a smartphone app: cross-country user characteristics and feasibility. J. Am. Med. Inform. Assoc. **22**(1), 29–34 (2015)
6. Raihan, M., et al.: Smartphone based ischemic heart disease (heart attack) risk prediction using clinical data and data mining approaches, a prototype design. In: 2016 19th International Conference on Computer and Information Technology (ICCIT). IEEE (2016)
7. Kose, M., Incel, O.D., Ersoy, C.: Online human activity recognition on smart phones. In: Workshop on Mobile Sensing: From Smartphones and Wearables to Big Data (2012)
8. Weka 3: Data Mining Software in Java. http://www.cs.waikato.ac.nz/ml/weka/. Accessed 5 Aug 2018
9. Dheeru, D., Karra Taniskidou, E.: UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets.html. Accessed 12 Aug 2018
10. Han, J., Pei, J., Kamber, M.: Data Mining: Concepts and Techniques. Elsevier, New York (2011)
11. Brooks, D., Dick, R.P., Joseph, R., Shang, L.: Power, thermal, and reliability modeling in nanometer-scale microprocessors. IEEE Micro **27**(3), 49–62 (2007)

Image Data Mining

Retinal Blood Vessels Extraction of Challenging Images

Toufique Ahmed Soomro[1](✉), Junbin Gao[2], Zheng Lihong[1], Ahmed J. Afifi[3], Shafiullah Soomro[4], and Manoranjan Paul[1]

[1] School of Computing and Mathematics,
Charles Sturt University, Bathurst, Australia
{tsoomro,lzheng,mpaul}@csu.edu.au

[2] The Business School, The University of Sydney, Sydney, Australia
junbin.gao@sydney.edu.au

[3] Computer Vision and Remote Sensing,
Technische Universität Berlin, Berlin, Germany
ajaigi@gmail.com

[4] Department of Computer Science and Engineering,
Chung-Ang University, Seoul, South Korea
s.soomro@vim.cau.ac.kr

Abstract. Retinal fundus examination is necessary for the early diagnosis of eye disease, especially diabetic retinopathy. Population screening often results in poor quality retinal images that complicate the automated diagnosis of retinal features, such as precise segmentation of blood vessels, microaneurysms, cotton stains, and hard exudates. Fluorescein fundus angiogram (FFA) has solved some problems, but it is invasive and has side effects. In this research work, we proposed a method of image enhancement based on contrast-sensitive steps as a valuable aid for the automatic segmentation of pathological (unhealthy) images. Experimental results based on the Digital retinal images for vessel extraction (DRIVE) and STructured analysis of the retina (STARE) databases showed that the proposed image enhancement method improved the performance over other existing methods, from 92% to 95% in accuracy and from 71% to 75% in sensitivity. This significant improvement in the contrast of retinal background images of retinal color has the potential to provide better vessel images for observing ocular diseases.

Keywords: Retinal color fundus images · Enhancement method · Segmentation of vessels

1 Introduction

Ophthalmic screening is an important part of regular health checks to detect eye abnormalities, especially among early indicators of the severity of diabetic retinopathy (DR). These abnormalities can be used early in the DR as an indicator of mild DR, non-proliferative retinopathy (NDPR) with mainly certain

R. Islam et al. (Eds.): AusDM 2018, CCIS 996, pp. 347–359, 2019.
https://doi.org/10.1007/978-981-13-6661-1_27

microaneurysms present at severe, proliferating (PDR) with new vessel growth in addition to microaneurysms, cotton wool spots and hard exudates within to complete vision loss. In addition, there is a subclassification of the NPDR according to the presence of anomalies. These rankings are: Mild NPDR, Moderate NPDR and Severe NPDR. The earlier the detection, the better treatment outcomes. The corresponding actions can help patients to prevent blindness in cases of advanced DR [2]. Therefore, the key question is how to accurately identify these anomalies in digital images and this plays a key role in the diagnostic procedure.

Images of the retinal fundus can either be captured as mydriatic or non-mydriatic eye color images, or by fluorescein injection to improve the observation of blood vessels, hemorrhages and microaneurysms [16]. High quality color images often require in particular preparation of dilation of the eyes to avoid varying degrees of contrast and noise. The quality of digital color background images is essential for doctors to accurately detect any abnormality from retinal fundus images.

For example, fluorescein angiograms (FFA) photography involves injecting a contrast agent to obtain a good contrast of the blood vessels on the retinal fundus [3]. However, FFA have side effects such as nausea, vomiting and dizziness resulting in anaphylactic shock that can be fatal [3].

However, retinal images have a low and varying contrast and in practice contain “noises”. Improving the contrast of retinal images, especially by adjusting the contrast between the vessels in relation to their background, is a crucial step in obtaining well-segmented images.

Image enhancement is an essential step to improve retinal images for accurate diagnosis. Many image enhancement techniques have been implemented that involve various tasks such as noise elimination, blurring, contrast enhancement, and changing the gray scale dynamic range [20]. Conventional image contrast enhancement techniques include histogram equalization (HE) [10] and other histogram-based techniques [4]. These methods work well for neural image analysis but have failed for other types of images due to different image properties. The best practice is to select the green channel of the retinal color image. However, a green band image also contains noise introduced by the image acquisition process. The discontinuity of the intensity level of the green channel image can be observed. Hence, an appropriate image enhancement technique is needed to solves the problem mentioned above in and to obtain a better image quality. Color enhancement is required during initial pre-processing to observe the affected vessels due to pathologies or the central reflex of light.

For the implementation of enhancing the image contrast of retinal images, the primary challenge is to deal with a low-varying contrast along with the noise in the retinal fundus images. Noise in retinal images usually is introduced by the image acquisition process such as artifact on the lens or movement of the patient. Here, a four-step image enhancement technique is proposed. Firstly, the color contrast enhancement technique by using contrast-limited adaptive histogram equalization (CLAHE) on color retinal fundus image is introduced to obtain an enhanced color image. The advantage of color enhancement of retinal fundus image is to identify information of vessels affected by pathologies. Following that, the issue of uneven illumination is resolved by using the morphological operations such as top hat and bottom hat morphological operations. The third step is to improve the low contrast problem that affects the visualization of tiny vessels. Low contrast issue is resolved by using the independent component analysis (ICA). The last step is to get an enhanced image based on highest contrast among red-green-blue (RGB) of retinal images. This process is called an image enhancement technique and is a pre-processing step. The proposed image enhancement technique is used as a pre-processing step of the segmentation model to evaluate the impact of the image enhancement technique on retinal vessel segmentation. The full pipeline of the proposed method is presented in Fig. 1.

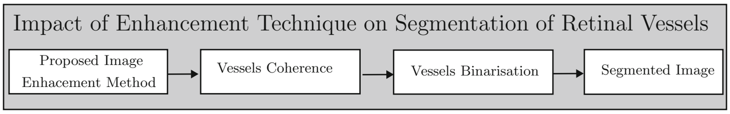

Impact of Enhancement Technique on Segmentation of Retinal Vessels
Proposed Image Enhacement Method
Vessels Coherence
Vessels Binarisation
Segmented Image

Fig. 1. Flow diagram of the incorporation of the proposed image enhancement technique into the segmentation process.

The paper is organized as follows. Section 2 explains the implementation of the proposed method. Section 3 contains the databases used in this work and the evaluation parameters. Section 4 discusses the results and the experimental outcomes. Section 5 contains a conclusion of this research paper.

2 Proposed Method

Two subsections are here to discuss the proposed method. The first subsection elaborates the theoretical reason for the implementation of the image enhancement technique, and each step of the proposed image enhancement technique is explained in detail. Section 2.2 contain the implementation of the post-processing steps to obtain the final image of the segmented vessels.

2.1 Proposed Image Enhancement Technique

As most current retinal enhancement techniques are based on the enhancement of fundus images for the healthy retinal images, the challenge of enhancing unhealthy images which suffer from low contrast, central light reflex and contains the abnormality was the primary focus. The proposed method consists of several sequential processing steps, where the aim of each step is to enhance the contrast of vessels against their background. Each stage of the proposed method is elaborated as follows.

Color Image Enhancement of Retinal Image: Due to the image acquisition process and external lighting conditions, the colored background images undergo varying variations in contrast and shading. Improved color contrast is used to remove shadows before further analysis. In this research, a color contrast enhancement method for color retinal fundus images is proposed which overcomes the shading effect and which improves the contrast in retinal fundus image. The main idea behind this technique is to analyse the intensity distribution of pixels by using the intensity histogram information via a modified CLAHE algorithm. The CLAHE method separates an image into several tiles, and adjusts the contrast such that the titles of the histogram have the required shape. For optimization, the adaptive histogram equalization algorithm is adopted to adjust the number of tiles, number of histogram bins, and clipping level. In the case of retinal images, the proposed algorithm relates to the two parameters: number of tiles and clip limit. The number of tiles considers as a particular number of rectangle contextual regions (titles) into which each image is divided. The contrast transform function is computed for each region individually. The optimal number of tiles depends on the type of input image. The second parameter, clipping limit, is a contrast ratio that avoids excessive image saturation, especially in homogeneous areas. These homogeneous areas are characterized by a high peak in the histogram of the image. The clipping limit helps CLAHE to produce better results than an original image.

Moreover, in color fundus images, the intensity values of those abnormal regions are low. Figure 2 shows the original color image and the corresponding enhanced image along with their histograms. It is evident that the contrast of retinal blood vessels is improved in the color-enhanced image, that can be observed from the histogram of the enhanced image and the histogram of the original color image. The intensity distribution in the enhanced image shows some irregularities due to uneven illumination. Thus, the following task is the removal of the non-uniform background details, can be found following in the next stage. Apparently, the color enhancement tactic works well on the retinal images in both healthy and unhealthy images as shown in the Fig. 3. Thus, blood vessels are obviously more observable than an original color image.

Uneven Illumination Removal: A color-enhanced image contains three channels: red, green and blue. Due to uneven illumination, there is a considerable pixels intensity variation in the background. It causes difficulty to identify the vessels pixels whose intensity is lower than that of the background. To get rid of affection by uneven illumination, we use morphological operations on each channel individually. Firstly, the mathematical morphology opening or closing is used to get an estimated background image. The estimated background is then subtracted from the original image. This series of operations is named as "top hat". Top hat operation removes high frequencies (considered as reflectance) and keeps low frequencies (considered as illumination). It has a black top and a white top. The black top hat is used for the clear background, and white top is adopted for a dark background. The result of these steps is shown in the in Fig. 4.

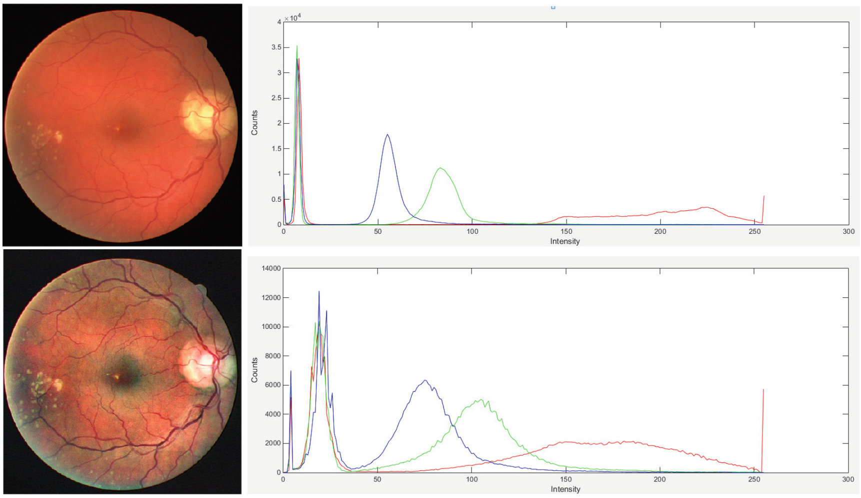


Fig. 2. Results of the color image enhancement: the first row presents the original image and its histogram (The histogram is based on kernel density estimates.). The second row presents the improved contrast color image and its histogram (The histogram is based on kernel density estimates.). (Color figure online)

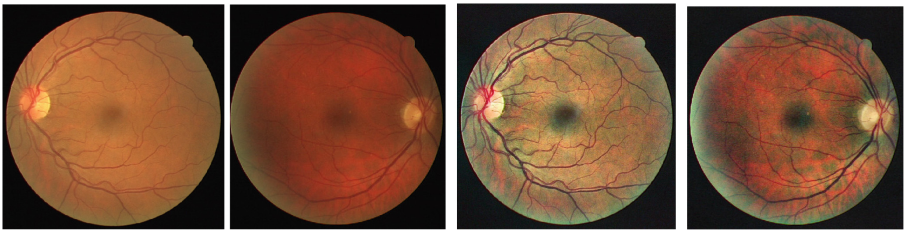

(a) (b) (c) (d)

Fig. 3. Results of image enhancement in color contrast: figures (a) and (b) show the original image of retinal funuds, figure (a) is a health retina image and figure (b) is an unhealthy retina image. Figures (c) and (d) show their enhanced color image using the proposed color enhancement technique. (Color figure online)

This process is applied on for all three channels independently. This operation though resolves major illumination discontinuity around the optical disc but still, the image suffers from the low contrast issue. Thus the Independent Component Analysis (ICA) is adopted later to enhance the contrast of each uniform background channel. This is elaborated in stage 3.

Enhancement of RGB of Retinal Image: The ICA is adopted to overcome the low contrast issue of retinal blood vessels with respect to background noise and illumination. Each RGB channel is processed through the ICA. The grey RGB retinal color channels have different properties; the red channel provided the most luminance information, as well as noise. The green channel contains the least noise, and the blue channel includes shadow as well as noise. The grey-green channel preserves the contrast of retinal blood vessels compared to red and blue channels. Independent component analysis is used to extract suitable components from the mixture components. ICA is used to enhance the three channels and to separate the three color channels according to their original representation as the retinal ICA model. The ICA components determined from this process are shown in Fig. 4. The best contrast image is selected by measuring value of contrast between blood vessels against their background. We also measured the contrast of the retinal blood vessels of non-uniform background removal images by using our proposed procedure [15,17]. The Fig. 4 clearly shows that the green ICA gives the highest contrast:51.98 compared to other red ICA contrasts:48.15 and blue:43.21. Based on this enhanced contrast, the green channel is selected as an enhanced image for improving the segmentation of retinal blood vessels.

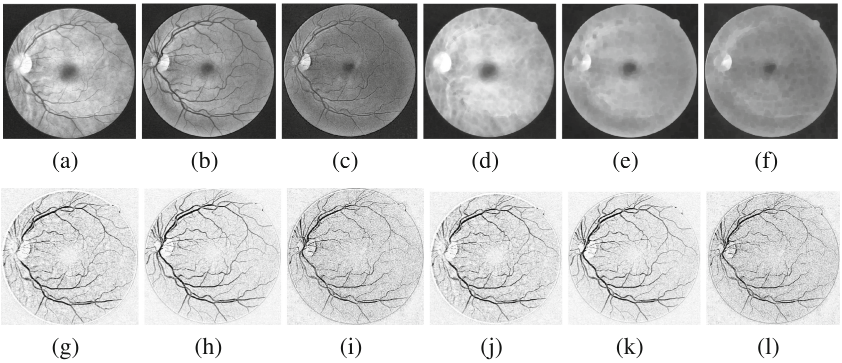


Fig. 4. Uneven illumination removal process for the red, green and blue channels. Figure (a) red input image (Contrast:18.07). Figure (b) green input image (Contrast:30.98). Figure (c) blue input image (Contrast:13.20). Figure (d) is the estimated background red image. Figure (e) is the estimated background green image. Figure (f) is the estimated background blue image. Figure (g) is the estimated uniform background red image (Contrast:28.51). Figure (h) is the estimated uniform background green image (Contrast:40.17). Figure (i) is the estimated uniform background blue image (Contrast:22.10). Figure (j) is Red ICA: ICA component 1 (Contrast:48.15). Figure (k) green ICA: ICA component 2 (Contrast:51.98). Figure (l) Blue ICA: ICA component 3 (Contrast:43.21) (Color figure online)

2.2 Post-processing Segmentation of Retinal Vessels

The post-processing segmentation contains two stages namely the coherence of the vessels and binarisation of the vessels. These steps are elaborated as follows.

Retinal Vessel Coherence: After vessel enhancement, still, there are two main issues that need to be addressed. First, the retinal blood vessels vary in their size, so it is challenging to detect retinal blood vessels in their actual size. Second, there are non-coherence levels between large and small retinal blood vessels. For this purpose, the following techniques are applied. Although large retinal blood vessels are illuminated well, small retinal blood vessels do not preserve the same intensity difference to their adjoining background. This becomes one of the main problems in the extraction of accurate retinal blood vessel images. This issue is resolved by adopting an oriented diffusion filtering, as proposed in [5]. The output of coherence retinal blood vessels is shown in Fig. 5(a). Retinal blood vessels width changes continuously. The multiscale LoG filter is employed to detect the vessels in proper width, and to make the binarisation step simpler. Figure 5(b) shows the output of multiscale LoG and retinal blood vessels extracted image. Some vessels are not detected correctly, especially those connecting small retinal vessels. This means that the final estimate accurate retinal vessel segmentation image can be achieved by doing some binarisation.

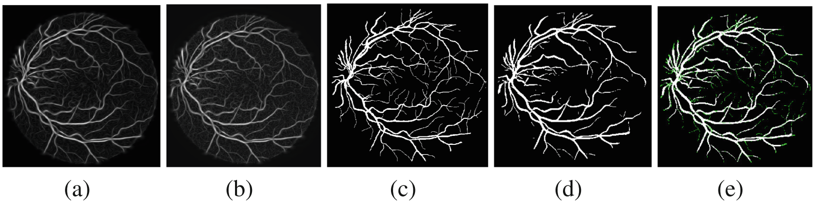


Fig. 5. Figure (a) Diffusion filter oriented output image. Figure (b) Output image of the multi-scale LoG detector. Figure (c) Ground truth image. Figure (d) Final segmented image. Figure (e) Missing small vessels (green color indicates) observation from the final segment image (Color figure online)

Retinal Vessel Binarisation: Eliminating light and dark spots from the LoG detector output resulted in a large variation in the intensity of the retinal blood vessels along their entire length. Good binarisation gives more accurate images of retinal blood vessels from their background. The concept of an improved double threshold method [13] is adopted. The algorithm is described as follows,

1. Select two thresholds T_1 and T_2. The T_1 is calculated as

$$T_1 = \frac{I_{\max} + I_{\min}}{2}. \quad (1)$$

where $I_{\max}$ and $I_{\min}$ are maximum intensity and minimum intensity of the retinal image. The T_2 calculates as

$$T_2 = \frac{I_{brv} + I_{frv}}{2}. \quad (2)$$

where I_{brv} is the mean intensity of background and I_{frv} is the mean intensity of foreground of retinal blood vessel image. They are computed as

$$I_{brv} = \frac{\sum\limits_{R(x,y)<T_1} R(x,y)}{\sum\limits_{R(x,y)<T_1} M(x,y)}. \tag{3}$$

$$I_{frv} = \frac{\sum\limits_{R(x,y)\geq T_1} R(x,y)}{\sum\limits_{R(x,y)\geq T_1} M(x,y)}. \tag{4}$$

where $R(x,y)$ is the intensity of retinal images and $M(x,y)$ is the number of pixels and process of the count of each pixel explained as follow. The motivation of applying the double threshold is to get more large and small retinal blood vessels. The large vessels can easily be detected with T_1 but T_2 gives further opportunity to detect small retinal blood vessels. Other steps of getting well segment retinal vessels images are explained as below.

2. Partition the retinal image into three type regions: $A1$, containing all pixels with gray values below T_1; $A2$, containing all pixels with gray values between T_1 and T_2; and $A3$, containing all pixels with gray values above T_2. Thus $A1$ corresponds to pure background, retinal blood non-vessel region, $A2$ retinal blood vessels with gray-level intensities, and $A3$ retinal blood vessels with white intensities.
3. Visit each pixel in region $A2$. If the pixel has a neighbour in region $A1$, then reassign the pixel to region $A1$. Eight-connectedness is assumed, which means that a neighbour would be *North, North-East, East, South-East, South, South-West, West, or North-West* using cardinal directions.
4. Repeat step 3 until no pixel is reassigned.
5. Reassign any pixels left in region $A2$ to $A3$ to get final retinal blood vessels segment image as shown in Fig. 5.

Following the implementation of the above method, this paper presents three main contributions.

1. A retinal image suffered from pathologies making it difficult to observe the appropriate vessels. Enhancement of the color contrast is implemented to improve the color contrast of the image for subsequent pre-processing in order to obtain a well image contrast of the vessels without any illumination or background noise.
2. Two simple tactics are proposed to deal with the problem of uneven illumination and to reduce background noise. These two stages are based on the morphological operation and the ICA, and these tactics successfully give an individual RGB vessel image. Based on the contrast measurement, the green band image is selected for post-processing to obtain an image of the vessels. All these pre-processing steps are called image enhancement technique.

3. To implement the proposed image enhancement technique, we proposed post-processing steps that included vessel coherency and binarisation of the image to obtain the image of the vessels. We proposed a new binarisation to get well vessels image. It is validated with the comparison in Sect. 4.

2.3 Proposed Algorithm

The proposed contrast enhancement mechanism is associated with the segmentation model to observe its impact on vessels segmentation. The entire model contains different steps and can be developed as a compound algorithm, according to the following procedure:

1. A color enhancement tactic is applied on each channel of the retinal color image (I_{rgb}) to obtain the enhanced color image (I_{rgb}).
2. Morphological operations are applied on each channel of the enhanced color retinal image (I_{rgb}) to obtain uniform background images.
3. The three uniform channel images are fed separately into an ICA module to obtain well-enhanced images $I_{ICA_{rgb}}$.
4. The contrast of all the enhanced images is measured and the highest contrast channel is selected for the segmentation of the retinal blood vessels.
5. To combat noise while conserving small vessels, an anisotropic diffusion-filtered image with an adequate timing strategy is adopted on the final improved image to obtain a coherent image of the vessel (I_{coh}).
6. To maintain or detect more vessels, an array of multiscale Laplacian of Gaussian (LoG) filter is applied to obtain a better vessels enhanced image (I_e) for binarisation.
7. Finally, the technique of hysteresis or double threshold is adopted to obtain the final image of the binary vessels (I_{binary}).

3 Databases and Evaluating Parameters

Two databases are used in this research work to validate the performance of the proposed algorithm. The DRIVE database [18] contains 40 images. The 20 images of STARE database [14] are also used. Ten of the 20 images in the STARE database contain pathologies, which provide a good opportunity to analyse about the ability of the segmentation algorithm at different DR stages. The DRIVE and STARE databases are used because they provide the groundtruth images. In addition, almost researchers tested their methods from 1984 to 2017 used these databases. It is therefore fair to using them to compare the performance of our method with existing ones.

Three common criteria are used to evaluate the performance of the proposed method. Sensitivity$(Se) = tp/(tp + fn)$, Specificity$(Sp) = tn/(tn + fp)$, Accuracy$(AC) = (tp + tn)/(tp + fp + fn + tn)$. tp, tn, fp and fn represent the identification of the vessels and non-vessels pixels. The tp, tn, fp and fn are respectively abbreviated as true positives, true negatives, false positives and false negatives. Accuracy is the dominant parameter that gives us the information of the global classification of the vessels pixels.

4 Experimental Results Analysis and Discussion

4.1 Impact of Image Enhancement Technique on Segmentation Model

The performance of the segmentation models with and without associated image enhancement models is shown in Table 1 (Note: The segmentation model without image enhancement contained three steps. The third step: Process the green channel in the proposed segmentation model.) The segmentation model with image enhancement has a significant impact on the performance of retinal blood vessels segmentation accurately. The accuracy is increased from 92.5% to 94.8% on the DRIVE database and from 90.6% to 95.1% on the STARE database. We mainly focused on the sensitivity of the segmentation as it indicates a greater detection of tiny vessels. It is clearly observed that the sensitivity of the image-enhanced segmentation model has increased from 71.1% to 75.5% in the DRIVE database, from 71.9% to 78.4% in the STARE database. STARE gives more improved performance although it has more than 50% of pathological images. It is observed that the segmented image visualizes the blood vessels closer to the ground truth image vessels in the STARE and DRIVE databases, as shown in Fig. 6.

Table 1. Segmentation model performance analysis.

Image	DRIVE			*STARE*		
Methods/Measuring parameters	*Se*	*Sp*	*AC*	*Se*	*Sp*	*AC*
Segmentation without image enhancement	71.1%	94.4%	92.5%	71.9%	94.6%	90.6%
Segmentation with image enhancement	**74.5%**	**96.2%**	**94.8%**	**78.4%**	**97.6%**	**95.1%**

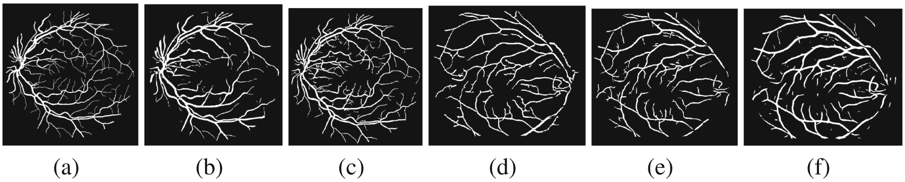


Fig. 6. Analysis of segment output image with and without image enhancement model. (a) Ground truth DRIVE image 01. (d) Ground truth STARE database image 0240. (b) Output image of DRIVE database without enhancement. (e) Output image of STARE database without enhancement. (c) Output image of DRIVE database with enhancement. (f) Output image of STARE database with enhancement.

4.2 Performance Comparison on Healthy and Unhealthy Images

The performance of the image segmentation model with and without image enhancement on healthy and unhealthy images of DRIVE databases is also observed. Table 2 shows the performance of the segmentation model with and without image enhancement on the healthy and unhealthy images of DRIVE and STARE databases. The performance of the segmentation model according to the image enhancement technique on unhealthy images gives a high sensitivity of 74.3 % on the DRIVE database and 78.9% on the STARE database, against 70.9% on the DRIVE database and 71.7% on the STARE image enhancement database.

Table 2. Segmentation model performance analysis.

Image	Without image enactment			With image enactment		
Methods/Measuring parameters	*Se*	*Sp*	*AC*	*Se*	*Sp*	*AC*
DRIVE databae healthy images	71.2%	94.9%	92.9%	74.7%	95.9%	94.9%
STARE databae healthy images	72.1%	95.0%	90.9%	77.9%	98.1%	95.2%
DRIVE databae unhealthy images	70.9%	93.8%	91.9%	74.3%	96.5%	94.8%
STARE databae unhealthy images	71.7%	94.3%	90.2%	78.9%	97.1%	94.9%

Table 3. Performance analysis of segmentation model

Database	DRIVE			*STARE*		
Methods/Measuring parameters	*Se*	*Sp*	*AC*	*Se*	*Sp*	*AC*
Supervised methods						
Soares et al. [14]	-	-	94.6%	-	-	94.8%
Lupascu et al. [7]	72%	-	**95.9%**	-	-	-
You et al. [19]	74.1%	97.5%	94.3%	72.6%	97.5%	94.9%
Liskowski and Krawiec [6]	-	-	94.9%	-	-	94.9%
Unsupervised methods						
Mendonca et al. [9]	73.4%	97.6%	94.5%	69.9%	97.3%	94.4%
Matinez-Perez et al. [8]	72.4%	96.5%	93.4%	75%	95.6%	94.1%
Al-Diri et al. [1]	72.8%	95.5%	-	75.2%	96.8%	-
Palomera-Perez et al. [12]	66%	96.1%	92.2%	77.9%	94%	92.4%
Nguyen et al. [11]	-	-	94%	-	-	93.2%
Yin et al. [21]	-	-	94.7%	-	-	-
Zhao et al. [22]	71.6%	**97.8%**	94.4%	77.6%	95.4%	94.3%
Proposed method	**74.5%**	**96.2%**	**94.8%**	**78.4%**	**97.6%**	**95.1%**

4.3 Comparison with Other Methods

In order to prove the effectiveness of the proposed method, the performance of the proposed segmentation method is compared to the other existing vessels detection models: DRIVE and STARE, as shown in Table 3. Compared with other supervised methods, the proposed method gives better accuracy, and the sensitivity of the proposed method is $Se = 75.9\%$ in the DRIVE database higher than existing methods. This proposed method gives comparable performances with [6] but the sensitivity of the proposed method in the two databases is slightly higher than [6]. In unsupervised methods, the sensitivity of the proposed method is superior to that of existing methods. Note that the specificity of Mendonca et al. [9] and Zhao et al. [22] is slightly higher than our method proposed in the DRIVE database. However, our proposed method outperforms [9] and [22] in terms of the sensitivity and the accuracy on DRIVE and STARE databases.

5 Conclusion

Automated retinal vessels segmentation is a difficult task because retinal fundus images are often noisy, include artifacts and low contrast throughout the image. The noise and low contrast between the blood vessels and the background make it difficult to accurately extract blood vessels. In this research, the implementation of contrast enhancement technique based on the morphological operation, ICA to improve blood vessels, is used to obtain a good contrast image. The proposed image enhancement method has been validated on the DRIVE and STARE databases. Experimental results have shown that the proposed image enhancement method provides enhanced contrast with its cooperative segmentation model, increases the accuracy from 92% to 95% and the sensitivity from 71% to 75%, while providing comparable performance against other existing methods. It is confirmed that the proposed image enhancement method has improved the segmentation of smaller vessels and reduced the noise. Future work will revolve around this method of image enhancement, which can be used before segmentation with improved binarisation of segmentation, so that better blood vessel extraction can be achieved with better performance of the segmentation model.

References

1. Al-Diri, B., Hunter, A., Steel, D.: An active contour model for segmenting and measuring retinal vessels. IEEE Trans. Med. Imaging **28**(9), 1488–1497 (2009)
2. Cuadros, J., Martin, C.: Diabetic retinopathy screening practice guide. In: Yogesan, K., Goldschmidt, L., Cuadros, J. (eds.) Digital Teleretinal Screening, pp. 11–30. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-25810-7_2
3. Faust, O., Acharya, U.R., Ng, E.Y.K., Ng, K.H., Suri, J.S.: Algorithms for the automated detection of diabetic retinopathy using digital fundus images: a review. J. Med. Syst. **36**(1), 145–157 (2012)
4. Gilchrist, J.: Computer processing of ocular photographs-a review. Ophthalmic Physiol. Opt. **7**, 379–386 (1987)
5. Gottschlich, C., Schonlieb, C.B.: Oriented diffusion filtering for enhancing low-quality fingerprint images. IET Biom. **1**, 105–113 (2012)
6. Liskowski, P., Krawiec, K.: Segmenting retinal blood vessels with deep neural networks. IEEE Trans. Med. Imaging **35**(11), 2369–2380 (2016)

7. Lupaşcu, C.A., Tegolo, D., Trucco, E.: FABC: retinal vessel segmentation using AdaBoost. IEEE Trans. Inf. Technol. Biomed. **14**(5), 1267–1274 (2010)
8. Martinez-Perez, M.E., Hughes, A.D., Thom, S.A.: Segmentation of blood vessels from red-free and fluorescein retinal images. J. Med. Image Anal. **11**(1), 47–61 (2007)
9. Mendonca, A.M., Campilho, A.: Segmentation of retinal blood vessels by combining the detection of centerlines and morphological reconstruction. IEEE Trans. Med. Imaging **25**, 1200–1213 (2006)
10. Menotti, D., de Albuquerque Araújo, A., Pappa, G.L., Najman, L., Facon. J.: Contrast enhancement in digital imaging using histogram equalization. In: Computer Science Universite Paris Est Universidade federal de Minas Gerais (Bresil), vol. 1, pp. 1–10 (2008)
11. Nguyen, U.T., Bhuiyan, A., Park, L.A., Ramamohanarao, K.: An effective retinal blood vessel segmentation method using multi-scale line detection. Pattern Recognit. **46**, 703–715 (2013)
12. Palomera-Perez, M.A., Martinez-Perez, M.E., Benitez-Perez, H., Ortega-Arjona, J.L.: Parallel multiscale feature extraction and region growing: application in retinal blood vessel detection. IEEE Trans. Inf. Technol. Biomed. **14**(2), 500–506 (2010)
13. Shen, Y., Shu-zhen, C., Bing, Z.: An improved doublethreshold method based on gradient histogram. Wuhan Univ. J. Nat. Sci. **9**(4), 473–476 (2004)
14. Soares, J.V.B., Leandro, J.J.G., Cesar, R.M., Jelinek, J.H.F., Cree, M.J.: Retinal vessel segmentation using the 2D gabor wavelet and supervised classification. IEEE Trans. Med. Imaging **25**(9), 1214–1222 (2006)
15. Soomro, T.A., Khan, T.M., Khan, M.A.U., Gao, J., Paul, M., Zheng, L.: Impact of ICA-based image enhancement technique on retinal blood vessels segmentation. IEEE Access **6**, 3524–3538 (2018)
16. Soomro, T.A., Gao, J.: Non-invasive contrast normalisation and denosing technique for the retinal fundus image. Ann. Data Sci. **1**, 1–15 (2016)
17. Soomro, T.A., Gao, J., Khan, M.A.U., Khan, T.M., Paul., M.: Role of image contrast enhancement technique for ophthalmologist as diagnostic tool for diabetic retinopathy. In: International Conference on Digital Image Computing: Techniques and Applications (DICTA), pp. 1–8. November 2016. https://doi.org/10.1109/DICTA.2016.7797078
18. Staal, J., Abràmoff, M.D., Niemeijer, M., Viergever, M.A., van Ginneken, B.: Ridge based vessel segmentation in color images of the retina. IEEE Trans. Med. Imaging **23**, 501–509 (2004)
19. You, X., Peng, Q., Yuan, Y., Cheung, Y.-M., Lei, J.: Segmentation of retinal blood vessels using the radial projection and semi-supervised approach. Pattern Recognit. **44**, 10–11 (2011)
20. Yannuzzi, L.A., et al.: Fluorescein angiography complication survey. Ophthalmology **93**(5), 611–617 (1986)
21. Yin, X., Ng, B.W.H., He, J., Zhang, Y., Abbott, D.: Accurate image analysis of the retina using hessian matrix and binarisation of thresholded entropy with application of texture mapping. PLOS ONE **9**(4), 1–17 (2014)
22. Zhao, Y., Rada, L., Chen, K., Harding, S.P., Zheng, Y.: Automated vessel segmentation using infinite perimeter active contour model with hybrid region information with application to retinal images. IEEE Trans. Med. Imaging **34**(9), 1797–1807 (2015)

Sequential Deep Learning for Action Recognition with Synthetic Multi-view Data from Depth Maps

Bin Liang[1](✉), Lihong Zheng[2], and Xinying Li[3]

[1] University of Technology Sydney, Sydney, Australia
bin.liang@uts.edu.au
[2] Charles Sturt University, Wagga Wagga, Australia
lzheng@csu.edu.au
[3] Changchun University of Technology, Changchun, China
cgydxlxy@126.com

Abstract. Recurrent neural network (RNN) has proven successful recently in action recognition. However, depth sequences are of high dimensionality and contain rich human dynamics, which makes traditional RNNs difficult to capture complex action information. This paper addresses the problem of human action recognition from sequences of depth maps using sequential deep learning. The proposed method first synthesizes multi-view depth sequences by rotating 3D point clouds from depth maps. Each depth sequence is then split into short-term temporal segments. For each segment, a multi-view depth motion template (MVDMT), which compresses the segment to a motion template, is constructed for short-term multi-view action representation. The MVDMT effectively characterizes the multi-view appearance and motion patterns within a short-term duration. Convolutional Neural Network (CNN) models are leveraged to extract features from MVDMT, and a CNN-RNN network is subsequently employed to learn an effective representation for sequential patterns of the multi-view depth sequence. The proposed multi-view sequential deep learning framework can simultaneously capture spatial-temporal appearance and motion features in the depth sequence. The proposed method has been evaluated on the MSR Action3D and MSR Action Pairs datasets, achieving promising results compared with the state-of-the-art methods based on depth data.

Keywords: Action recognition · Sequential deep learning · Depth map · Multi-view data

1 Introduction

The recognition of human actions has raised considerable interest and has become a very active research area over the last two decades. Starting from either of RGB data, depth maps, skeletal joints or some combination of these data sources, many action recognition methods have been developed and applied

R. Islam et al. (Eds.): AusDM 2018, CCIS 996, pp. 360–371, 2019.
https://doi.org/10.1007/978-981-13-6661-1_28

in various practical applications, such as human-computer interaction, motion-sensing gaming, video surveillance, rehabilitation and smart homes [1]. Human actions are performed in real 3D environments; however, traditional cameras only capture the 2D projection of a scene. As a result, projection of actions depends on the viewpoint. As visual appearances vary from different perspectives, a single view does not usually provide information sufficiently comprehensive to describe actions. Compared with single-view representations, multiple views encode more essential information for human action recognition. The use of multiple views in human action recognition has been shown to improve recognition accuracy [2].

With the advent and popularity of depth cameras such as Kinect, action recognition using depth sensors has prevailed in the research community. In a depth map, pixels encode the distance information of a scene rather than a measure of the intensity of colour in RGB data. In order to leverage depth information and reduce computational cost, many approaches have been proposed to transform the problem from 3D to 2D by projecting a depth map onto predefined planes [3–6]. Typically, the depth map of each frame is projected onto predefined 2D planes, and then recognition is performed on the projected 2D planes. In this way, a multi-view action recognition task is generalized, where each view refers to the corresponding projected plane.

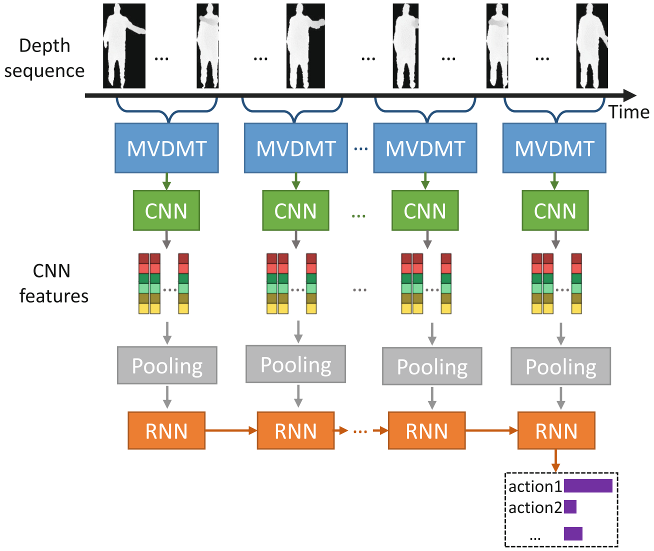


Fig. 1. Framework overview of the proposed method.

Multi-view information generated from a depth camera is very beneficial for action recognition, but how to model human actions using multi-view information is a non-trivial task. Besides, representing the temporal structural information in the point of multi-view is also quite challenging. To address these issues, this paper adopts a method to obtain an arbitrary number of synthetic multiple views from depth maps, providing additional information of multiple views for better action recognition [7]. A multi-view depth motion template (MVDMT) is employed to characterize the short-term motion and shape patterns of the synthetic multi-view data of actions, leading to an effective action representation. One of the advantages of MVDMT is that we can leverage the existing Convolutional Neural Network (CNN) models that have been pre-trained on massive image datasets, such as ImageNet [8]. In addition, a CNN-RNN network is employed for sequential deep learning. RNN is used to learn the dynamic CNN features from the MVDMT for the effective classification. Figure 1 shows the framework of the proposed method. For each MVDMT, the framework first extracts CNN features from each view, and then the features are fused through a pooling operation to produce the final temporal features. The temporal features finally are fed into the RNN to compute classification scores of actions. One major contribution of our work is to propose a sequential deep learning framework for depth action recognition using synthetic multi-view data. This method inherits the advantages of multiple views and CNN-RNN networks.

The reminder of this paper is organized as follows. We first review related work on depth action recognition in Sect. 2. Section 3 provides the details of proposed framework. The experimental validation is presented in Sect. 4. At last, Sect. 5 gives a conclusion of the paper.

2 Related Work

In recent years, human action recognition based on depth data has attracted much attention. This is largely because the additional depth channel is insensitive to illumination changes and includes rich 3D structural information of the scene. Compared to single-view data, multi-view data can provide complementary information to overcome the limitations in single-view systems, such as the change of observation view and potential occlusions. Therefore, many approaches have been proposed to synthesize multiple views from depth maps. In [3] and [4], human actions were recognized using depth motion maps (DMMs), which were developed to capture the aggregated temporal motion energies. In [9], a new framework based on pyramid motion history templates (PMHT) was proposed to perform human action recognition on depth sequences. However, more views might be required to ensure the sufficient description of more complex actions.

Along with the development in artificial intelligence, deep learning techniques have gained remarkable success in computer vision. Several recent works using deep learning for action recognition in temporal sequences have investigated the question of how to exploit the temporal information. To date, CNN models for video processing have successfully considered learning of 3D spatio-temporal filters over raw video sequence data [10], and learning of frame-to-frame representations [11]. In addition, Recurrent Neural Networks (RNNs) parse the video frames sequentially and encode the frame-level information in their memory. Furthermore, the Long Short Term Memory networks (LSTMs) [12], a special kind of RNNs, was introduced to model long-term actions. Researchers have made some progress in modelling spatio-temporal information by connecting CNNs and RNNs [13].

One typical challenge in deep learning based action recognition is how a depth sequence could be effectively represented and fed to deep neural networks for recognition. In this paper, we propose to encode the depth video into MVDMT using synthetic multi-view data, and exploit CNN-RNN networks to learn an effective representation for sequential patterns. This enhances the capability of deep learning for action recognition from depth maps.

3 The Proposed Framework

3.1 Multiple Views from Depth Maps

In order to make the best use of the additional motion information from depth maps for the generation of multiple views, each depth frame is firstly converted to point clouds in 3D space, and then projected onto multiple planes, representing multiple views. Figure 2(a) illustrates how to convert a depth value in a depth map to the corresponding 3D point. The coordinate (x_w, y_w, z_w) of the 3D point can be calculated by:

$$x_w = \frac{(x_p - c_x) \cdot z_w}{f_x}, \; y_w = \frac{(y_p - c_y) \cdot z_w}{f_y} \tag{1}$$

where (x_p, y_p) and z_w denote screen coordinates and the depth value respectively, (c_x, c_y) denotes the center of the depth map, and f_x and f_y are the focal lengths of Kinect camera. Here for Kinect, $f_x = f_y = 580$ [14].

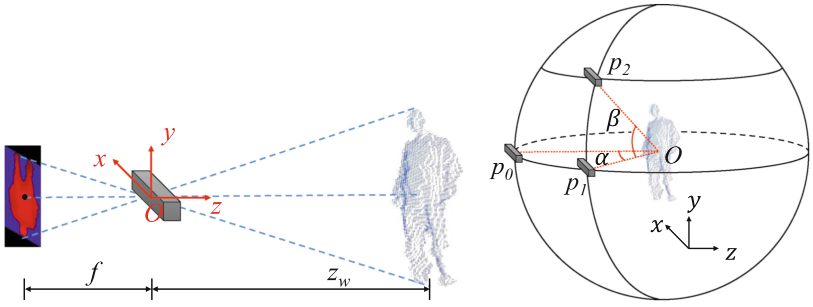
Then, the projections onto 2D planes can be obtained by rotating the calculated 3D point. The rotation of the 3D points can be performed equivalently by assuming that a virtual camera moves around the subject from different viewpoints as illustrated in Fig. 2(b). If the virtual camera moves from position p_0 to p_2, the process can be decomposed into the following two steps: firstly it moves from p_0 to p_1 by an angle α about the y axis, and then it moves from p_1 to p_2 by an angle β about the x axis. The corresponding coordinate (x_{rw}, y_{rw}, z_{rw}) of point clouds after rotation can be computed through multiplication by the transformation matrices $\mathbf{R}_y(\alpha)$ and $\mathbf{R}_x(\beta)$:

$$[x_{rw}, y_{rw}, z_{rw}, 1]^T = \mathbf{R}_x(\beta)\mathbf{R}_y(\alpha)[x_w, y_w, z_w, 1]^T \tag{2}$$

where $\mathbf{R}_y(\alpha)$ denotes the rotation about the y axis, and $\mathbf{R}_x(\beta)$ denotes the rotation about the x axis. The transformation matrices can be expressed as

$$\mathbf{R}_y(\alpha) = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \; \mathbf{R}_x(\beta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\beta & -\sin\beta & 0 \\ 0 & \sin\beta & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Each pixel value of the projection image is the number of 3D points with the same projected coordinates. Therefore, an arbitrary number of views can be obtained by rotating the virtual camera.



(a) 3D point clouds calculation. (b) 3D point clouds rotation.

Fig. 2. Illustration of synthetic multiple-view generation.

A depth camera captures 2.5D information because only the 3D structure of the point visible to the sensor is contained in the map; nothing is known about the other side of the object or scene. The rotation has to be within a range such that the projections still provide sufficient spatial information about the actions. In other words, both the angles of α and β have to be limited to a certain range. Figure 3 shows an example of multiple views from a depth map. It can be seen from the figure that each view captures the shape of the body from that viewpoint.

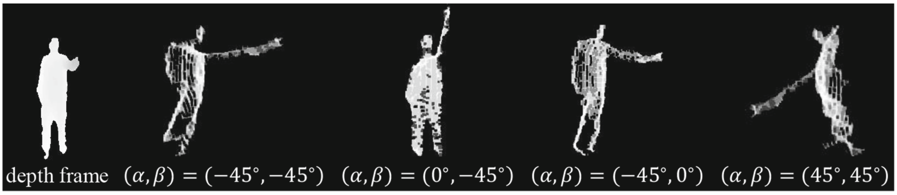

depth frame
$(\alpha,\beta) = (-45°,-45°)$
$(\alpha,\beta) = (0°,-45°)$
$(\alpha,\beta) = (-45°,0°)$
$(\alpha,\beta) = (45°,45°)$

Fig. 3. Multiple views from a depth map.

3.2 Construction of MVDMT

Given a multi-view action sequence, N short-term segments can be obtained. For each short-term segment, a multi-view depth motion template (MVDMT) is then adopted for the short-term action representation. Let P_v denote the pixel value on the projection image on the plane v, so the t-th depth frame $I_v(x_w, y_w, z_w, t)$ can be denoted as a set of V projected grey images: $\{I_v(P_v, t)\}_{v=1}^{V}$, where V is the number of views. These projected images are then converted to greyscale by normalization. In this way, multiple 2D planes are generated from each depth frame according to the selected V views.vadjust

With the purpose of finding motion regions, $D_v(P_v, t) = |I_v(P_v, t) - I_v(P_v, t-1)|$ is defined to calculate the differences of the t-th ($t \in [t_1, t_2]$) frame and its previous frame, where t_1 and t_2 are the start and finish time of the corresponding short-term segment. A MVDMT h_v at time t can be obtained in a recursive manner

$$\mathrm{h}_v(P_v, t) = \begin{cases} t_2 - t_1 + 1 & \text{if } D_v(P_v, t) > \xi \\ \max(0, \mathrm{h}_v(P_v, t-1) - \sigma) & \text{otherwise} \end{cases} \tag{4}$$

where ξ and σ are the values of threshold for motion detection and decay parameter, respectively.

Pseudo colouring [15] is a typical process used as a means of extracting more information from grey images. Motivated by this, it is proposed to further code a MVDMT into a pseudo-colour image, which enables to enhance the texture in the MVDMT corresponding to the short-term motion patterns of actions. Given an intensity value I, the power rainbow transform [15] encodes the value into a normalized colour (R', G', B') as follows:

$$\begin{aligned} R' &= \left[\left(1 + \cos(\tfrac{4\pi}{3\times 255} I)\right)\Big/2\right]^2 \\ G' &= \left[\left(1 + \cos(\tfrac{4\pi}{3\times 255} I - \tfrac{2\pi}{3})\right)\Big/2\right]^2 \\ B' &= \left[\left(1 + \cos(\tfrac{4\pi}{3\times 255} I - \tfrac{4\pi}{3})\right)\Big/2\right]^2 \end{aligned} \tag{5}$$

where R', G' and B' are the normalized RGB values from the power rainbow transform. An example of a MVDMT with pseudocoloring is shown in Fig. 4.

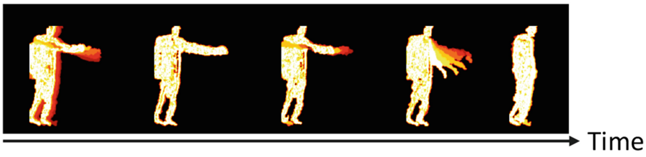


Fig. 4. MVDMTs with pseudocoloring.

The MVDMT can well characterize the motion and shape information of the short-term 3D action in the temporal direction. It is also a compact and informative representation of human actions. Another advantages of MVDMT is that we can leverage the existing CNN models that have been pre-trained on massive image datasets, such as ImageNet [8].

3.3 Sequential Deep Learning

In this subsection, we describe how to perform the sequential deep learning on a sequence of MVDMTs using deep neural network. Specifically, we extract features from MVDMT using CNNs. Then, we leverage the CNN features at all time steps to automatically learn spatial-temporal information through RNN.

First, we extract features from multiple views of the t-th MVDMTs in a depth sequence. To achieve it, we feed multiple views from MVDMT into CNNs architecture. In principle, any kind of CNNs can be adopted for feature extraction from MVDMT. For the t-th MVDMT, we extract a set of CNN feature vectors, $\mathrm{X}_t = \left\{\mathbf{x}_t^1, \ldots \mathbf{x}_t^V\right\}$, from the fully-connected layer of CNNs, where V is the number of different views. After that, a pooling scheme is employed to generate a final feature vector $\mathbf{x}_t$. As a result, each depth sequence can be represented as a sequence of pooled multi-view CNN features, i.e., $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, where N is the number of MVDMTs.

After extracting the sequences of deep features from multi-view CNNs, it is natural to use RNN to encode temporal structure of these sequentially-ordered features. The t-th feature vector $\mathbf{x}_t$ can be used as the t-th input to train RNN to learn the dynamics of the sequence. As a special kind of RNNs, LSTMs are widely used to model temporal dependency and have been successfully applied to natural language processing [16], speech recognition [17], image and video description [18]. Therefore, an LSTM with a linear layer that computes classification scores based on the CNN features of the current MVDMT and the hidden states and memory of the LSTM from the previous MVDMT. LSTMs provide a solution by incorporating memory units that allow the network to learn when to forget previous hidden states and when to update hidden states given new information. In our CNN-RNN network, a simplified LSTM model is adopted as in [12,17,19], which is illustrated in Fig. 5

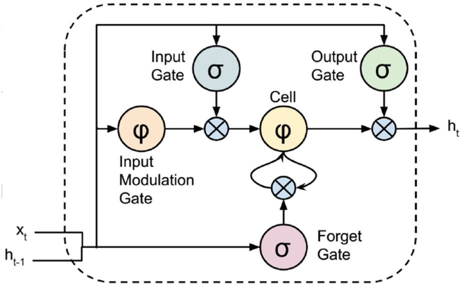


Fig. 5. The LSTM unit used in this paper.

In detail, given a feature vector $\mathbf{x_t}$ generated from the action sequence, an LSTM unit computes the output h_t based on the following equations recursively:

$$\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}_{xi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{b}_i) \\
\mathbf{f}_t &= \sigma(\mathbf{W}_{xf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{b}_f) \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \phi(\mathbf{W}_{xc}\mathbf{x}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1} + \mathbf{b}_c) \\
\mathbf{o}_t &= \sigma(\mathbf{W}_{xo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{b}_o) \\
\mathbf{h}_t &= \mathbf{o}_t \odot \phi(\mathbf{c}_t)
\end{aligned} \tag{6}$$

where $\odot$ denotes the element-wise multiplication, $\mathbf{x}_t$ and $\mathbf{h}_t$ are the input and hidden states for this LSTM unit for the t-th MVDMT, $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{c}_t$ and $\mathbf{o}_t$ are the states of the input gate, forget gate, memory cell and output gate, respectively.

$\mathbf{W}_{ab}$ is the weight matrix between gate a and gate b, $\mathbf{b}_a$ is the bias term of gate a, σ is the sigmoid nonlinearity, defined as $\sigma(x) = (1 + e^{-x})^{-1}$, which squashes real-valued inputs to (0, 1) range, and ϕ is the hyperbolic tangent nonlinearity, defined as $\phi(x) = (e^x - e^{-x})/(e^x + e^{-x}) = 2\sigma(2x) - 1$. In addition to a hidden unit $\mathbf{h}_t$, the LSTM includes an input gate $\mathbf{i}_t$, forget gate $\mathbf{f}_t$, output gate $\mathbf{o}_t$, input modulation gate $\mathbf{g}_t$, and memory cell $\mathbf{c}_t$. The memory cell unit $\mathbf{c}_t$ is a summation of two things: the previous memory cell unit $\mathbf{c}_{t-1}$ which is modulated by $\mathbf{f}_t$, and $\mathbf{g}_t$, a function of the current input and previous hidden state, modulated by the input gate $\mathbf{i}_t$. Because $\mathbf{i}_t$ and $\mathbf{f}_t$ are sigmoidal, their values lie within the range (0, 1), and $\mathbf{i}_t$and $\mathbf{f}_t$ can be thought of as knobs that the LSTM learns to selectively forget its previous memory or consider its current input. Likewise, the output gate $\mathbf{o}_t$ learns how much of the memory cell to transfer to the hidden state.

To model temporal dependency, we feed the outputs of CNN into LSTMs. The illustration of the model is shown in Fig. 1. The whole network is composed of 4 parts: depth sequence, MVDMTs, CNNs, and RNNs. The CNN part contains convolutional layers, pooling layers, and fully-connected layers. The RNN part is a single layer RNN network which consists of one LSTM layers. With the CNN-RNN architecture, we are able to fuse spatial and temporal features, and get a prediction for each MVDMT based on its previous ones.

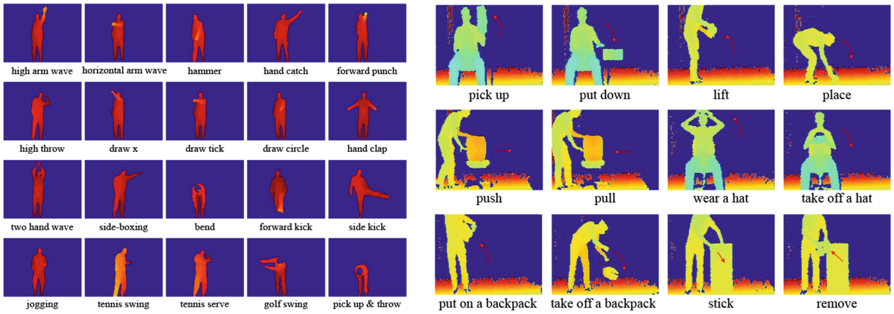
4 Experimental Results

To evaluate action recognition performance of the proposed framework, experiments are conducted on two public datasets: MSR Action3D [20] and MSR Action Paris [21] datasets. Our method is compared to the existing depth-based approaches. For fair comparison, the methods based on skeleton data are not included in our experiments. The proposed framework consistently achieves comparable results with the state-of-the-art methods.

4.1 Datasets

The MSR Action3D dataset [20] is an action dataset of depth sequences captured by a depth sensor. It contains 567 depth map sequences. There are 20 action types, which are chosen in the context of gaming and cover a variety of movements related to arms, legs, torso and their combinations. Sample frames of the action sequences are shown in Fig. 6(a). The background in this dataset is preprocessed to clear the discontinuities induced by undefined depth regions. Nevertheless, this dataset remains challenging because many of the actions are highly similar to each other.

The MSR Action Pairs dataset [21] selects pairs of activities, so that within each pair the motion and the shape cues are similar, although their correlations are different. Therefore, this dataset can be used to evaluate the performance of action representation in the context of capturing the prominent cues in the sequence. Six pairs of actions are collected, and the sample frames are shown in Fig. 6(b). Each action in these datasets is performed by 10 different subjects



(a) MSR Action3D dataset.

(b) MSR Action Pairs datasets.

Fig. 6. Sample frames from used datasets.

for 2 or 3 times. Following the original experimental settings, 5 subjects in each action class are used for training, with the rest used for testing; known as a cross-subject test. This presents a challenge, as the subjects are free to perform actions in their own manner or style. We train the models on the training set and report the recognition accuracy on the test set over all classes.

4.2 Implementation Details

In our experiments, for MVDMT generation in Eq. 4, the value of threshold ξ is set at 50, and the value of decay parameter σ is set at 1. With respect to the number of MVDMTs N and the number of views V, we run experiments with different settings. The overall accuracies show that best performance can be achieved with $N = 5$ and $V = 9$. The average pooling and max pooling schemes are used to generate the final feature vector for each MVDMT.

We choose widely-used CNN architectures as the feature extraction module. The pre-trained 16-layer network (VGG-16 model) used by the VGG team is chosen. For each view of MVDMT, we extract the fully-connected feature vector with dimensionality of 4096 from the last layer of VGG-16 model. The dimension of all hidden variables in LSTM of our model is 512. The network weights were learnt using the mini-batch stochastic gradient descent with momentum (set to 0.9). The batch size for training RNN is 128, and the learning rate started from 10^{-3}. The training procedure stops at 700 iterations. We implement our network using Keras [22].

4.3 Comparison with Other Methods

We compare our method with the state-of-the-art methods on MSR Action3D [20] and MSR Action Pairs [21] datasets. The comparison of recognition accuracy results is shown in Table 1. On MSR Action 3D dataset, the bag of 3D points [20] encodes the 3D body shape information using depth data, and the recognition accuracy is only 74.70%. This is probably due to the small number of subjects and also the significant variations of the same action performed by different subjects. STOP [23] leverages the spatial and temporal contextual information while allowing for intra-class variations, and achieves an accuracy of 84.80%. [24] proposes a 3D depth features called ROP to deal with the noise and occlusion problems and obtains an accuracy of 86.20%. The recently proposed depth-based action recognition methods, including HON4D [21], DSTIP [25], 3DMTM-PHOG [5], DMM-HOG [3], Range-Sample Feature [26], and SLCL [7] achieve accuracy of 88.89%, 89.30%, 90.70%, 91.70%, 95.62%, and 95.77%, respectively. The proposed framework obtains the best performance (97.28%) among all reported approaches on MSR Action3D dataset. Furthermore, the proposed framework achieves an accuracy of 97.14% on MSR Action Pairs dataset that is ranked first among recent depth-based methods. This dataset is collected to investigate how the temporal order affects action recognition. It is therefore crucial to capture the spatio-temporal orders to distinguish the actions with similar motion and cues. Different to MSR Action3D dataset, the background of the actions in MSR Action Pairs dataset is not clear which is more challenging for action recognition. Thus, some compared work only presents the results on MSR Action3D dataset. Furthermore, two pooling schemes are analysed in our experiments, i.e., max pooling and average pooling. It is observed that better results can be achieved with average pooling. In our approach, the spatial information is encoded by MVDMT generated from the synthetic multiple views. In addition, the spatio-temporal orders are embedded in sequential deep learning. Therefore, our framework achieves the state-of-the-art result than the depth-based methods that have been performed on these datasets.

Table 1. Recognition accuracy (%) comparison on used datasets

Method	MSR Action3D	MSR Action Pairs
Bag of 3D points [20]	74.70	-
STOP [23]	84.80	-
ROP [24]	86.20	-
HON4D [21]	88.89	96.67
DSTIP [25]	89.30	-
3DMTM-PHOG [5]	90.70	-
DMM-HOG [3]	91.70	66.11
Range-Sample [26]	95.62	-
SLCL [7]	95.77	95.43
Ours (max pooling)	96.68	96.00
Ours (average pooling)	97.28	97.14

5 Conclusions

In this paper, a sequential deep learning framework has been proposed for action recognition using synthetic multi-view data from depth maps. Multi-view action sequences are firstly represented by the multi-view depth motion template (MVDMT), characterizing the multi-view motion and shape patterns within a short-term duration. A CNN-RNN network is then employed to learn an effective representation for sequential patterns. To evaluate the framework, a series of experiments is conducted on two public depth action datasets. The experimental results demonstrate that the proposed method achieves state-of-the-art performances compared with depth-based methods on these datasets.

References

1. Jaimes, A., Sebe, N.: Multimodal human-computer interaction: a survey. Comput. Vis. Image Underst. **108**(1), 116–134 (2007)
2. Atrey, P.K., Hossain, M.A., El Saddik, A., Kankanhalli, M.S.: Multimodal fusion for multimedia analysis: a survey. Multimedia Syst. **16**(6), 345–379 (2010)
3. Yang, X., Zhang, C., Tian, Y.L.: Recognizing actions using depth motion maps-based histograms of oriented gradients. In: Proceedings of the 20th ACM International Conference on Multimedia, pp. 1057–1060. ACM (2012)
4. Chen, C., Liu, K., Kehtarnavaz, N.: Real-time human action recognition based on depth motion maps. J. Real Time Image Proc. **12**(1), 153–163 (2016). https://doi.org/10.1007/s11554-013-0370-1
5. Liang, B., Zheng, L.: 3D motion trail model based pyramid histograms of oriented gradient for action recognition. In: 2014 22nd International Conference on Pattern Recognition (ICPR), pp. 1952–1957. IEEE (2014)
6. Jetley, S., Cuzzolin, F.: 3D activity recognition using motion history and binary shape templates. In: Jawahar, C.V., Shan, S. (eds.) ACCV 2014, Part I. LNCS, vol. 9008, pp. 129–144. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16628-5_10
7. Liang, B., Zheng, L.: Specificity and latent correlation learning for action recognition using synthetic multi-view data from depth maps. IEEE Trans. Image Process. **26**(12), 5560–5574 (2017)
8. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Li, F.-F.: ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009), pp. 248–255. IEEE (2009)
9. Liang, B., Zheng, L.: Spatio-temporal pyramid cuboid matching for action recognition using depth maps. In: 2015 IEEE International Conference on Image Processing (ICIP), pp. 2070–2074. IEEE (2015)
10. Ji, S., Wei, X., Yang, M., Kai, Y.: 3D convolutional neural networks for human action recognition. IEEE Trans. Pattern Anal. Mach. Intell. **35**(1), 221–231 (2013)
11. Karpathy, A., George, T., Shetty, S., Leung, T., Sukthankar, R., Li, F.-F.: Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1725–1732 (2014)
12. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)

13. Wu, Z., Wang, X., Jiang, Y.-G., Hao, Y., Xue, X.: Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. In: Proceedings of the 23rd ACM International Conference on Multimedia, pp. 461–470. ACM (2015)
14. Smisek, J., Jancosek, M., Pajdla, T.: 3D with kinect. In: Fossati, A., Gall, J., Grabner, H., Ren, X., Konolige, K. (eds.) Consumer Depth Cameras for Computer Vision. Advances in Computer Vision and Pattern Recognition, pp. 3–25. Springer, London (2013). https://doi.org/10.1007/978-1-4471-4640-7_1
15. Abidi, B.R., Zheng, Y., Gribok, A.V., Abidi, M.A.: Improving weapon detection in single energy X-ray images through pseudocoloring. IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.) **36**(6), 784–796 (2006)
16. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems, pp. 3104–3112 (2014)
17. Graves, A., Mohamed, A.-R., Hinton, G.: Speech recognition with deep recurrent neural networks. In: 2013 IEEE International Conference on Speech and Signal Processing (ICASSP), pp. 6645–6649. IEEE (2013)
18. Xu, J., Yao, T., Zhang, Y., Mei, T.: Learning multimodal attention LSTM networks for video captioning. In: Proceedings of the 2017 ACM on Multimedia Conference, pp. 537–545. ACM (2017)
19. Shi, Y., Tian, Y., Wang, Y., Huang, T.: Sequential deep trajectory descriptor for action recognition with three-stream CNN. IEEE Trans. Multimedia **19**, 1510–1520 (2017)
20. Li, W., Zhang, Z., Liu, Z.: Action recognition based on a bag of 3D points. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 9–14. IEEE (2010)
21. Oreifej, O., Liu, Z.: HON4D: histogram of oriented 4D normals for activity recognition from depth sequences. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 716–723. IEEE (2013)
22. Chollet, F.: Keras (2015). https://github.com/fchollet/keras
23. Vieira, A.W., Nascimento, E.R., Oliveira, G.L., Liu, Z., Campos, M.F.M.: STOP: space-time occupancy patterns for 3D action recognition from depth map sequences. In: Alvarez, L., Mejail, M., Gomez, L., Jacobo, J. (eds.) CIARP 2012. LNCS, vol. 7441, pp. 252–259. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33275-3_31
24. Wang, J., Liu, Z., Chorowski, J., Chen, Z., Wu, Y.: Robust 3D action recognition with random occupancy patterns. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part II. LNCS, pp. 872–885. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33709-3_62
25. Xia, L., Aggarwal, J.K.: Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2834–2841. IEEE (2013)
26. Lu, C., Jia, J., Tang, C.-K.: Range-sample depth feature for action recognition. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 772–779. IEEE (2014)

Provenance Analysis for Instagram Photos

Yijun Quan[1(✉)], Xufeng Lin[2], and Chang-Tsun Li[1,2]

[1] University of Warwick, Coventry, UK
Y.Quan@warwick.ac.uk
[2] Charles Sturt University, Wagga Wagga, Australia

Abstract. As a feasible device fingerprint, sensor pattern noise (SPN) has been proven to be effective in the provenance analysis of digital images. However, with the rise of social media, millions of images are being uploaded to and shared through social media sites every day. An image downloaded from social networks may have gone through a series of unknown image manipulations. Consequently, the trustworthiness of SPN has been challenged in the provenance analysis of the images downloaded from social media platforms. In this paper, we intend to investigate the effects of the pre-defined Instagram images filters on the SPN-based image provenance analysis. We identify two groups of filters that affect the SPN in quite different ways, with Group I consisting of the filters that severely attenuate the SPN and Group II consisting of the filters that well preserve the SPN in the images. We further propose a CNN-based classifier to perform filter-oriented image categorization, aiming to exclude the images manipulated by the filters in Group I and thus improve the reliability of the SPN-based provenance analysis. The results on about $20,000$ images and 18 filters are very promising, with an accuracy higher than 96% in differentiating the filters in Group I and Group II.

Keywords: Digital image forensics · Sensor pattern noise · Social media · Provenance analysis

1 Introduction

The provenance of a digital image constitutes the most essential information about the history of the image, thus its determinability is crucial for any successive forensic investigation to be conducted on the image. For instance, forensic investigators are often faced with the challenge of analyzing a large corpus of images of unknown provenance, e.g. downloaded from the social media sites. If the image provenance information can be recovered, the forensic investigators will be able to focus on the images of the same provenance and conduct more effective investigations, e.g. associating the images to the cameras or cellphones belonging to a suspect. Occasionally, the provenance information of an image

R. Islam et al. (Eds.): AusDM 2018, CCIS 996, pp. 372–383, 2019.
https://doi.org/10.1007/978-981-13-6661-1_29

can be extracted from the attached metadata, e.g. EXIF header, but this only grants limited reliability as the metadata can be easily edited or erased. A more reliable way would be to infer the provenance from the image data itself. It has been shown that some artifacts introduced by the in-camera processing components, either hardware or software, of the acquisition pipeline can be used to "fingerprint" the source camera. One such artifact is sensor pattern noise (SPN) [1], which mainly arises from the manufacturing imperfections of imaging sensors. The same SPN is left in every image taken by the same source camera, therefore, the images of the same provenance can be identified by examining the similarities between their SPNs, which are usually approximated as the noise residuals of the images.

Various SPN-based methods have been proposed to identify the source device of images [2–7] or group images of the same provenance [8–11]. However, almost all of these methods were evaluated on the high-quality images straight out from the camera without undergone any off-camera post-processing. With the rise of social networks, digital images have been continuously uploaded from computers or portable devices and shared through social media platforms. The increasingly rich built-in photo-editing features on social media platforms, e.g. the photo filters on Instagram or camera effects on Facebook, allow users to produce visually attractive photos at the tap of a finger. Consequently, the images downloaded from social networks may have gone through a series of manipulations, most of which are unknown, before they are handed to forensic investigators. In view of these facts, it is reasonable to question the trustworthiness of SPN in determining the provenance of images downloaded from social media sites. In this paper, we are particularly interested in the provenance analysis of the images posted on Instagram, which is one of the most popular photo-sharing platforms and offers a number of pre-defined photo filters. Using the performance of SPN-based provenance-oriented image clustering as an indicator, we intend to investigate the effects of Instagram filters on SPN-based image provenance analysis.

The rest of this manuscript is organized as follows: Sect. 2 briefly introduces the background and related works. Section 3 describes the details of the evaluation methodology. Section 4 presents the experimental results while Sect. 5 draws the conclusion and outlines the future work.

2 Background and Related Works

SPN, as its name indicates, is the fixed noise pattern that originates from the imaging sensor. The dominant component of SPN is the photo response non-uniformity (PRNU) noise, which is due to the variation of pixels' capabilities in converting photons into electrons. Such pixel-to-pixel discrepancy is very unique and commonly presents in every image captured by a sensor, making SPN a feasible choice for source device fingerprinting from images. Given an image $\boldsymbol{I}$, its SPN $\boldsymbol{n}$ can be approximated by the noise residual [1]:

$$\boldsymbol{n} = \boldsymbol{I} - \hat{\boldsymbol{I}}, \tag{1}$$

where $\hat{\boldsymbol{I}}$ is the denoised image. To see if the image is taken by a camera C, the normalized cross-correlation (NCC) similarity is examined:

$$\rho = corr(\boldsymbol{n}, \boldsymbol{r}) = \frac{(\boldsymbol{n} - \bar{\boldsymbol{n}}) \cdot (\boldsymbol{r} - \bar{\boldsymbol{r}})}{\|\boldsymbol{n} - \bar{\boldsymbol{n}}\| \|\boldsymbol{r} - \bar{\boldsymbol{r}}\|}, \tag{2}$$

where $\boldsymbol{r}$ is the reference SPN constructed by averaging the SPNs of multiple images captured by C to suppress other random interferences. $\boldsymbol{I}$ is deemed to be taken by C if ρ exceeds a predefined threshold. This task is referred to as source camera identification (SCI) in the literature. Similarly, for any two single images, if they are captured by the same camera, they should share the same SPN and have a relatively larger similarity. Based on the similarities, we are able to cluster the images of the same provenance. We will refer to this task as provenance-oriented image clustering, which is more challenging due to the unavailability of reference SPN and the necessity of examining the similarities of SPNs pairwise.

SPN has been proven to be effective for identifying the provenance of images [12]. However, most of the techniques based on SPN were evaluated on the images coming directly out of cameras, ranging from various private datasets to the widely used Dresden image database [13]. Nowadays, with the popularization of social media, the amount of digital images uploaded to and shared through social networks has explosively increased. Conducting forensic investigations on images downloaded from social networks will thus become increasingly common in the foreseeable future. The manipulations, usually unknown, that different social media platforms apply to images may attenuate the SPN signal in the image and thus are casting doubt on the trustworthiness of SPN in determining the provenance of digital images.

Despite the above problem, few studies have actually tried to evaluate the effectiveness of SPN on images from social media. Goljan *et al.* [12] conducted a large-scale test of camera identification from SPN on images downloaded form Flickr. Experiments on over one million pictures showed a false rejection rate <0.0238 at a false acceptance rate $<2.4 \times 10^{-5}$, which is a very promising result given that the images were taken by 6896 cameras covering 150 camera models. But it should also be noticed that Flickr allows the uploaded images to be stored in their original resolution with no or very little compression, so the results on Flickr are not representative compared to other social media platforms which may apply a series of manipulations on images.

In [14], Satta and Stirparo used SPN for linking a photo to social network accounts belonging to the person that has shot the photo. A probe photo P is considered to be taken by the account containing the image with the highest matching score to P. Evaluated on 2896 images from Flickr, Facebook, Google+ and personal blogs belonging to 30 different accounts, the method gave a correction recognition rate of ∼50%. Though it may not be high enough for accurate identification, such a performance shows the feasibility of using SPN for provenance analysis of images on social media. However, the rapidly evolving new tools and techniques constantly being deployed on social media platforms leave a huge gap for further studies.

Recently, it has been shown in some works, e.g. [15,16], that the image manipulations applied by each social network will leave some distinctive traces that can, in turn, be used to trace back to the social network origin of the image. This opens up new possibilities to associate an image with its social network provenance, but it also poses new challenges for tracing back to an image's acquisition device provenance via SPN, which may have been attenuated or removed after the image is uploaded. In this paper, we intend to investigate the effects of the photo filters of Instagram on the task of SPN-based provenance-oriented image clustering, the performance of which is a good indicator of how well the SPN is preserved after the filters are applied. Note that we did not evaluate the performance of SPN-based source camera identification because in many real-world scenarios, especially for social network data analysis, the reference SPN is not easy or impossible to obtain, but we believe that the performance of SPN-based source camera identification on Instagram photos should exhibit similar patterns and trend as the results presented in this paper.

3 Methodology

In this work, we sought to answer the following three closely related questions:

1. Would the filtering operations affect the SPN significantly and whether directly clustering the Instagram photos irrespective of the applied filters is possible?
2. How different filters affect the SPN?
3. Is it possible to identify the applied filter so that more reliable provenance analysis can be conducted with the information of the identified filters?

To answer the above three questions, a series of evaluations will be conducted. We will first blindly cluster a collection of images manipulated by different filters without knowing what filters have been applied. The clustering results of this experiment should give us an answer about to what extent the SPN can be affected by Instagram filters. Different image filters may have different impacts on the SPN, so in the second experiment, we will investigate the effects of different filters on SPN by performing SPN-based image clustering on images processed by individual filters. As can be expected, some filters may severely attenuate or remove the SPN in the images. So for the reliability of the provenance analysis, identifying and excluding the images manipulated by such filters is important. This requires us to be able to identify the filter applied to an image. For this purpose, in the third experiment, we will train three photo filter classifiers with Convolutional Neural Network (CNN), using the unprocessed images (with filters applied), denoised images and noise residuals, respectively.

Network Design. The design of the network is inspired by the well-known VGG-network [17] and the work in [18], where the VGG-net is used to extract the perceptual artistic styles from artworks. A main feature of VGG-net is that

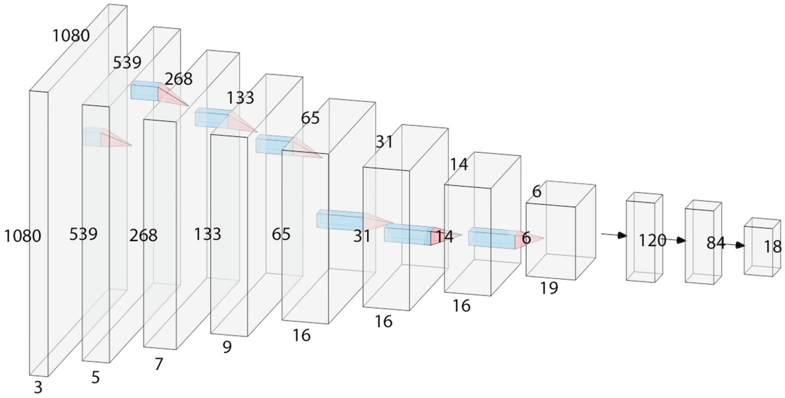


Fig. 1. The network architecture of the filter classifier. For brevity, only the convolutional and fully-connected layers are shown. The red pyramid represents the combination of convolution and max-pooling operations. The blue cuboid represents the receptive field of each convolutional filter, which has a size of 3 × 3 throughout the network. The dimensions of each layer are labeled by the numbers on the edges. (Color figure online)

it uses convolutional layers with a small receptive field (e.g. 3×3) combined with a large depth of the network. The layers in the convolutional neural network can be considered as a set of different image filters and each filter extracts certain features of the image. While including higher layer information can lead to a finer representation of the artistic style, the artistic style can be mostly represented by the lower layers of a neural network. It indicates that a shallower network might be able to capture the features to classify the Instagram filters without compromising too much performance. To further reduce the computational cost, in this work, we use a network with 7 convolutional layers followed by 3 fully connected layers (Fig. 1), which is more compact than the shallowest VGG-net architecture (11 layer VGG-net) in [17] and the 19 layer VGG-net used in [18]. All the convolutional layers have a receptive field size of 3×3 and all the hidden layers are equipped with the rectification non-linearity (ReLU) and batch normalization. Each convolutional layer is followed by a max-pooling layer, which is performed over a 2×2 pixel window. The network takes 3-channels (RGB) images of size 1080×1080 pixels (the maximum allowable image resolution on Instagram at the time being) as input and returns a vector of length 18 to accomplish an 18-class classification (to classify 17 different Instagram filters tested in this work and the original image).

Network Inputs. Since we can consider the Instagram filters as the artistic styles of the images as a whole, removing the noise can reduce the pixel-wise disturbance and may improve the training of the classifier. However, on the other hand, we notice that the filters can not only change the visual style in the

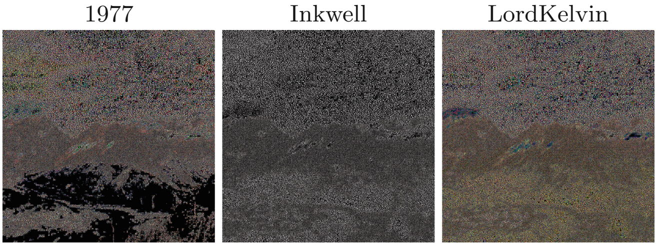


Fig. 2. The noise residuals extracted from the three images shown in Fig. 3.

images but also alter the noise residuals of the images in very different manners. For example, some filters have high contrast level which can suppress the noise at both ends of histogram. So the filtered images tend to have more flattened regions of noise, as exemplified by the image manipulated by filter '1977' in Fig. 2. As another example, some filters can have different color profiles, so the noise behavior may vary considerably across different color channels (e.g. comparing Inkwell and LordKelvin in Fig. 2). These significant differences introduced by different filters in the noise residuals motivate us to use noise residuals as the input for constructing the filter classifier. Given all these facts, we train three classifiers of the same neural network architecture using three different inputs: the unprocessed images $\boldsymbol{I}$ (with filters applied), denoised images $\hat{\boldsymbol{I}}$ and image noise residuals $\boldsymbol{n}$. For simplicity, we call the three networks as $\boldsymbol{I}$-net, $\hat{\boldsymbol{I}}$-net and $\boldsymbol{n}$-net, respectively.

4 Experiments

4.1 Dataset

The experiments were conducted on the images from 25 different mobile devices in the VISION dataset [19], with each device accounting for more than 200 JPEG images. These images were first cropped to a size of 1080 × 1080 pixels. For each image, we then applied 17 different filters with the Instagram app on iOS. We consider the filters as black boxes without knowing the details of how the images are manipulated. The list of the filters applied can be found in Fig. 3, where one example image is shown for each filter while the original image is labeled as 'Normal'. Thus, together with the original image, we have 18 different versions for each image from the VISION dataset, which results in $96,660$ images in total. The whole set or a subset of these $96,660$ images will be used in the following experiments.

4.2 Evaluation Matrix

As can be noticed in Fig. 3, to produce different artistic visual effects, each filter may alter each color channel in very different manners. For this reason, we will evaluate the clustering performance with the noise residuals extracted from

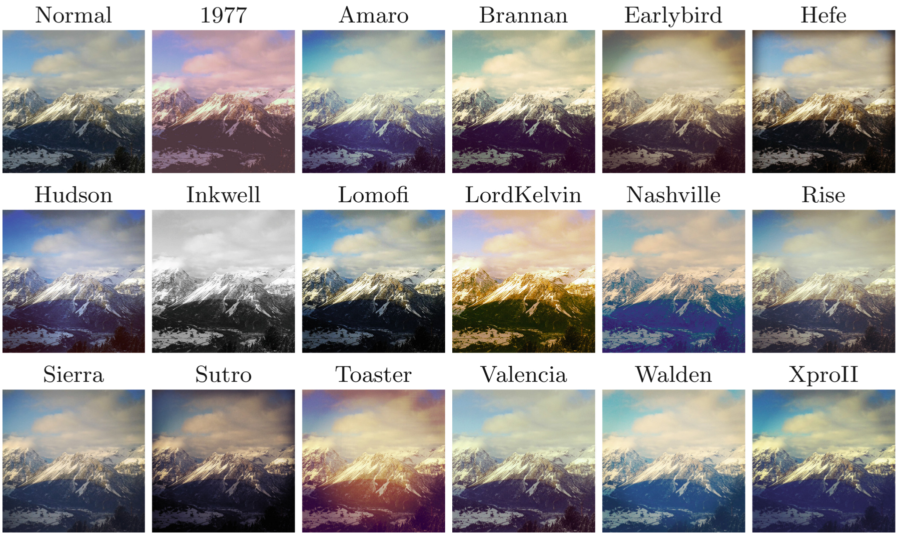


Fig. 3. Example images of the 17 Instagram filters together with the original image (Normal) used in our experiment.

all three color channels (RGB) using BM3D denoising algorithm [20]. The fast clustering algorithm in [21] will be used to cluster the images and the clustering quality is measured by F1-measure, which is defined as:

$$\mathcal{F} = 2 \cdot \frac{\mathcal{P} \cdot \mathcal{R}}{\mathcal{P} + \mathcal{R}} \tag{3}$$

where $\mathcal{P}$ and $\mathcal{R}$ are the average precision and recall rate respectively. They can be calculated as:

$$\begin{cases} \mathcal{P} = \sum_i |c_i \cap \psi_{j^*}| / \sum |c_i| \\ \mathcal{R} = \sum_i |c_i \cap \psi_{j^*}| / \sum |\psi_{j^*}| \end{cases} \tag{4}$$

$|c_i|$ is the size of the cluster c_i, and $|\psi_{j^*}|$ is the size of the class ψ_{j^*} (i.e. the number of images captured by camera j^*) corresponding to the largest sub-cluster in cluster c_i, i.e. $j* = \mathrm{argmax}_j\{|c_i \cap \psi_j|\}$.

4.3 Results and Analyses

Is It Possible to Blindly Cluster the Instagram Photos? We randomly selected 1800 images filtered by 18 different filters (including the 'Normal' class), each responsible for 100 images. As can be seen in Table 1, the low recall rates indicate the failure of the clustering as many small or singleton clusters are produced. If we assume that the filters do not affect the SPN at all, there will be 72 images from each camera on average. Given the results reported in [21], clustering such a dataset is an easy task and a high $\mathcal{F}$ should be expected for the clustering algorithm described in [21]. However, the rather contradictory results presented in Table 1 imply that the photo filters severely affect the SPN and make direct clustering on Instagram photos infeasible.

Table 1. Clustering results on unclassified images.

Color channel	$\mathcal{F}$ (%)	$\mathcal{R}$ (%)	$\mathcal{P}$ (%)
Red	7.78	4.16	60.28
Green	7.59	4.06	58.67
Blue	7.63	4.08	58.56

How Different Filters Affect the SPN? We then investigate the effect of individual filters on the performance of SPN-based image clustering. We randomly selected 1000 images from the 25 cameras for each filter. To make sure the comparison of clustering performance between the filters is representative and unbiased, images selected for each filter are generated from the same set of the original images. Table 2 shows the clustering results based on the noise residuals from R, G, B channels for different filters. The highest F1-measure among the three color channels is highlighted in bold for each filter. Based on the performance difference relative to the 'Normal' class, we divide the 17 filters into two groups:

- Group I (filters highlighted in the violet background in Table 2). The clustering on the images processed by the filters in this group fails completely, with an F1-measure of 12.38%. Further investigation reveals that the clustering algorithm produces a singleton cluster containing all the images processed by each filter, which indicates the filters in this group greatly damage the SPN in the image. The 1000 test images for each filter are selected randomly from 25 cameras. The most common source camera for the test images contributes 66 images, which accounts for the F1-measure of 12.38% in a singleton cluster of 1000 images.
- Group II (filters highlighted in the green background in Table 2). The highest $\mathcal{F}$ among different color channels is comparable to that of the 'Normal' class. So for the images processed by the filters in this group, SPN-based techniques are still effective for analyzing the provenance. Furthermore, the clustering performance stays quite stable across different color channels, with the exception of 'LordKelvin' filter, which applies radical adjustments to the blue channel.

For all the filters we have tested in this paper, it might be a little surprising to see that the green channel delivers a better (or at least comparable) performance than the other two color channels. So, without any prior information about the applied filters, the best bet would be to apply analysis on the SPN extracted from the green channel.

Table 2. Clustering results for different Instagram filters using SPNs extracted from different color channels.

Filters	$\mathcal{F}(\%)$		
	Red	Green	Blue
Normal	85.83	**85.96**	86.4
1977	77.45	**85.90**	72.21
Amaro	12.38	**12.38**	12.38
Brannan	85.05	**85.61**	83.84
Earlybird	**87.06**	85.96	82.62
Hefe	12.38	**12.38**	12.38
Hudson	12.38	**12.38**	12.38
Inkwell	85.75	**85.75**	85.75
Lomofi	84.7	**87.07**	82.39
LordKelvin	82.08	**86.58**	36.71
Nashville	82.97	**85.75**	81.99
Rise	12.38	**12.38**	12.38
Sierra	12.38	**12.38**	12.38
Sutro	12.38	**12.38**	12.38
Toaster	12.38	**12.38**	12.38
Valencia	84.43	85.13	**85.65**
Walden	85.17	**86.27**	80.22
XproII	84.76	**85.29**	83.71

Is It Possible to Identify the Applied Filter? We have seen that for the filters in Group I, the clustering fails completely for all three color channels. Therefore, for the reliability of forensic investigations, the images processed by those filters should be identified beforehand and excluded in the subsequent provenance analysis. For this purpose, we train three photo filter classifiers with the same network architecture depicted in Fig. 1, using the unprocessed images (with filters applied), denoised images and noise residuals, respectively. We train each network for 50 epochs using the cross-entropy loss function and a learning rate of 0.002. 96, 660 images in the dataset are split into training, validation and test sets with a ratio of 60%:20%:20%.

We evaluate the trained classifiers on 19, 332 test images and show the detailed classification results for individual filters in terms of $\mathcal{P}$ and $\mathcal{R}$ in Table 3. On average, the $\boldsymbol{I}$-net, $\hat{\boldsymbol{I}}$-net, and $\hat{\boldsymbol{I}}$-net achieve an accuracy of 79.91%, 86.93% and 88.38%, respectively. It shows that by training with the de-noised images or noise residuals, a better overall classification performance can be achieved. Though $\boldsymbol{I}$-net and $\hat{\boldsymbol{I}}$-net have seemingly good overall classification accuracy, they may be problematic in some cases. Taking the 'Normal' class as an example, the recall and precision rates of $\boldsymbol{I}$-net and $\hat{\boldsymbol{I}}$-net are considerably lower compared to $\boldsymbol{n}$-net, which makes $\boldsymbol{I}$-net (52.89%) and $\hat{\boldsymbol{I}}$-net (66.76%) unsuitable for identifying the images with no filter applied at all. For an original natural image, although no filter is applied, it may look like as if a specific filter has been applied just because its content tends to be similar to the filter's artistic style. So as pointed out in [18], the image content and artistic style always co-exist in natural images, which makes it difficult to classify different filters (or artistic styles) for the original images. In contrast, the precision rate of $\boldsymbol{n}$-net (98.53%) on classifying original images is nearly perfect because by using the noise residuals as the input data, the network is less affected by the image content and focuses more on the features that can differentiate the original images from other filtered images.

Table 3. Classification performance of $\boldsymbol{I}$-net, $\hat{\boldsymbol{I}}$-net, and $\hat{\boldsymbol{I}}$-net, which are trained with three different types of data: the unprocessed images $\boldsymbol{I}$ (with filters applied), denoised images $\hat{\boldsymbol{I}}$ and image noise residuals $\boldsymbol{n}$, respectively.

Filters	$\mathcal{R}(\%)$			$\mathcal{P}(\%)$		
	$\boldsymbol{I}$-net	$\hat{\boldsymbol{I}}$-net	$\boldsymbol{n}$-net	$\boldsymbol{I}$-net	$\hat{\boldsymbol{I}}$-net	$\boldsymbol{n}$-net
Normal	65.55	67.69	**93.48**	52.89	66.76	**98.53**
1977	79.14	**92.36**	92.09	86.47	**91.68**	89.75
Amaro	69.55	83.33	**84.17**	**83.74**	79.27	73.80
Brannan	**94.41**	87.62	91.24	72.90	**91.72**	82.35
Earlybird	85.85	95.16	**96.00**	84.90	**88.03**	72.61
Hefe	**85.94**	85.01	84.08	83.00	87.79	**94.55**
Hudson	91.90	95.07	**98.70**	87.58	90.11	**98.15**
Inkwell	56.42	94.88	**98.60**	94.10	90.66	**95.66**
Lomofi	69.46	68.53	**73.09**	58.88	69.50	**90.54**
LordKelvin	94.97	92.55	**95.44**	90.19	93.86	**97.53**
Nashville	85.57	**94.04**	80.26	81.76	**92.24**	89.60
Rise	64.62	**77.28**	72.16	80.32	**86.82**	80.31
Sierra	77.09	81.01	**95.53**	72.44	80.04	**98.65**
Sutro	88.27	**92.74**	92.55	87.21	**95.22**	90.12
Toaster	92.27	96.00	**97.21**	**98.21**	97.45	94.90
Valencia	69.55	77.37	**78.21**	66.11	78.40	**82.27**
Walden	88.83	94.04	**96.09**	91.47	**95.92**	80.56
XproII	78.96	**90.13**	71.88	87.78	90.21	**91.69**

As the images manipulated by the filters in Group I are not suitable to be used in SPN-based provenance analyses, we need to move one step further to differentiate the filters from the two groups, so we further conduct a binary group-wise classification to classify the images into two groups corresponding to the Group I and II in Table 2. The confusion matrix for the group-wise classification by the three classifiers is shown in Table 4. We can see that the $\boldsymbol{n}$-net clearly outperforms the other two classifiers with higher true positive rates and lower false rates. It shows that by applying the $\boldsymbol{n}$-net to classify the images before provenance analysis, we will be able to identify and exclude the majority (96.83%) of the images in Group I and only discard a very small portion (1.61%) of the images with reliable SPN in Group II. This will greatly help the forensic investigators to reduce the size of data to be analyzed and make the results from SPN-based techniques more trustworthy.

Table 4. Confusion matrix for the classification of the Group I and Group II Instagram filters.

Actual/Predicted	$\boldsymbol{I}$		$\hat{\boldsymbol{I}}$		$\boldsymbol{n}$	
	Group I	Group II	Group I	Group II	Group I	Group II
Group I	0.8889	0.1111	0.9258	0.0742	**0.9683**	**0.0317**
Group II	0.0468	0.9532	0.0416	0.9584	**0.0161**	**0.9839**

5 Conclusions and Future Work

In this work, we have shown that some Instagram filters, i.e. the filters in Group I, can significantly attenuate or modify the SPN signal in the images and thus hinder the SPN-based image provenance analysis, while some other filters, i.e. the filters in Group II, well preserve the SPN in the images. Furthermore, due to the varying quality of the preserved SPNs across different image filters, separate treatments are needed for the images processed by different filters when we conduct SPN-based provenance analysis on Instagram photos. We also show that it is possible to identify the filter applied to an image by training a CNN-based classifier. We trained three classifiers of the same network architecture with three different types of inputs: the unprocessed images (with filters applied), the denoised images and the noise residuals. We found that the classifier trained with the noise residuals clearly outperforms the other two classifiers, with an accuracy of more than 96% in identifying the filters in Group I and II. We believe this work can help the forensic investigators to pre-process the images and facilitate the reliable SPN-based provenance analysis for Instagram photos. As the future work, we will conduct a larger-scale evaluation to cover more Instagram filters and extend this work to the analysis of the data on other social media platforms.

References

1. Lukas, J., Fridrich, J., Goljan, M.: Digital camera identification from sensor pattern noise. IEEE Trans. Inf. Forensics Secur. **1**(2), 205–214 (2006)
2. Chen, M., Fridrich, J., Goljan, M., Lukás, J.: Determining image origin and integrity using sensor noise. IEEE Trans. Inf. Forensics Secur. **3**(1), 74–90 (2008)
3. Hu, Y., Yu, B., Jian, C.: Source camera identification using large components of sensor pattern noise. In: Proceedings of International Conference on Computer Science and Its Applications, pp. 291–294 (2009)
4. Li, C.T.: Source camera identification using enhanced sensor pattern noise. IEEE Trans. Inf. Forensics Secur. **5**(2), 280–287 (2010)
5. Kang, X., Li, Y., Qu, Z., Huang, J.: Enhancing source camera identification performance with a camera reference phase sensor pattern noise. IEEE Trans. Inf. Forensics Secur. **7**(2), 393–402 (2012)
6. Lin, X., Li, C.T.: Preprocessing reference sensor pattern noise via spectrum equalization. IEEE Trans. Inf. Forensics Secur. **11**(1), 126–140 (2016)
7. Lin, X., Li, C.T.: Enhancing sensor pattern noise via filtering distortion removal. IEEE Signal Process. Lett. **23**(3), 381–385 (2016)

8. Bloy, G.J.: Blind camera fingerprinting and image clustering. IEEE Trans. Pattern Anal. Mach. Intell. **30**(3), 532–534 (2007)
9. Li, C.T.: Unsupervised classification of digital images using enhanced sensor pattern noise. In: Proceedings of the IEEE International Symposium on Circuits and Systems, pp. 3429–3432, May 2010
10. Caldelli, R., Amerini, I., Picchioni, F., Innocenti, M.: Fast image clustering of unknown source images. In: Proceedings of IEEE International Workshop on Information Forensics and Security. pp. 1–5, December 2010
11. Lin, X., Li, C.T.: Large-scale image clustering based on camera fingerprints. IEEE Trans. Inf. Forensics Secur. **12**(4), 793–808 (2017)
12. Goljan, M., Fridrich, J., Filler, T.: Large scale test of sensor fingerprint camera identification. In: IS&T/SPIE Electronic Imaging, p. 72540I. International Society for Optics and Photonics (2009)
13. Gloe, T., Bhme, R.: The dresden image database for benchmarking digital image forensics. J. Digital Forensic Pract. **3**(2–4), 150–159 (2010)
14. Satta, R., Stirparo, P.: On the usage of sensor pattern noise for picture-to-identity linking through social network accounts. In: International Conference on Computer Vision Theory and Applications (VISAPP), vol. 3, pp. 5–11 (2014)
15. Caldelli, R., Becarelli, R., Amerini, I.: Image origin classification based on social network provenance. IEEE Trans. Inf. Forensics Secur. **12**(6), 1299–1308 (2017)
16. Amerini, I., Uricchio, T., Caldelli, R.: Tracing images back to their social network of origin: a CNN-based approach. In: 2017 IEEE Workshop on Information Forensics and Security (WIFS), pp. 1–6, December 2017
17. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
18. Gatys, L.A., Ecker, A.S., Bethge, M.: A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576 (2015)
19. Shullani, D., Fontani, M., Iuliani, M., Shaya, O.A.: VISION: a video and image dataset for source identification. EURASIP J. Inf. Secur. **2017**(1), 15 (2017)
20. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE Trans. Image Process. **16**(8), 2080–2095 (2007)
21. Li, C.T., Lin, X.: A fast source-oriented image clustering method for digital forensics. EURASIP J. Image Video Process. **1**, 69–84 (2017). Special issues on image and video forensics for social media analysis

Industry Showcase

How Learning Analytics Becomes a Bridge for Non-expert Data Miners: Impact on Higher Education Online Teaching

Katherine Herbert and Ian Holder

Charles Sturt University, Wagga Wagga, NSW 2650, Australia
kherbert@csu.edu.au

Abstract. This paper builds on the current studies on data mining's potential benefits to online learning environments. Many Teaching Academics who are non-experts in data mining techniques however are not able to take advantage of these potential benefits. The objective of this paper is to illustrate how learning analytics is bridging the gap between data mined from Learning Management Systems and teaching practice development in higher education, specifically for Teaching Academics who recently transitioned into online teaching. The authors suggest that bridging this gap is an essential step in the development of online teaching practices and online courses. A customised Dashboard that curates data mined from a university's LMS is discussed, showcasing the impact on the practices of Teaching Academics. The results from the preliminary exploration suggest that learning analytics can bridge the gap between expert and non-experts of data mining techniques and can become a valuable tool for teaching practice development.

Keywords: Learning and teaching · Professional learning · Learning analytics · Data mining visualisation

1 Introduction

Studies on the application of data mining techniques to monitor student progress in Learning Management Systems (LMS) [1–3] provide insight into data mining's potential in online learning environments. The techniques however are essentially established and written by and for data mining experts [4]. Therefore, many Teaching Academics who are non-experts in data mining techniques are not able to take advantage of these potential benefits. The objective of this paper is to illustrate how learning analytics is bridging the gap between data mined from LMS and teaching practice development in higher education, specifically for Teaching Academics who recently transitioned into online teaching. The authors suggest that bridging this gap is an essential step in the development of online teaching practices. Utilising a learning analytics Dashboard, customised information is curated to provide insight into the data that is mined from the LMS. This showcases the impact on the practices of Teaching Academics supported by the authors, respectively a Lecturer in Learning and Teaching and a Learning Analytics Specialist, during the winter and spring semesters in 2017.

R. Islam et al. (Eds.): AusDM 2018, CCIS 996, pp. 387–395, 2019.
https://doi.org/10.1007/978-981-13-6661-1_30

The results from the preliminary exploration suggest that learning analytics can bridge the gap between expert and non-experts of data mining techniques and can become a valuable tool for teaching practice development.

This paper will proceed with an overview of the current state of online learning in higher education. It then looks at what data is currently mined by a university's LMS. The Dashboard design and implementation is then showcased. Finally, results from the preliminary exploration is discussed, including implications for future research around professional development for Teaching Academics in online learning environments.

2 Online Learning in Higher Education

Flexibility of access to learning is one characteristic of online learning. Online learning delivery offers opportunities to study anytime and anywhere. This observed improved access to learning, and more flexible learning, is attributed to both better technologies and the internet [5]. It has been suggested by surveys, conducted on the state of higher education in the last 5 years [6, 7], that it is this flexibility that has contributed to student-demand for online learning courses. Enrolment into online courses enables students to further their education while remaining in steady employment [6].

To service higher education institutions' strategic plans on quality online courses, research in the design of online learning and courses in higher education has grown since the introduction of the internet over 15 years ago [8]. It is therefore logical that designing online learning experiences for students entails Teaching Academics having the skills and knowledge to deliver online courses.

One of the most common ways online courses are delivered is through an LMS. The data produced by the activity of both students and Teaching Academics within the LMS can therefore be mined and analysed. While studies by [1–4] provide insights into the experiences of students when using data mining techniques to design online learning experiences; there appears a need to look at how these techniques are then interpreted for non-experts who teach in these environments, to support and grow their teaching practice. Specifically, those who teach online in higher education.

This paper now turns its attention to what data is currently mined in the University's LMS.

3 LMS Data Mining

Blackboard is the University's LMS platform. The platform not only houses content, Blackboard tools also offer opportunities for students to engage online with each other, as well as with the Teaching Academic, both synchronously, as well as asynchronously. How well the tools are utilised depends greatly on the skills of the Teaching Academic.

Due to the rapid development of online learning tools including those that are implemented in an LMS, it can be said that data generated by activities that happen within the LMS is valuable to those who design, deliver and teach online only if it can be interpreted and used to improve their practice [4]. While [4] proposed a recommender framework based on meta-features, the authors of this paper suggest that without the support of a tool that will make data interpretations more concrete, then Teaching Academics who are non-experts in data mining will not benefit from the data available to them.

In this paper's particular context, Teaching Academics are aware of readily available reports in Blackboard which provides reports in various sections of the tool including Course Reports, Performance Dashboard, Retention Centre, SCORM Reports, and Site Analytics. The first four of these are reported directly from the Blackboard relational database, whereas Site Analytics reports from two different sources – a SQL Server Database that has aggregated the data from the Blackboard relational database and a Data Warehouse which has transformed data from the SQL Server Database is used by the learning analytics team to report and create dashboards for staff consumption. These reports from the Blackboard Database and several reports from the Analytics SQL Server Database are automatically generated and easily accessed via the course sites of subjects taught by Teaching Academics. It was evident however that these reports are often unused because the description of the reports are too vague or could have many meanings, and Teaching Academics often do not understand how the data is being curated. This is especially true for Teaching Academics who have taught in face-to-face classrooms but are now expected to create the same learning experiences or even better in online settings. As [8] case study shows, teaching online requires a different set of skills which need to be learned and enhanced as quickly as technology develops. Therefore, a single dashboard which specifically targets minimum skills and knowledge requirements to teach online seemed useful.

4 LLT Dashboard

The role of Lecturer in Learning and Teaching (LLT) calls for mentoring and coaching Teaching Academics moving into online environments for the first time, as well as continuous improvement of online teaching practices of experienced Teaching Academics. This is in line with the University's aim to become a leader in higher education online learning.

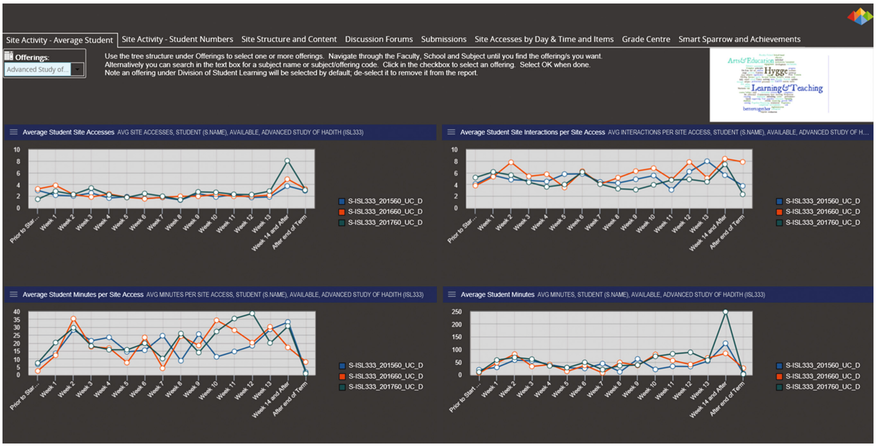
Two policies were introduced to leverage the University's strategic goals of providing innovative quality online learning and teaching student. The first was the Initiatives and Improvement (I & I) plan that is implemented by each faculty on an annual basis. The second was the Quality Learning and Teaching (QLT) standards.

In 2017, the I & I plan for each faculty outlined the key focus areas for learning and teaching. The key areas of improvement highlighted were LMS layout and design, student to student engagement, as well as improvement of the quality of student to teacher interaction online. The QLT standards introduced in late 2016, are a set of key performance indicators (KPI) that underpins the minimum expectation for student learning experiences online as well as on-campus. These standards inform the I & I plan and together frame the continuous improvement of online courses. While it is not the intention of this paper to discuss the I & I plan and QLT standards in-depth, it intends to reflect that these two frameworks informed the work then carried out to design the LLT Dashboard.

Having identified a gap between LMS Data Mining Reports and Teaching Academics' skill to understand what the reports meant for their practice, the LLT and Learning Analytics Specialist began work on a Dashboard. In the Autumn Semester of 2017, the LLT, in collaboration with the Learning Analytics Specialist, decided that curating and displaying the data visually and in one space could bridge this gap.

As part of the Blackboard Analytics for Learn [A4L] package, access is given to the data visualization tool Pyramid BI Office. This tool connects to the A4L Data Warehouse which has a default configuration by Blackboard and can be customized to the institution's needs. To create the LLT dashboard both standard features were used, and standard measures were retrieved from the data warehouse, as well as custom measures and values being calculated to present meaningful data to the LLT.

The LLT Dashboard (Fig. 1) is designed as an eagle's eye view of subjects, viewed both on their own, as well as across past semesters. The tabs at the top of the Dashboard looks at course site structure and content, as well as how students navigate within the course site (LMS layout and design); the next area looks at communication tools such as online meetings, as well as discussion forums (student to student engagement); and finally, a heatmap which reveals activities of both students and Teaching Academics (quality of student to teacher interaction online). The Dashboard aims to operationalise and internalise the data mined from the LMS and draw direct links to the Teaching Academics online teaching activities.


Use the tree structure under Offerings to select one or more offerings. Navigate through the Faculty, School and Subject until you find the offering/s you want. Alternatively you can search in the text box for a subject name or subject/offering code. Click in the checkbox to select an offering. Select OK when done. Note an offering under Division of Student Learning will be selected by default; de-select it to remove it from the report.

Offerings: Advanced Study of...

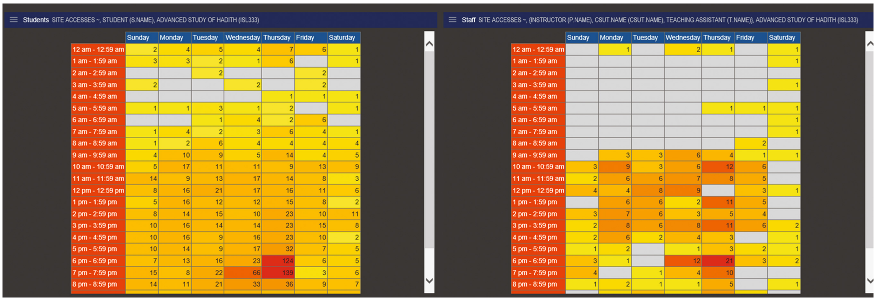
Site Activity - Average Student
Site Activity - Student Numbers
Site Structure and Content
Discussion Forums
Submissions
Site Accesses by Day & Time and Items
Grade Centre
Smart Sparrow and Achievements
Arts&Education
Hygge
Learning&Teaching
bettertogether
Average Student Site Accesses AVG SITE ACCESSES, STUDENT (S.NAME), AVAILABLE, ADVANCED STUDY OF HADITH (ISL333)
Average Student Site Interactions per Site Access AVG INTERACTIONS PER SITE ACCESS, STUDENT (S.NAME), AVAILABLE, ADVANCED STUDY OF H...
Average Student Minutes per Site Access AVG MINUTES PER SITE ACCESS, STUDENT (S.NAME), AVAILABLE, ADVANCED STUDY OF HADITH (ISL333)
Average Student Minutes AVG MINUTES, STUDENT (S.NAME), AVAILABLE, ADVANCED STUDY OF HADITH (ISL333)
0
2
4
6
8
10
5
15
20
25
30
35
40
50
100
150
200
250
Prior to Star...
Week 1
Week 2
Week 3
Week 4
Week 5
Week 6
Week 7
Week 8
Week 9
Week 10
Week 11
Week 12
Week 13
Week 14 and After
After end of Term
S-ISL333_201560_UC_D
S-ISL333_201660_UC_D
S-ISL333_201760_UC_D

Fig. 1. LLT Dashboard layout and tabs

The site structure and content tabs are a great way to get an overview, as well as a way to initially gauge how the LMS is being used in the subject, i.e. is it simply a repository? Does it serve as a jumping point? Or is it the main platform for learning in this subject?

Communication tools in the LMS play different parts in online learning. Discussion forums are a good starting point for asynchronous interaction, and one of the easiest ways to engage students who need flexible access to learning. The data drawn from the discussion forums allows the LLT and Teaching Academic to review what the data is saying about the strengths of peer interaction in the online space.

The heatmap of activity (Fig. 2) reveals the students times of high activity within the LMS (heatmap on the left), including when they access the course site the most. The heatmap on the right, reveals times that the Teaching Academic accesses the site.



Students SITE ACCESSES ~, STUDENT (S.NAME), ADVANCED STUDY OF HADITH (ISL333)

	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
12 am - 12:59 am	2	4	5	4	7	6	1
1 am - 1:59 am	3	3	2	1	6		1
2 am - 2:59 am			2			2	
3 am - 3:59 am	2			2		2	
4 am - 4:59 am					1	1	1
5 am - 5:59 am	1	1	3	1	2		1
6 am - 6:59 am			1	4	2	6	
7 am - 7:59 am	1	4	2	3	6	4	1
8 am - 8:59 am	1	2	6	4	4	4	4
9 am - 9:59 am	4	10	9	5	14	4	5
10 am - 10:59 am	5	17	11	11	9	13	9
11 am - 11:59 am	14	9	13	17	14	8	3
12 pm - 12:59 pm	8	16	21	17	16	11	6
1 pm - 1:59 pm	5	16	12	12	15	8	2
2 pm - 2:59 pm	8	14	15	10	23	10	11
3 pm - 3:59 pm	10	16	14	14	23	15	8
4 pm - 4:59 pm	10	16	9	16	23	10	2
5 pm - 5:59 pm	10	14	9	17	32	7	5
6 pm - 6:59 pm	7	13	16	23	124	6	5
7 pm - 7:59 pm	15	8	22	66	139	3	6
8 pm - 8:59 pm	14	11	21	33	36	9	7

Staff SITE ACCESSES ~, {INSTRUCTOR (P.NAME), CSUT.NAME (CSUT.NAME), TEACHING ASSISTANT (T.NAME)}, ADVANCED STUDY OF HADITH (ISL333)

	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
12 am - 12:59 am		1		2	1		1
1 am - 1:59 am							1
2 am - 2:59 am							
3 am - 3:59 am							1
4 am - 4:59 am							
5 am - 5:59 am					1	1	1
6 am - 6:59 am							1
7 am - 7:59 am							1
8 am - 8:59 am						2	
9 am - 9:59 am		3	3	6	4	1	1
10 am - 10:59 am	3	9	3	6	12	6	
11 am - 11:59 am	2	6	6	7	8	5	
12 pm - 12:59 pm	4	4	8	9		3	1
1 pm - 1:59 pm		6	6	2	11	5	
2 pm - 2:59 pm	3	7	6	3	5	4	
3 pm - 3:59 pm	2	8	6	8	11	6	2
4 pm - 4:59 pm	2	6	2	4	3		1
5 pm - 5:59 pm	1	2		1	3	2	1
6 pm - 6:59 pm	3	1		12	21	3	2
7 pm - 7:59 pm	4		1	4	10		
8 pm - 8:59 pm	1	2	1	1	5		1

Fig. 2. Heatmap of activity

This Dashboard was then used during winter and spring semesters in 2017. The LLT would provide support and mentoring throughout the two sessions, utilising the Dashboard in one-to-one professional learning sessions with Teaching Academics.

5 Discussion

What emerged from this exploration was opportunities to optimise uptake of professional learning through the LLT Dashboard. The Dashboard became a benchmarking tool of sorts throughout the semester. Having the ability to check the Dashboard to see any changes in student engagement with content, forums, general access to the site, and compare it to previous semesters, allowed Teaching Academics to find the connection between how the LMS is utilised and designed to leverage the online delivery of their subjects, all underpinned by data mined from their LMS course sites. In some cases, to validate the Dashboard information, Teaching Academics engaged their students in a mid-session student survey, using google forms. Exploring what is working and what is not and comparing the feedback to data being curated and displayed in the Dashboard. As an example of this, in the middle of semester it was found that students were asking for location of files in folders. The Dashboard revealed that navigation to the files could be better designed. After a quick review of folder depth and number of clicks on the Dashboard, the content areas of the course site were rehashed so clicking would be minimised and content quickly accessed.

Student to student engagement is one area where those who teach in online courses need visualisation to concretise the connection between the communication tool in the LMS and student interaction. Discussion boards are readily available as a communication tool in the LMS. This is also one of the main tools introduced to Teaching Academics when they first teach online. For the beginner, one transitioning from face-to-face teaching to online teaching, discussion boards can be set up for safe spaces to encourage peer conversations. For a user with some experience of discussion boards, this could be the opportunity to further develop the use of the online discussions. For advanced users, the data collected by the Dashboard allowed them to check the effectiveness of their strategies.

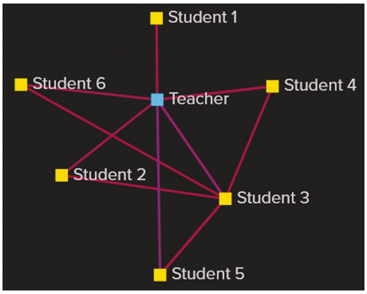


Fig. 3. Example of a Beginner Teaching Academic utilizing a discussion forum

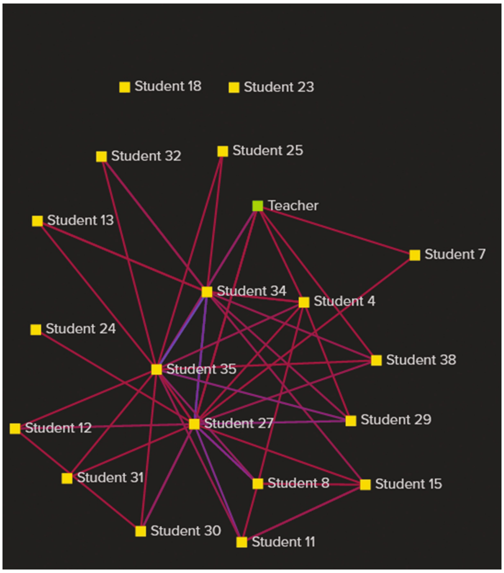


Fig. 4. Example of Teaching Academic with some experience of discussion forum

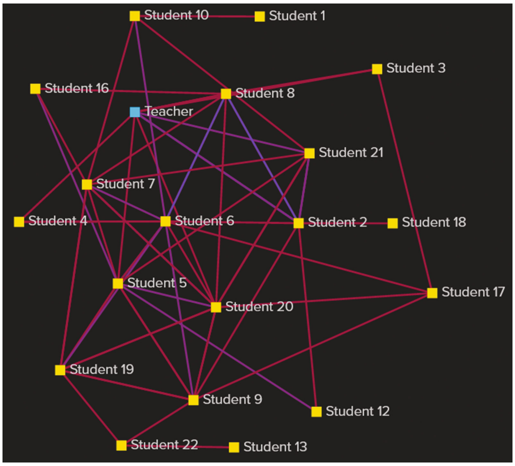


Fig. 5. Example of Teaching Academic with advanced skills in discussion forum

For a beginner, it can be seen in Fig. 3 that there is good evidence that structuring the forum to encourage discussion allows the Teaching Academic to create a safe space for peer conversations and discussion. The Teaching Academic with some experience (Fig. 4) utilised the discussion forums a step further by responding on a weekly basis via a group announcement, highlighting key points made by students. In Fig. 4, you can see that the Teaching Academic had a lot of engagement with the discussion forum, but he also noticed that two students were talking to themselves. Finally an advanced user (Fig. 5) who was curious to see if his strategies really worked in terms of allowing the discussions to be led by students, you can see there is evidence that it is happening. The Teaching Academic is not at the centre of the conversation. This visualisation has informed the Teaching Academics' practice and the evolution of how a discussion forum can be used is already evident. This also illustrates how understanding where data is mined from and how it then forms these visualisations help Teaching Academics buy into the application of data mining to their teaching practice.

Towards the end of the session, the Dashboard is revisited. Reflecting and reviewing what the Teaching Academic experienced while teaching online and what the Dashboard captured. In one instance, student patterns of activity during the week were explored. In this specific subject, where online meetings were scheduled once a week, the students appeared to access the course site an hour before the online meeting, downloading readings and creating forum posts. During the delivery of this specific subject, the Teaching Academic was encouraged to reply to forum posts the day before the scheduled online meeting. It was discovered that the Teaching Academic got more engaged discussions from students in the online meeting itself because students were accessing the course site and responding on the forums just before the online meeting. Teaching Academics revealed that students were found to quote each other during the online meetings.

In the main, the LLT Dashboard helps with conversation starters, just-in-time analysis, as well as a reflection tool for professional development. Where once, the LMS Data Mining Reports were vague and did not appear to have any relevance to non-expert data miners, the Dashboard allows Teaching Academics to operationalise and internalise the LMS data. They could also see how data mined from the LMS, interpreted and leveraged by the Dashboard could benefit their teaching practice and continuous improvement.

6 Conclusion

This paper gave an overview of the current state of online learning in higher education, and why LMS data mining techniques could positively impact online teaching practice. It then looked at what data is currently mined by an LMS. The Dashboard design and implementation was showcased. Finally, results from the preliminary exploration was discussed. This showcased the impact on the practices of Teaching Academics supported by the authors, respectively a Lecturer in Learning and Teaching and a Learning Analytics Specialist, using the LLT Dashboard during winter and spring semesters in 2017.

The results from the preliminary exploration suggest that learning analytics can bridge the gap between expert and non-experts of data mining techniques and can become a valuable tool for teaching practice development. Utilising a learning analytics Dashboard, customised information is curated to provide insight into the data that is mined from the LMS. The implications that data mining techniques have a potential to increase the quality of online course design and delivery needs future research, especially around professional development for Teaching Academics in online learning environments. While data mining techniques are developed and utilised by experts, Teaching Academics are mostly non-experts. Therefore, there is a need to be able to not only access the data mined but also interpret and analyse the information to benefit their online teaching practice.

The objective of this paper was to illustrate how learning analytics is bridging the gap between data mined from LMS and teaching practice development in higher education, specifically for Teaching Academics who recently transitioned into online teaching. The authors suggest that bridging this gap is an essential step in the development of online teaching practices.

References

1. Poon, L.K.M., Kong, S.-C., Wong, M.Y.W., Yau, T.S.H.: Mining sequential patterns of students' access on learning management system. In: Tan, Y., Takagi, H., Shi, Y. (eds.) DMBD 2017. LNCS, vol. 10387, pp. 191–198. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-61845-6_20
2. Poon, L.K.M., Kong, S.-C., Yau, T.S.H., Wong, M., Ling, M.H.: Learning analytics for monitoring students participation online: visualizing navigational patterns on learning management system. In: Cheung, S.K.S., Kwok, L.-F., Ma, W.W.K., Lee, L.-K., Yang, H. (eds.) ICBL 2017. LNCS, vol. 10309, pp. 166–176. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59360-9_15

3. Kostopoulos, G., Lipitakis, A.-D., Kotsiantis, S., Gravvanis, G.: Predicting student performance in distance higher education using active learning. In: Boracchi, G., Iliadis, L., Jayne, C., Likas, A. (eds.) EANN 2017. CCIS, vol. 744, pp. 75–86. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-65172-9_7
4. Espinosa, R., García-Saiz, D., Zorrilla, M., Zubcoff, J.J., Mazón, J.-N.: Enabling non-expert users to apply data mining for bridging the big data divide. In: Ceravolo, P., Accorsi, R., Cudre-Mauroux, P. (eds.) SIMPDA 2013. LNBIP, vol. 203, pp. 65–86. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46436-6_4
5. Norton, A., Cakitaki, B.: Mapping Australian Higher Education 2016. Grattan Institute, Victoria (2016)
6. Allen, E., Seaman, J.: Online report card: Tracking online education in the United States (2016). https://onlinelearningconsortium.org/read/online-report-cardtracking-online-education-united-states-2015/
7. Gaebel, M., Kupriyanova, V., Morais, R., Colucci, E.: E-Learning in European higher education institutions: Results of a mapping survey conducted in October-December 2013 (2013). http://www.eua.be/Libraries/publication/e-learning_survey
8. Kaplan, A.M., Haenlein, M.: Higher education and the digital revolution: about MOOCs, SPOCs, social media, and the Cookie Monster. Bus. Horiz. **59**, 441–450 (2016)

Author Index